

販売管理パッケージのメンテナンス スでの 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

富士通アプリケーションズ株式会社

Copyright (C) 2011 富士通アプリケーションズ株式会社 All rights reserved

インタビューの内容

インタビュー日：2010年10月20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）

インタビュイー：富士通アプリケーションズ株式会社 森崎 雅稔

イントロダクション

[中原]

それでは、始めさせていただきます。UMTP、オフショアソフトウェア開発部会主査の中原です。オフショアソフトウェア開発部会では、UMLを使うことによってオフショア開発を成功させようということで、どのようにUMLを使えば効果的に開発できて成功するかということを研究しています。「UML適用ガイドライン」を作ったり、そのガイドラインの普及活動、あるいはいろいろと事例を集めて皆さんにPRするというような活動をしております。その一環で今回、御社のパッケージ保守事例のインタビューをさせていただくことになりました。

では、事前にいただいているアンケートに従ってインタビューをさせていただき、適宜、補足質問をさせていただきます。よろしくお願いします。

プロジェクト概要

プロジェクトについて

御社名は、富士通アプリケーションズ株式会社。ご所属部署名は、ソフトウェアエンジニアリングセンター。ご氏名が、森崎雅稔（まさとし）様。

はい。

この会社にはどれくらいお勤めですか。

私は昨年1月からですので、まだ2年たっていないですね。その前は富士通に所属していました。

同じ開発のお仕事ですか。

そうです。パッケージの開発を入社以来、ずっと担当してまして、昨年1月に富士通アプリケーションズのほうに出向という形で異動してきました。

出向してこられてからの初のお仕事というか大仕事が、オフショア開発だったということですか。

そうではなくて、こちらに来てからは、開発現場の生産性の改善や品質の改善などのために必要な標準化やツールの整備の仕事をしています。かつてパッケージをつくっていたときに使っていた手法——プロセスやツール、規約などを富士通アプリケーションズでの新しい仕事に生かすという目的でこちらに移ってきました。

今回お話しいただくのは、販売管理パッケージ障害調査・修正ということでよろしいでしょうか。

はい。

対象プロジェクト概要はパッケージ開発ということで、これは自社のパッケージでしょうか。

正確に言うと、当社の親会社である富士通の販売管理パッケージということになります。

それから、対象システムの処理系統は業務システムで、対象システムの構築形態は既存パッケージの保守ということですね。

はい。

プロジェクトの予算規模は数百万円とありますが、これは 10 年間、開発してきて、数億円を投資した製品の保守ですか。

数億円を投資して 10 年間、ずっとバージョンアップを続けてきた販売管理パッケージの、ある保守業務、1 プロジェクト分の予算規模という形になります。

オフショア先について

今回のオフショアにかけた（予算規模）ということですね。そして数カ月程度ということですが、具体的には何カ月ぐらにかかったのですか。

実際にプロジェクトとして活動したのは、昨年の 3 月から 9 月までの約 6 カ月間です。

ただし、従来より継続していたものではなく、ずっと国内で保守をやっていたものを今回、このタイミングでオフショアに切りかえたということもありまして、当然、切りかえの準備期間が必要になりました。例えばオフショア先の技術者のスキル調整などもありますし、彼らにやってもらえる仕事かどうかという問題などいろいろとありましたので、その準備期間に約 3 カ月をかけました。実際の保守作業は、残りの 3 カ月ぐらいでした。富士通側から受託した期間も実際には後ろの 3 カ月間の範囲に関してという形で作業をしまして、その期間にかかった費用というイメージです。

準備に 3 カ月かかって、実際に修正した時期は 3 カ月ということで、その実際に動いた 3 カ月が数百万円の予算規模だった。その準備期間の 3 カ月は、オフショア先の選定やスタッフの能力、契約などに費やしたということでしょうか。

そうですね。まずオフショア先という意味では、会社を選ぶことにはあまり迷いませんでした。「プロジェクトのスケジュール」に書いていますとおり、北京富士通系統工程有限公司という富士通の関係会社です。

実は当社が通常使っているオフショア先として、北京と西安の 2 カ所に富士通のグループ企業がありますが、今回のパッケージが C++ という言語で記述されていることもあり、その言語が得意な会社ということで北京を選びました。

また、すぐに立ち上げたいので、技術者がたくさんいて、かつ、すぐに動いてもらえる部署ということで、弊社の渡辺社長と北京の副社長との間で電話をして、すぐにプロジェクトが立ち上がったという形で進めさせていただきました。

北京富士通は、言語としては C++ がすごく強いのですか。

そうですね。例えば CAD 関係のオフショア開発をやっているという経歴もあって、C++ にたけているという印象を持っていました。

プロジェクトの特徴的な部分については、また後で説明していただくかと思いますが、次の項目に行かせていただきます。オフショア先は中国で、北京富士通ですね。どれぐらいの方がいらっしゃるのでしょうか。

約 300 人です。

オフショア先である北京富士通の UML スキルの初期状況については、「当初よりオブジェクト分析も含め高い UML スキルを保有していた」ということですが、この辺の情報は準備期間に既にわかっていたのでしょうか。

いえ、中国で UML を条件にすると技術者の選定がなかなか難しいのではないかと考えていましたので、そこは問わずに、C++ の言語スキルを中心に人を集めてもらいました。ところが実際に

話をしてみると、UMLを知っていますということでしたので、そこはすごくラッキーだったと思います。

では、中国では UML スキルを持っている技術者を見つけるのがなかなか難しい中で、たまたま C++ と UML の使える技術者を集めてもらったということでしょうか。

集めてもらったということではなく、集まった人の中にたまたま知っている人がいたというイメージだと思います。最初にテレビ会議をやったときに、「できれば UML を使って説明したいのですが、わかりますか」という話をしたところ、「大丈夫です」と。ただ、日ごろ仕事で使っているわけではなさそうで、知識として持っているということでした。

では、招集されたというか集まってきた技術者の中にたまたま UML を知っている人がいたということですね。仕事ではあまり使っている様子ではなく、たまたま知っていたということですね。

そうですね。多分、学校で勉強したとかというレベルだったのかもしれませんが。

UML 活用状況

「オフショア向け UML 適用ガイドライン」について

「オフショア開発向け UML 適用ガイドライン」ですが、ガイドラインの存在は特にご存じなかったということですね。

はい。申しわけありませんが、知りませんでした。

UML を採用した工程

「オフショアのプロジェクト実施の際、UML をどの工程で採用されましたか」ということに対しては、製造工程で修正箇所と影響確認を互いに理解するために採用されたとのことですが、その効果としては、プロジェクト関係者の理解のレベルがそろったということですね。

はい。お互いに相手に対して正確に伝えることができたと思います。アプリケーションやパッケージの動作の仕掛けや、障害が起きたときに想像された状況について、こちらから UML のシーケンス図やクラス図を使った図で説明しました。ソースコードも全部先方に渡したのですが、先方からは、「ソースコードを読んだ結果、このように動作していると理解しました」という理解の状況の報告についても、彼らがソースコードから書き起こしたシーケンス図で資料が送られてきました。それでお互いに保守の対象を理解しているレベルをちゃんと確認し合えたということが実際にありました。

では、オフショア側から送られてきた資料を見て、どれぐらい理解できているかを確認できたということですね。

そうですね。そういう意味では、正確にソースコードを読み取ってくれていると感じましたし、UML を使って Q&A のやりとりなどをして、動作や構造について正確に伝え合えたのではないかと思います。

その中身について正確に理解を深めることができたのですか。

はい。保守工程ということで、プロジェクトの対象となるアプリケーションは既にでき上がったもので、しかも彼らがつくったものではないので、そこをいじってもらうときにはきちんと理解できていることが重要です。また、クラスの関係やメソッドの呼び出しの関係などが図でちゃんと書き起こせるスキルがあることで、修正したときの影響が予測できたり、あるいは、外部から見た障害の現象から、内部のどこを探っていけばたどり着けるかということ、彼らに仕事とし

てやってもらえるという感触をそのときに得ました。

結合テスト工程で採用されたということで、この理由としては、必要なテストの範囲を確認するために採用されて、これも同じように理解のレベルがそろったということですね。

はい。実際には 800 からのクラスの集まりで一つのパッケージができています。ところが、今回の修正の対象は 1 クラス、2 クラスを修正するので、どの範囲でテストをすれば今回の修正が正しく行われたことが確認できるかということを決めなければいけませんでした。そのときに、クラス図やシーケンス図を使いながら確認し合ったということです。

UML を使用したきっかけ

次に、UML を使用したきっかけについて確認します。先ほどの話とダブるかもしれませんが、「今回保守したパッケージは、UML を使って設計し、C++ で実装していたため、初めて保守を担当するメンバーに対する説明に UML が都合よかった」とのことです。日本側で担当される方の中にも UML を使われる方が多かったのでしょうか。

そうです。実はこのパッケージは 1998 年から開発を開始して、約 2 年間かけて作りました。恐らく富士通のパッケージとしては、初めて UML と C++ を使ったものだと思います。

プロジェクト立ち上げ時は社内にも UML の技術者は全くいませんでしたし、C++ も、ミドルウェアとか OS に近いところをやっているソフトウェア事業部にはたくさんいたのですが、いわゆる SE 職という職種の中に C++ を扱える人はあまりいませんでした。このプロジェクトを起こすときに、そういう言語の教育やオブジェクト指向、あと OMT の教育を受講しました。オージス総研さんの教育を受講させていただいたり、直接指導に来ていただいたりしながら、勉強させていただきました。自分たちなりにいろいろとデザインパターンを工夫したり、ソースコードの自動生成エンジンを自分たちでつくったりして、800 クラスぐらいで 700 キロステップぐらいの開発規模で最初のバージョン 1 というものをつくりました。

その後の保守でもずっとクラス図を使っています。自動生成も、当時、ラショナル社の Rose (Rational Rose) という、クラス図やシーケンス図が書けるツールがあったのですが、それですべてクラス図を書き、設計情報のシーケンス図を書いていました。そして、その Rose を自分たちで改造して、ソースコードの自動生成ができるようにしました。その後の保守も、ずっとクラス図を使いながらやり続けているという形です。

では、イメージとしては、1998 年から 2 年間、初めて UML と C++ の組み合わせでパッケージを作られて、10 年間ほどエンハンスを続けてきたのですね。

そうですね。10 年間で 3 度のバージョンアップをしていますので、4 世代の製品を作りました。今回保守の対象になったのはその最後のもので、4 世代目なのになぜか v.11 ですけれども (笑)。ほかのファミリー製品のバージョンとの関係で途中が飛んでいるのですが、それまで国内でやっていた V11 の保守の一部を中国オフショアでやってみようということになりました。

UML を使う上で工夫/改善した点

次に、UML を使う上で工夫・改善された点を確認します。UML の図面 (電子文書) 上に自由にメモを書き込み、保存やディスカッションができるようにするために、UML はパワーポイント (やエクセル) で記述されたのですか。

そうです。中国側では UML を日常使ってはいませんでしたので、知識はあるけれども、例えば UML を記述するためにツールを導入しているとか、そういう様子ではありませんでした。ですから、エクセルかパワーポイントで書くしかないかなと思いました。お互いにそこにいろいろと書き込みをしながら、電子文書を交換していました。

実は今回のオフショアは、日本と中国とで本当のオフショアでやりました。要するにオンサイトではなく、オフショアです。私はこの期間、中国に行っていませんし、中国のメンバーも日本には来ていません。コミュニケーションのスタイルとしては、電話とメール、あとは日中間で使えるワークプレイスといえますか、そういうコラボレーションツールみたいなものが社内に開発インフラとしてあります。そのなかにドキュメントとか資産を共有できる仕掛けがありますので、それと電話とメール、あとはテレビ会議だけを使って6カ月間、一緒に仕事をしました。

UML を使ってうまくいった点

準備期間も含めた6カ月間のお仕事の中では、実際に森崎さんが行かれたり、向こうから技術者が来たりすることはなく、電話、メール、ワークプレイス、テレビ会議を使ってコミュニケーションを図られたということでしょうか。

そうです。そのときにUML図を使って、お互いに正しく理解されているかどうかを確認できたのではないかと考えています。

そのコミュニケーションをする上でUMLが、正しく理解しているかどうかという意味ですごく機能していたと感じていらっしゃるのでしょうか。

そうですね。例えば、先ほども言いましたけれども、先方がソースコードを読んで、そこから理解できた情報をシーケンス図に書いて送り返してくれるとか、そういうことをしてくれると、ちゃんと理解できているな、ちゃんと読んでいるな、どこまで読んでくれているのかなということがわかります。もちろん間違いなどに気がつけば、そこに書き込みをして送り返すというのをしながら、やりとりをしたという感じですね。

おおむね正確に読み取ってくれていたと思います。あと、とても日本語が堪能で、中国側は4名という体制だったのですが、少なくともそのうちの2名はとても日本語がお上手でした。Q&A表というものをエクセルシートで作って、課題管理表みたいな感じで交換していたのですが、非常に正しい日本語で質問してくれました。また、それに対してこちらはどうしても難しい日本語で回答しがちなのですが、割ときちんと読み取ってくれていたと思います。そういう文書だけのやりとりも、今回は相当うまくいったと思っています。

中国側の4名は、日本語がかなりお上手なのですか。

少なくとも2名です。4名体制のうち2名は日本語がお上手でした。あとの2名の方とは電話もメールも交わしたことがありませんでしたが、技術者としてはしっかりした方だとは思っています。

では、中国側の4名の技術者のうち2名の方が日本語を非常によく使われて、コミュニケーションが図れたということでしょうか。

そうですね。それから、先ほど言いましたQ&A表や課題管理表のようなエクセルシートのやり方は、とても日本的なのではないかと思っています。今までも日本と一緒に仕事をしていて、私たち以外の委託元との間でもエクセルシートを使って課題管理などをしていただろうと思いますが、仕事のパートナーとして非常にやりやすかったですね。いろいろな方が質問を掲げてくるのですが、必ず同じ方からすぐに私のところに連絡が来て、その人に対して返すと向こう側のプロジェクト内で周知してくれるということがきちんとできていたと思います。

コミュニケーションができるだけでなく、中国ではなかなか浸透していないと考えられるQ&A表や課題管理表をエクセルシートでつくって管理していたのですか。

そうですね。とても日本的なスタイルで、向こうからいろいろとやってくれたということです。

UML の活用についての今後の基本方針

次に、今後、オフショア開発をする際の基本方針ということです。条件により UML を使用するというので、今回のような保守案件でアプリ構造やソースコードの解説が必要な場合に使用したいと思っていて、保守対象がオブジェクト指向言語で実装されていて、きれいなオブジェクト設計であることも条件になると感じていらっしゃるのでしょうか。

そうですね。今回は割と短期間で、プロジェクトとしてはたかだか数百万円とはいえ、もともとは数億円をかけてつくったものですので、保守の対象として、先方が受け入れてくれるまでのやりとりが結構大変です。向こうもリスクを負いますからね。中がどんなつくりになっているかわかりませんし、何百キロステップもあるものを保守として引き受けるのは不安もあるかと思えます。ですから、特に今回のような保守プロジェクトを気持ちよく向こうに受けてもらうためには、UML 図にリバーズして書き起こしやすいついとか、全体的にアーキテクチャに一貫性がある、きれいに設計できている、きれいに書いているとか、そういう保守対象が持っている素質のようなものがとても重要ではないかと感じました。

我々も中のつくりにはある程度自信がありましたので、向こうにこちらが期待しているレベル以上の人がいれば保守できるだろうと思って中国に打診したわけですが、それまで 10 年近く、ずっと国内でやっていたものを国外に切り出すということにはなかなか思い切りが要りました。

・オフショアに UML を導入する際に必要を感じたもの

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すかということです。UML 図を記述するための無償のツールが欲しかった。ホワイトボードのようにコメントの書き込みやマーキングの描画が自由にでき、議論の道具になるもの。そういう無償のツールという感じですね。UML は設計書として清書するだけのツールしかなかったもので、UML 図の上に好き勝手に書き込みができ、考えを練るためのツールが必要だと感じられていたのでしょうか。

そうです。実際にはエクセルやパワーポイントを使ってやることしかできなかったのですが、UML 図を書くのに最適なツールとは言えなかったかなと思っています。ただ、日本側にも中国側にもあって、かつ、一度書いた図の中にお互いに好きに書き込みができるということを考えときに、有償であればそういうツールはたくさんあるのかもしれませんが、無償ではその域に達しているものがなかったと思います。普通に UML 図を書いて設計書として使うということであれば、オープンソースでもそれなりにあったと思いますが、いろいろと相談したり議論したり、いっぱい書き込んだり、いわゆるホワイトボードを一つの部屋でみんなが共有して見ているような形を、離れたオフショア先と日本との間でワークプレイスを使ってやろうとしたときの道具立てとして、今は手に入るものが不足しているかなと感じました。

特に予算が潤沢なプロジェクトでもなく、コミュニケーションという目的のためだけにツールを購入することもできませんでしたので、考えを練るためのツールというものが簡単に手に入ればもっとよかったかなと思います。

今回はエクセル、パワーポイントだけを使っていく中で、もっと日中間で共通に自由に使用できる無償のツールがあればよかったということでしょうか。

そうですね。特にこのオフショア開発は、当社の場合、基本的に日本語で行われています。日本語で UML 図が書けるということが条件なのですけれども、有償の製品では中国側での入手が難しかったり、こちらで購入して先方に送る場合の輸出の手続きが面倒くさかったりするわけです。OSS などで、日本でも中国でもネットから自由にダウンロードして使えるようなものがあればもっと楽しお金もかからないので、いいのではないかと思います。

中国のこの会社にとっては、パワーポイントやエクセルは日本とビジネスをする上で普通に社員が使っているツールで、日本語版のものを使っているのですね。資料に日本語を書き込むとい

うようなことは普通にできますので、今回は日本語が使える、それぞれが持っている共通のツールということで、エクセルやパワーポイントを使うことになりました。

予算のために無償ということではなく、有償のものでは輸出をするときに制限がかかってしまうためですか。

その面倒くささもありました。今回、初めて中国側に保守を委託するに当たって、日本国内、特に当社製のみドルウェアを幾つか先方に提供するのも社内手続などがいろいろと必要でしたし、ワークプレイスを日中間で共有することに関してもいろいろな手続・申請が必要で、準備期間にも結構面倒なことがありました。ですから、立ち上げるまでの期間は十分に余裕を見ておかなければいけないということがよくわかりました。

ちなみに、私はこの仕事が初めてのオフショア作業になりました。ですから手続的に知らないことが多くて、発注元である富士通側の担当者の方と私でいろいろと調べながら、手分けして社内手続を進めたり、いろいろな手配をしたりするという準備が、実際にプロジェクトがスタートするまでにありました。

質疑応答

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：富士通アプリケーションズ株式会社 森崎 雅稔

[中原] 最初に UML を使ったときには、日本側でプログラムの動きとか、どのように動作するかとか、その辺はシーケンス図で説明して、オフショア側がソースコードをもとにシーケンス図に起こして確認したということでした。これは、障害が起きて、このあたりに不良がありそうだという目星をつけて、日本側の技術者が概略のシーケンス 図をつくって、説明して、ソースコードからシーケンス図を起こさせたという理解でいいですか。

[森崎] そうです。

[中原] 母体は、数百キロステップもあって、どういう形で目星がついたのでしょうか。

[森崎] 結構大変でした。最初の3カ月間の体制づくり、立ち上げ期間に向こうの技術者に対していろいろなレクチャーをしました。一つは、パッケージがどういう発想でつくられているかというアーキテクチャを知っていただくということもあって、これが最初に使った資料ですけれども（以下、幾つか資料を提示しながら発言）、ざっくりとした全体のパッケージの考え方、構造と、中にどういうコンポーネントがあるかという具体的な話などをしました。その上でだんだん掘り下げていって、標準的なクラスの階層構造はこんな形になっていますという話から、我々の開発に採用した独自のデザインパターンというものがあるのですね。こういうパターンを伝票の更新や修正などに全面的に適用しています、どの伝票も一貫して必ずこのパターンを守ってつくられていますというようなことをまず説明しました。

それから、トランザクションをどのように制御しているかとか、インスタンスのライフサイクルをどのように制御しているかというようなことを全部 UML で説明しました。

開発してから最初の10年間、日本国内でも技術者がどんどん入れかわったり、日本国内で外注発注してメンバーをふやしたりしたときに日本人向けに説明したときのものを、中国向けに少しリファインして用意しました。

最後のほうに、こちらが想定している障害がありそうな場所について、また別の資料を使って、内部のメモリーでどのようにオブジェクトが管理されているかという話をしました。障害が起きたときのシーケンス図ですね。こういう流れがあったとき、どうもこの辺で起きていそうだと

うことを書きました。先方にはソースコードを全部お渡ししたのですが、彼らが調査するときに絞り込まなければいけない場所について、こちらからある程度ヒントを与えてあげたという形です。

あと、800個からあるクラスにはそれぞれどういう役割があるかというサマリーも書いてお渡ししましたし、もちろんマニュアルもあります。すべてのクラスについてのリファレンスがあるのですが、それは877ページもあって、すべて日本語で書かれていますので、それを全部読んでくださいという、それだけで何カ月もかかってしまいます。ですから、必要などころだけをつまみ食いして読めるように、補助資料として見出しをつくりました。障害周辺に登場してくるクラスや処理の流れについて、また、障害の再現方法、こうやると起きるといことも資料化して、先方にお渡ししました。

そうすると、向こうもいろいろと追跡してくれました。なかなかやるなと思ったのは、私はアプリケーション寄りの領域を詳しく説明して、こういう流れで来たときに異常が起きますという話をしたんですけども、その先にもパッケージの、OSやDBに近い層のクラスがごちゃごちゃとたくさんあって、そこは書かなかったわけです。しかし、彼らはその中をソースコードから追跡して、こうですねと書いて送り返してくれました。

要するに、日本側で作った資料のこのこと、中国側で作った資料のここがつながるような感じですよ。中国側がこれを送り返してくれました。ちゃんと日本語も入っています。これが実際のクラス名です。私たちが見なれているシーケンスとはちょっと違う癖もあるのですが、彼らの理解の程度がとてもよく見えました。

これはエクセルで書いてきています。私がパワーポイントで送ったら、エクセルでその続きを書いてくれました。これは彼らに渡したソースコードから書き起こしてくれたのです。多分、この範囲が調査の対象だろうということに向こうから言ってきてくれました。

[中原] 開発元としてドキュメントはメンテされていたわけですか。細かい処理は別として、基本的な構造はメンテされていましたか。

[森崎] そうです。

[中原] メンテされていて、それをベースにこういうものをつくられたということですか。

[森崎] そうです。特にこの奥の深いところは、これも関係会社に委託してつくってもらったものですが、その部分は、UMLはあまり採用しておらず、UML図は納品されたものの中に一切入っていなかったのですよ。どちらかという、何百ページもあるマニュアルという形で納品されたものが多かったです。業務ルールに近い部分は割とまじめにオブジェクト分析をして、デザインパターンなどを決めて、きれいに書いてあります。そういう意味では、オージス総研さんのセミナーで教わったとおり、業務をちゃんとモデル化して、オブジェクトにして、実際にC++のクラスに落としていくというところまでかなりまじめにつくりました。

ところが、その先の、例えばデータベースと会話する部分とか、今の言葉でいうとO/Rマッピングとか、そういうものをC++で独自に、販売管理パッケージ専用のミドルウェアみたいな感じでつくったのですね。そこはあまりUMLを考えてつくった部分ではなく、純粋にC++のオブジェクト言語のテクニックだけで一気に書いたわけです。

この絵でいっても、この線よりもこちら側（業務寄り）は結構たくさんUML図の設計書が社内に残っていたのですが、このレイヤーよりこちら（ミドルウェア寄り）はソースコードとマニュアルという昔のスタイルの設計書といいますか、フローチャートみたいなものしか残っていないという感じでした。ですから今回は、このレイヤーより奥は、彼らを書いてくれたことで初めてこういう形（UML）になったという感じです。

[竹政] 中国側の方というのはもともと、本で読んだ程度のUML知識だったわけですね。

[森崎] そうだと思われま。日常は使っていないということでした。

[竹政] UMLのレクチャーなどは特にされていないのですか。

[森崎] ませんでした。もともと最初から自社側に UML で記述した資料があった関係で、UML 図を使った形で情報提供したいけど大丈夫ですか、もし知らなかったら UML を勉強してもらわなければならないですけどという話をしたところ、大丈夫です、知っていますという話でしたので、ではこの資料で説明させてもらいますという形で始めました。

[竹政] すごいですね。

[中原] 富士通以外のほかの実務経験があったのでしょうか。

[森崎] 彼らの得意言語は C++なので、今回のメンバーは富士通の CAD 系の開発をやっていたのではないかと思います。

あと、少し予想外というか想定外のことがありました。「プロジェクトのスケジュール」にも少し書いているのですが、立ち上げ準備期間中、ここでは一番下の四角の中ですが、3月13日に初めてキックオフをしました。それから1ヵ月ちょっとたった4月24日に、メンバー全員交換ということがありました。私もびっくりしたのですが、1ヵ月、いろいろと説明して、それを聞いてちょっと難しいということで、もっとスキルの高いメンバーに置きかえたいという話が出てきました。

最初、第四開発部というところから4名、精鋭を用意してもらって、UML もわかります、日本語も大丈夫ですということだったのですが、ソースコードを渡して、説明したとおりにチェックしていった中で、ちょっとハードルが高いと思っただけで、もう少しスキルの高い人をアサインしたいということになったわけです。第六開発部の最適なメンバーがちょうどあいているからということで、ごそっと入れかわりました。

ところが、最初に私がレクチャーした4人から次の4人への引き継ぎは、向こうが全部きちんとやってくれたのです。私としては再度レクチャーし直すことはなく、胸をなでおろしたという感じでした。その後はまた順調に進みました。ですから、先ほどの図面は後期のメンバーが書いてくれました。それも準備期間がちょっと長引いた事情の一つになっているのかなとは思いますが、私自身の負担がふえたということはありませんでした。中国側でうまく引き継ぎをしてくれました。

そういう意味では、意外に UML が読める人も少なからずいるのだなと。また、前半、私が説明したことがちゃんと引き継がれており、それだけしっかりと理解してくれていたんだなということを感じました。

[藤野] 細かい話になりますが、トラブルの内容、実際に発生している場所は、この図ではこの末端のところのように見えますが、実際には中のコンポーネントで起きているわけですよね。

[森崎] そうです。

[藤野] そうすると、ドキュメントは末端のコンポーネントの中の詳細な記述で上がってくるわけですか。末端のコンポーネントの中のドキュメントのアップデートというのはどのようにされたのでしょうか。

[森崎] ここを境に業務層とミドル層という感じで分かれていたのですが、実際には障害の病巣はミドル層の中にあつたということです。それは例えばコアダンプをとって追いかけていたりしながら起きている箇所を確認できていました。

日本側で確認された再現方法と、原因の仮説を先方に提示し、こちらが想定したものが原因かどうかということを確認してもらったりしました。

顛末を言いますと、この中で病巣、どこが悪いかはわかったのです。しかも、この仮説が合っていたという話にはなったのですが、実際にどう修正するかという議論をしたときに、今回は深層で直すのはやめて、異常の引き金となる事象の発生時にふさぐという直し方をしたわけです。

要は、データのあるパターンがこの辺に悪さをするということがわかったわけです。しかし、その状況をつくらないようにすることが業務層で可能だという議論ができましたので、業務層の共通部品を修正することで、ミドル層の深いところにストレスを与えないようにする。異常な事象がおきたときに、ここで制御可能な状態に変えることができるような仕掛けを追加して直したというのが顛末です。今回の作業では、深層の詳細部分のドキュメントの変更はなく、業務の共通部品の設計ドキュメント変更になりました。

そういう異常の仮説を立てて、裏づけがとれて、直す箇所に対して議論ができ、実際に直した結果、異常が回避できたというところまで確認できたというやり方になります。

[藤野] 準備期間が3カ月で、残りの3カ月で対応されたというのは、その一つの現象だけでなく、一連の課題というか障害を直された、保守されたわけですか。

[森崎] 今回、一番プライオリティが高かった障害、この中で異常が起きるというものは一つだったのですが、それ以外にも、今後、RDBなどがバージョンアップしたり、OSが新しくなったりしたときに、改修し続けなければいけないわけです。そういう仕事も先方に頼まなければならないときに、全面的にお願いして大丈夫か。この仕事を通じて中を十分に理解することで、引き受けられる代物かどうか、自信を持ってもらうための期間でもあったと思っています。

そういう意味では、障害の調査対象は、テーマとしてはちょうどいいタイミングだったかなと思っています。例えばRDBMSの古いバージョンで動いたものを新しいバージョンにかえたいということを一気にいきなり出しても、改修するところが多くて大変だと思います。今回はピンポイントで直すためでしたが、全体の流れや構造、仕組みなどをいろいろと読んで理解してもらうということが重要な目的だったのではないかと思います。

障害修正は確かにうまくできて、品質を上げることができました。さらにその先に、今後、OSのバージョンアップやミドルウェアのバージョンアップに対応して直さなければいけないというとき、自分の手に負える代物なのかどうかということが技術者の感覚として判断できるレベルにまで行ったのではないかと思います。

[藤野] 実際に向こうの技量が確認できて、その後、10月以降は保守の体制として確立されているわけですか。

[森崎] そこが非常に回答しづらいところですが。実際には我々の保守作業は途切れています。日本国内で保守していたときも事情は同じですが、業務層には頻繁に手が入っています。また、販売管理の業務パッケージですので、必ずカスタマイズします。そのため、日本国内で業務層を保守する人たちは日々、仕事が回っているのですが、この線よりこちら側、いわゆるミドル層と言われているところは、技術的には安定状態にあります。そういう事情もあって、我々としては障害が起きたときにだけスポットで直しているという感じです。そういうものしか来ないような形になっています。

ですから、これに関しては9月末で作業は完了して、中国側にもこれを扱える人がそれなりに育ってきたというところで、一旦要員をリリースした状態になっています。富士通側のエンハンス計画の中で、業務層はどんどん変わっているのですが、ミドル層は当面このままでいいということになっています。うちで仕事がもらえるのはミドル層周辺だけということもあり、今の仕事は途切れているという状態です。

[中原] UMLを知っている人は、業務の経験はあまりないというお話がありました。業務上でUMLは使っていないくて、本などで勉強した程度ですか。

[森崎] そうだと思います。まず、知っているという事実はありました。ただ、仕事では使っていないという話でした。

[中原] いきなりできるものですかね。リーダークラスの人で経験者がおればできるのですか。

[森崎] そうですね。大丈夫だったようです。モデリングから始める新規開発案件と比べ、保守案件では UML を扱うスキルを多少低く設定できるのだと思います。

[竹政] 今回はオフショアだったのですが、日本で発注するときの説明に使用される資料と同じですか。

[森崎] そうです。日本国内で使っているものとほぼ同じ資料です。日本国内で使っているものはもっとボリュームが多くて 200~300 ページぐらいあるのですが、今回は短期で立ち上げるために、そこから抜粋してダイジェストで説明しました。また、今回はパッケージ全体を保守するという委託ではなかったので、全体の話は概要程度にとどめました。今回は DB アクセス周りのトランザクション上の不具合だったのですが、そこに近いところの議論に早くたどり着けるようなストーリーで説明しました。

[竹政] 日本国内で説明されたときと中国に説明されたときでは、リアクションというか理解度に差はありましたか。

[森崎] この 10 年間、日本国内でもいろいろな人に説明する機会が結構ありました。例えば 2000 年前後、プロジェクトを立ち上げて活動を始めたばかりのころ、人が足りなくなっでどんどん集めました。そのときに富士通の協力会社さんに対してこういう話をしましたが、UML や OMT に関して知っている人は一人もいないというのが当時の実情でした。ただ、説明資料や教科書はつくっていましたが、日本国内でやったときには、UML 技術者認定制度のブロンズレベルに合格していることを発注の条件にして、協力会社さんに勉強してもらいました。ブロンズレベルの合格証を持った人しか、うちのプロジェクトに入れませんと。何度も受け直して合格証を持ってきた人もいました。

一番最初、1998 年にプロジェクトを立ち上げるスタートアップ時のメンバーは全員、オーグス総研さんのセミナーを受けました。特別に当社向けに開催していただき、講師に来ていただいて、受講者は 20~30 名だったと思いますが、全員がプロジェクト要員という形でセミナーを受けました。スタートアップはそういう形でやりました。その後は要員がふえるたびに、ブロンズレベルに合格していることを条件にして独自に勉強していただきました。こちらから教えるということはありませんでした。

[竹政] そうすると、UML の教育という面では、日本は日本でそれなりに苦勞されているということでしょうか。

[森崎] そうですね。そういうことはありました。

[竹政] 今回はたまたま、向こう側にわかる方が多かったということですね。

[森崎] そうですね。日本でも最初、そういう苦勞がありましたので、中国にもあまり期待していませんでした。情報としても、中国ではそんなに浸透していないという話を聞いていましたので、人を集めてもらうときに、例えば「プロジェクトの概要と背景」の下に「技術者に要求したスキル」として三つほど書いています。

C++がわかること、マルチスレッドプログラミングがわかること、Windows Server 2003、2008 に精通していること。それから、富士通製のアプリケーションサーバ、Interstage Application Server、特に CORBA サービスを使って通信をしていましたので、その辺の実装知識のある人を要求しました。それから、Oracle Database でのアプリケーションを開発したことがあるということが必須で、特に排他の仕掛けなどをきちんと実装できる人というような条件をつけました。また、いわゆる ODBC など使わずに、Oracle Call Interface を使っていましたので、それで実装されていても保守できるということをスキルとして要求しました。

しかし、UML については、特に知っていなければいけないということはありませんでした。

ただ、こちらから説明する資料に UML のシーケンス図やクラス図などが出てきますので、必要だということは言ったのですが、それは大丈夫ですという話でした。そこはラッキーだったと思います。4人が4人とも知っていたかどうかは定かでないのですが、少なくとも1人でもいてくれたので円滑にいったのではないかと思います。

[中原] 製品のアーキテクチャはきっちり設計されていて、保守する上で不具合が発生することはありませんでしたか。似たようなクラスが複数あると、いろいろなところを修正しなければならないということになると思うのですが、ちゃんとクラス設計はされていたのでしょうか。

[森崎] そうですね。その辺はかなり気を使ってやりました。特に SE の中で、当時、1998 年ごろにはオブジェクト指向そのものがまだあまり広まっていなかったこともありますし、オブジェクト指向言語の Java が出て間もないころで、Java での開発は時期尚早と判断し、C++が一番まともかなということでしたが、C++で業務アプリをつくったという人もまだ社内にはいなかったと思います。1998 年という、そういう時期ですよ。

それでどうしたかといいますと、研究所にお手伝いいただきました。川崎にある富士通研究所の、当時のソフトウェア研究部の研究員 3 名ぐらいの方に交代で先生として来ていただいて、アーキテクチャ設計や販売管理業務向けのデザインパターンを独自に開発することに支援いただきました。

また、Rose を改造して自動生成したという話をしましたが、実はその Rose の改造も全部研究所で実施してもらいました。自分たちで開発したデザインパターンを Rational Rose の自動生成エンジンに、OLE のインタフェースで独自の仕組みを差し込んで、クラス図を書くと、それに必要な業務アプリ向けのソースコードが自動生成できるようにつくったわけです。その自動生成されたソースコードの中に、業務ロジックを入れる場所はここからここですというマークが入るのですが、SE がプログラムをそこに書き足していくという形で開発しました。

そのような開発プロセスにより、アーキテクチャ的には一貫性が結構とれています。販売管理ですので、受注伝票、出荷伝票、売上伝票、仕入伝票、発注伝票、入荷伝票、あと入金関係など、8 種類ぐらいの伝票を起票して、確認して、登録してというようなことがすべてありますし、伝票間で連携しますよね。それ全体に対して、一貫したアーキテクチャのつくり方でやっています。当時、ソースコードの 66% ぐらいは自動生成して、残りの 34% を業務ロジックとして SE がプログラミングするという形でつくりました。

その間、1 年から 1 年半ぐらい、研究所の方につきっきりでずっと面倒を見ていただきました。研究所の方も、それが論文ネタになったようです。これだけの規模で、これだけの生成率で、こんなロジックがあつてというように数字をたくさんつくって出してあげていましたので、それを使って学会に論文発表したり、本を書いたりされたようで、お互いにメリットのある開発だったのかなと思います。それが 1998 年から 2000 年ぐらいの話です。

その後、当時開発していたメンバーの多くがプロジェクトを離れ、代わりに新しい人へと入れかわってしまいました。課長さんなどの偉い人レベルはまだ残っているのですが、実際にコードを書いていた人は、1 人か 2 人は残っているかもしれませんが、ほぼみんな入れかわってしまっています。ただ、1998 年、1999 年ぐらいに決めたアーキテクチャは今でもきちんと守っています。

残念ながら、自動生成ツールは陳腐化してしまって、もう動かないかもしれません。クラス数は開発当初よりもずいぶん増えていますが、かつてのデザインパターンやアーキテクチャは、変わらず使って維持されています。その辺はすごくこだわりを持って保守しています。

また、今回の仕事をもらったときに、当時を思い出してソースコードを見ましたが、すごくそろっていて、きれいに書けていると思いました。クラスの粒度やクラス間の関係など、いわゆるクラスの持つ責任というものがありますが、その責任の持たせ方などは 1998 年当時に決めたことが今でも守られていて、新しくつくられたクラスもそのポリシーで設計されているということが、見ると感じられます。

[藤野] デザインパターンやアーキテクチャ設計の骨子のところは、研究所の方の指導で構造設計されたということですか。

[森崎] そうです。

[藤野] ロジックのところは SE の方が書き込まれたということですが、逆に言うと、SE の方はそれほどオブジェクト指向に精通していなくてもできるということでしょうか。

[森崎] どちらかという、めちやくちゃオブジェクト指向に精通しているというレベルではなかったかもしれないですね。ただ、デザインパターンは販売管理業務向けにあつらえたものですので、それは SE の頭にきっちりたたき込みました。今まで販売管理パッケージを COBOL で書いていたような人たちですので、パターンの説明や名前なども、その人たちに理解しやすい言葉で書きました。〇〇伝票パターンとか〇〇残高パターンとか、そういうものを 20 前後ぐらいつくって、それを適切に当てはめながら、クラス設計とコード生成と実装の追加ということを繰り返しやってもらったという形です。

そういう意味では、オブジェクト指向でモデリングするという感覚はみんなに結構備わっていたと思います。ただ、継承の技術にすごく詳しいとか、ポリモーフィズム (Polymorphism) をめちやくちゃ知っているという人は全然いませんでした。それどころか、メモリーリークに関する知識もあまりなかったと思います。要は、COBOL などを開発していた人を中心に C++ を習得させました。

「new」や「delete」のやり方にはものすごく気を使いました。基本的にはプログラマーには書かせない。書かせずに済むようなデザインパターン、実装パターンをアーキテクト側で用意するというのを徹底しました。「new」「delete」が必要なコードは抽象クラスとかパターンの中にみんな埋め込んで、プログラマーが自分で「new」と書かずに済むようにしました。

多分、COBOL をやっていた人が C や C++ に移って最初に困るのは、メモリーリークではないかと思っています。それを理解させて使いこなさせるのも育て方の一つだとは思いますが、あえてそれはとらず、メモリーリークについては一切教えない。そのかわり、起こりそうな語彙も教えませんでした。また、ひょっとすると「new」というものが C++ にあるということを知らないで書いている人もいるかもしれない。そのときには、そういうことをやりました。

オブジェクト指向言語でプログラミングするときに必要なスキルという意味では、網羅的に知識を持っている人ばかりではなかったと思います。しかし、モデリングやクラスの役割、責任に関しては、各 SE が業務知識をオブジェクトに結びつける感覚は持っていたと思います。

成功へのポイント

- 製品全体のアーキテクチャ、アプリケーションの構造と振る舞いの説明を UML で行うことで、オフショア先に仕様が正確に伝わり、後戻り作業が発生しない。
- 製品開発時に、アーキテクチャ(UML)、パターン、自動生成等の規則を決め、規則にしたがって開発すれば、保守作業をオフショアに発注する際に改めて UML を使って仕様書を作成する必要がない。オフショアであれば、オンデマンドに優秀な人材を確保できる。