要求定義品質向上への取り組みとEXA-SRS

技術部
ITエンジニア
宮井　剣士郎
Kenshiro　Miyai

複雑化するソフトウェア開発で、お客様に高品質なシステムを提供するためには、仕様漏れなどのプロジェクト初期の仕様に関する問題の解決が必要である。本論文では、当社の要求定義・要求管理の品質向上のための取り組みと、その成果の一つである、オブジェクト指向開発の要求定義フェーズの成果物である「EXA-SRS」(EXA-Software Requirements Specification)について紹介する。

1. はじめに

1.1. ソフトウェア開発の動向と問題

今日のソフトウェア開発は、システム化範囲の拡大によるITシステムにかかわる利害関係者の多様化や、ビジネスの複雑化によるソフトウェアへのニーズの多様化･複雑化により、困難の一途をたどっている。

ビジネスの変化のスピードは留まることを知らず、システムを開発する期間の短期化が求められている。また、異業種間の連携や業務統合などによる、ビジネスの複雑化により、システムへのニーズはさらに複雑になってきている。また、低コスト化に対するニーズはなくなることはない。

ソフトウェア開発者は、お客様に満足するサービスを提供するため、これらの「短納期･低コスト･高品質」を実現しようと、日夜過酷な開発作業を続けている。しかし、仕様の誤り、矛盾、漏れ、そして伝達の失敗といった、ソフトウェアの仕様に関するさまざまな問題が発生してしまう。これらの仕様に関する問題は、プロジェクトの上工程が原因で起きるが、実装やテストといったプロジェクトの下工程で表面化する。発見されたときにはすでに大きな問題となっており、仕様変更･修正･再テストなどの手戻り作業となって開発者を苦しめる。その結果、納期は遅れ、コストは増大し、品質は低下する。このような、仕様が原因で発生する問題は後を絶たない。

では、どうすればこのような問題を解決できるだろうか。

そのためには、お客様のニーズからソフトウェアの仕様を確定する「要求定義」フェーズの品質向上が重要となる。

1.2. 要求定義フェーズにおける課題

要求定義フェーズは、プロジェクトの初期フェーズであり、ソフトウェア要求の抽出･分析･仕様化･妥当性検証･管理の作業が行われる。ソフトウェア要求（以後、要求と略す）とは、問題を解決し、目的を達成するために、ソフトウェアに求められる機能や能力のことである。要求をドキュメントに記述し定義（これを仕様化と呼ぶ）したものが仕様である。

また、要求管理とは、プロジェクトライフサイクル全体で行う要求の管理作業のことである。

前節で説明したソフトウェアの仕様に関する問題を解決すなければならない。

- どうすれば、利害関係者から過不足なく要求を抽出できるか。
- どうすれば、要求や業務を正確に理解できるか。
- どうすれば、開発スコープの拡大を防ぐことができるか。
- どうすれば、要求を正確に仕様化できるか。
- どうすれば、開発者に正確に仕様を伝達することができるか。
- どうすれば、プロジェクトライフサイクル全体で要求の管理を維持できるか。

現在当社では、上記の各課題を解決するために、過去のプロジェクトでの経験や、要求定義のさまざまな方法論をベースに、体系的な要求定義･要求管理手法の定義を行っている。

本論文では、２章で当社の要求定義･要求管理の品質向上のための取り組みを説明し、３章でその成果の一つである要求仕様書「EXA-SRS」について紹介する。

2. 当社の要求定義･要求管理の取り組み

当社では、過去にデータ中心型アプローチでの標準化を行った経験があり、その過去の経験や、さまざまな方法論の調査を基に、体系的な要求定義･要求管理手法の定義を行っている。

特に、今日増加しているWeb/Java案件などの、オブジェクト指向開発ベースの開発プロジェクトをターゲットとした、要求定義手法を作成している。

本章では、この要求定義手法のコンセプトと、コンセプト実現のためのアクションを説明する。

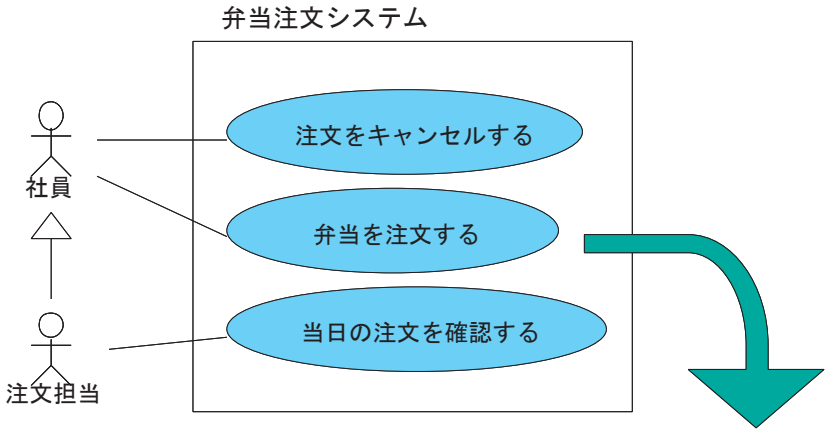
2.1. 要求定義手法コンセプト

当社で現在定義しようとしている、要求定義手法のコンセプトについて説明する。

① ユースケース法による機能要求の抽出

ユースケース法は、Ivar Jacobsonによって作られた、使用法を中心とした観点からの要求抽出とモデリングの手法である（図１）。

ユースケース法では、ユーザから見たシステムの振る舞い（＝ユースケース）を形式的に表現することができ



ユースケース名	弁当を注文する
概要	指定した日の弁当を注文する
目的	アクタが注文した弁当情報をシステムで管理したい
アクター	社員
事前条件	会社が営業日であること
事後条件	アクターの指定した日付の弁当の注文が注文状態になっていること
基本系列	1.アクタはユースケースを起動する 2.システムは、アクタに注文日を提示するよう通知する 3.アクタは、システムに弁当の注文日を通知する 4.システムは、アクタに通知した注文日の弁当が注文されたことを通知する
代替系列	4.a アクタが提示した日付が注文できない日の場合 1. システムはアクタに指定した日には注文が行えないことを通知する
備考	注文ができない日は、会社の非営業日、弁当会社の非営業日、現在の日付の10日前より前である

図１　ユースケース法（弁当注文システムの例）

る。システムの操作をイメージしやすいので、システムに求める機能についての要求を抽出しやすい。

②　オブジェクト指向モデリングによる業務分析

ソフトウェアを開発するには、要求を理解するだけでは不足であり、ソフトウェアの背景に存在する業務の内容を十分に理解しておく必要がある。

システムを期間内にお客様に納品できたとしても、業務の一部品として正しく機能しなければ、お客様が望んだ価値を提供できたことにはならない。業務の一部品として正しく機能できるようにするためには、業務知識を理解し、業務上でのソフトウェアの役割を明確にする必要がある。もし業務知識が把握できないなら、ソフトウェアが業務の一部品として正しく機能することをテストで確認することもできない。

しかし、今日の業務は非常に複雑になっており、業務内容を正確に理解することは困難になってきている。

そこで当社では、業務分析に、オブジェクト指向モデリングを導入することで、業務知識をモデル化し、理解の向上を図る。オブジェクト指向モデリングの表記法には、OMGが規定している標準の表記法であるUML（Unified Modeling Language）を使用する。業務の流れをアクティビティ図で表現し、業務に存在する要素とその関連をクラス図で表現する。このように振る舞いや構造という異なる複数の視点から分析することで、業務の正確な理解が可能となる。

③　追跡可能性（Traceability）による要求の管理

優れた要求に見受けられる特徴として、追跡可能であることが挙げられる（図２）。

要求は、お客様のニーズ（ビジネスや経営上の問題が反映された要望）をシステムで実現するためのシステムの機能や能力として抽出される。お客様のニーズに変更がある場合、その変更がどの要求に影響を与えるかを把握しなければならない。また、要求に漏れや誤りがないことを確認するには、要求の発生元である、ニーズを特定する必要がある。このように、ニーズと要求間の双方向の追跡可能性を把握することで、要求の品質を管理す

ることができるようになる。

また、要求から、ソースコードやテストケースなどの下流工程の成果物への追跡可能性も存在する。例えば、開発中の機能に不具合が起きた場合は、その原因を突き止めるために、発生元の要求への追跡を行う必要がある。

そこで、ニーズから要求、さらにソースコードやテストケースにまで追跡可能性を表すリンクを設定し、プロジェクトライフサイクル全体で追跡可能性リンクの管理を行う。要求定義フェーズでは、ニーズと要求、要求と要求間の追跡可能性リンクを設定する。

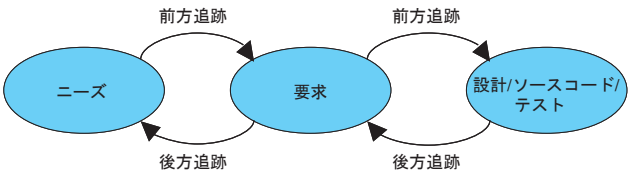


図２　追跡可能性の向上

しかし、プロジェクトの開発者が追跡可能性リンクを一つ一つのニーズ･要求･ソースコード･テストなどに設定し、管理することは非常にコストの高い作業である。また、開発者のスキルレベルにより、追跡可能性リンクの設定の品質にばらつきが出る可能性もある。

追跡可能性リンクの設定･管理を低コストで実現するには、ニーズや要求間の追跡可能性を抽象的なレベルであらかじめ定義することが重要であると考えられる。

④　蓄積した知識の再利用

過去のプロジェクトで収集した、業務知識や要求は、再利用が可能である。

例えば、同業種の業務アプリケーションの場合、法的に決められた業務ルールは共通のルールであるので、過去に作成した業務ルール資料の内容を再利用することができる。また、同業種であれば、その他の業務ルールやシステムの振る舞いに関しても再利用できる可能性がある。

また、システムの特性（「Webアプリケーション」、「C/S」、利用技術など）が似ている場合、非機能要求の内容を再利用できる。例えば、Webアプリケーションの場合、セキュリティ（ログイン権限をもつユーザのみ利用可能）などの品質属性やその実現手段（フォーム認証/ベーシック認証）は再利用が可能である。

このように、業務知識や要求を蓄積して再利用することにより、同じ問題が繰り返し発生することを防ぎ、後のプロジェクトの品質を向上することができる。

さらに、要求の再利用は、その要求を実現するシステムの機能の再利用に発展できる。

2.2.　要求定義手法の構築･導入のアクション

この節では、2.1節で述べた、要求定義手法のコンセプトを具体化し、プロジェクトに導入するために、現在当社で行っているアクションを説明する。

①　要求定義成果物の定義

2.1節の要求定義手法のコンセプトをベースにした、要求定義フェーズで作成するさまざまな成果物を定義する。成果物定義の材料として、当社の過去のプロジェクトで作成した要求定義フェーズ成果物･IEEE[8)]の標準規格・RUPやその他の各種方法論を参考にする。要求定義フェーズの最終的な成果物だけではなく、その過程で作成する中間成果物に関しても定義する。３章で説明するEXA-SRSはこの成果物定義作業の成果である。

②　追跡可能性モデルの作成

高品質な要求の管理を実現するには、追跡可能性を意識することが必要であることを2.1節で述べた。追跡可能性による要求の管理を実現するために、追跡可能性モデルを作成し、要求や仕様間の追跡可能性の関連を抽象的なレベルで定義する。プロジェクトの開発者は、追跡可能性モデルで定義された関連を具現化（インスタンス化）すればよい。これにより、追跡可能性リンクの設定作業の負荷やリスクを軽減することができる。

追跡可能性モデルは、要求定義フェーズ内で抽出されるニーズや要求間の追跡可能性（要求間の妥当性の検証やスコープの管理に利用する）と、要求から設計フェーズ以降に抽出される各種仕様への追跡可能性（変更依頼の対応に使用する）の両方を作成する。

③　要求定義フェーズの開発プロセスの定義

要求定義フェーズの開発プロセスを定義し、要求定義フェーズで行う作業手順を体系化する。この要求定義プロセスでは作業･作業手順･作業の担当者･作業の結果作成される成果物を定義する。各作業で作成する成果物には上記①で定義した成果物を使用する。

④　UMLモデリングのノウハウの蓄積と普及

2.1節で説明した要求定義手法では、オブジェクト指向モデリングを利用する。オブジェクト指向モデリング

の表記法として利用するUMLは、オブジェクト指向の標準化団体であるOMG[10]により表記法が規定されているが、正確で理解しやすいモデルを作成するには、モデリングのノウハウが必要となる。そこで、ノウハウを収集・蓄積し、モデリングガイドを作成して開発者に提供する。

また、社内へのUMLモデリングの普及のため、モデリングの社内教育を行っている。

⑤　要求抽出に利用できるテクニックの収集

利害関係者から過不足なく要求を抽出するには、開発プロセスのような体系化された手法だけでなく、さまざまなテクニックを利用することが重要となる。

要求の抽出に利用できるテクニックには、さまざまなものがある。例えば、ゴール指向分析・ソフトシステム方法論などの問題解決のためのテクニックや、ブレーンストーミング・インタビュー・ロールプレイなどの要求抽出手法、そしてマインドマップや要因分析図などの図がこれに該当する。これらのテクニック調査し、ガイドを作成して開発者に提供する。

⑥　サポート資料の作成

①～⑤で定義した手法のプロジェクトへの導入作業を容易に行うため、ガイドラインやチェックリスト、テンプレートなどのサポート資料を作成する。

次章では、上記で述べたアクションの一つである、要求定義成果物の定義（①参照）で作成した、当社のオブジェクト指向開発における、要求定義フェーズの成果物である「EXA-SRS」を紹介する。

3.　EXA-SRS

ソフトウェア要求仕様書（Software Requirements Specification　以下SRSと略す）は、要求定義フェーズの最終成果物であり、要求をまとめた仕様書である。

SRSは、お客様と要求分析者が共同で作成し、システムの利害関係者が読者となる。お客様にとってはシステムの仕様を確認するための資料であり、システムの開発者にとっては、システムを設計・開発・テストするために必要な情報が記述された仕様書である。

EXA-SRSは、当社のオブジェクト指向開発プロジェクトで作成するSRSの定義と、その定義を記述した各種ドキュメント類の総称である。当社の要求定義手法をベースに、IEEE830[1]の標準規格や、RUP（Rational Unified Process）、Volere[9]などで定義されている各種SRSテンプレートを参考に考案した。

EXA-SRSは2005年9月時点で、Version2.0を公開している。

3.1.　EXA-SRSの構成

・基本構成

EXA-SRSは複数のドキュメントから構成されている（図３）。一般的なSRSは１枚のドキュメントにすべての要求を記述するが、EXA-SRSではこのように複数のドキュメントに要求を記述することになる。その中心となるドキュメントのことを「SRSドキュメント」と命名する。その他のドキュメントは「参照資料」と呼び、これらはSRSドキュメントが参照する各種ドキュメントのことを指す。参照資料に記述されている内容も要求記述である。

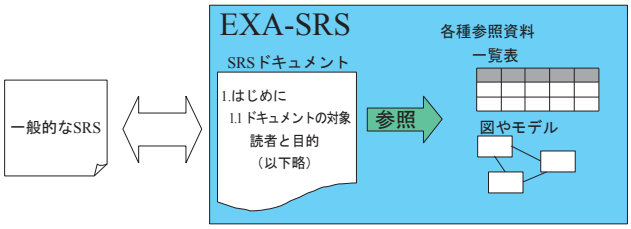


図３　EXA-SRSの基本構成

・　EXA-SRSのドキュメント構成

表1にEXA-SRSのドキュメント構成を示す。

最も重要なドキュメントは、SRSドキュメントである。SRSドキュメントは章や節で段落付けられたテキストドキュメントであり、EXA-SRSの全体の構成を表している。

また、品質属性などのテキスト形式で記述する要求記述は、SRSドキュメントに記述される。

SRSドキュメント以外のドキュメントは、SRSドキュメントの各章が参照する参照資料である。参照資料は、モデルや図、一覧表などのテキスト形式以外で記述されたドキュメントである。参照資料の内容をSRSドキュメントに直接記載して、一つのドキュメントに統合してもかまわない。

表１　EXA-SRSのドキュメント構成（Version2.0）

ドキュメント名	説明	記述形式
SRSドキュメント		テキスト
用語集	「1.3 用語定義」の参照資料	一覧表
業務フローモデル	「2.2 業務概要」の参照資料	アクティビティ図
概念モデル		クラス図
ビジネスルール		一覧表
コンテクスト図	「2.3.1システムの境界」の参照資料	図
ユースケースモデル		ユースケース図
ユースケース仕様書	「3.1.1 ユースケース」の参照資料	形式的なテキスト
ストーリーボード	「3.2.2 設計･実装に関する要求」の参照資料	図
UIサンプル（画面･帳票）		図
外部インターフェース仕様書		図･表

表２　SRSドキュメントの段落構成（Version2.0）

1. はじめに	1.1 ドキュメントの対象読者と目的	
	1.2 ドキュメントのスコープ	
	1.3 用語定義	
	1.4 参照資料	
	1.5 ドキュメントの構成	
2. システム概要	2.1 システム化の背景とシステムの目的	
	2.2 業務概要	2.2.1 業務の流れ
		2.2.2 概念モデル
		2.2.3 ビジネスルール
	2.3　システムの全体観	2.3.1 システムの境界
		2.3.2 ユーザ
		2.3.3 外部システム
		2.3.4 システムが提供する機能
	2.4　システムの基本要件	
	2.5　システムが利用される環境	
3. 要求の詳細	3.1　機能要求	3.1.1 ユースケース
		3.1.2 共通機能
	3.2　非機能要求	3.2.1 品質属性
		3.2.2 設計･実装に関する要求
		3.2.3 本番稼動準備要求
		3.2.4 ユーザ教育要求
		3.2.5 システム運用の要求
	3.3　その他の要求	
4. 付録		

3.2. SRSドキュメントの段落構成

EXA-SRSの記述内容の構成を表しているSRSドキュメントの段落構成（表２）について説明する。SRSドキュメントの段落構成は、大きく３つの章に分かれており、１章でドキュメントの説明、２章でシステムの概要、３章でシステムの詳細について記述する。

下記に、主要な項目について説明する

- 「2.2業務概要」

  業務の内容をシステムへの要求と区別して記述する。業務フローと概念モデルはモデル化し、ビジネスルールは一覧表で整理して表記する。
- 「2.5システムが利用される環境」

  ユーザ数･ピークタイム･クライアントの環境やサーバ環境など、システムを導入した場合に想定している環境に関しての記述である。
- 「3.1.1ユースケース」

各ユースケースに対する詳細（ユースケース仕様）を記述する。

- 「3.1.2共通機能」

  共通機能には、個人認証･アクセス管理･証跡などのすべてのユースケースで共通の機能についての説明を記述する。

- 「3.2.1品質属性」

  品質属性は、さらに、パフォーマンス･信頼性･セキュリティ･保守性･移植性･ユーザビリティに分類される。合意が困難で、あいまいになりやすい要求ではあるが、システムを利用する人にとっては非常に重要な要求である。

- 「3.2.2設計･実装に関する要求」

  開発者が設計や実装時に満さければならない制約である。また、アーキテクチャに影響を与えるので、実現可能性の早期検証が必要な要求である。

  この要求は、さらに、ユーザインターフェース要求･外部インターフェース要求･データベース要求･ネットワーク要求･標準化などに分類される。

- 「3.2.3本番稼動準備要求･3.2.4ユーザ教育要求･3.2.5システム運用の要求」

  これらの要求は、必ずしもシステムやソフトウェアの仕様に関するものではないが、開発者には重要な要求となる。また、この要求が、新たな要求を導出する可能性がある。例えば、システム導入時に、現在使用しているデータをシステムに移行する必要がある場合、移行をサポートするための機能がシステムに必要かもしれない。

3.3. EXA-SRSの特徴

この節では、EXA-SRSの特徴について説明する。ここで述べる特徴とは、IEEE830で定義されている標準的なソフトウェア要求仕様書との相違点である。

① モデル･図･表の利用

要求を仕様化する場合、テキスト形式で記述するのが一般的である。しかし、複雑な対象をテキスト形式の記述で説明しようとすると、本来伝えたい内容とは異なる解釈をされてしまい、利害関係者間で共通の理解を得ることができなくなる場合がある。

EXA-SRSでは、このような問題を回避するため、各種要求をどのような記述法で記述するべきかを指定している。特にモデルや図、一覧表などの表記法を利用することを奨励している。

EXA-SRSで利用されるモデル･図･表には以下のものがある。

- UMLのダイアグラム（ユースケース図･クラス図･アクティビティ図など）
- 図(コンテクストダイアグラム､ストーリーボードなど)
- 表（用語集など）

また、普段上記の表記法とは別の表記法を使用しているプロジェクトなら、異なる表記法を使用することも可能である。

② 業務知識を記載する

プロジェクトの開発者は、作成された要求仕様書の内容をベースにソフトウェアを設計･実装する。しかし、業務システムの場合、仕様化された要求を実現するだけでは不十分であり、作成されたソフトウェアがお客様の業務に適応できていなければならない。業務に正しく適応したシステムを開発するには、開発者の業務知識の理解が重要となる。

そこでEXA-SRSでは、開発者が知っておくべき業務知識（業務フロー、業務に登場する要素と要素の構成、業務上のルール）を記述する。これは開発者に業務上でのシステムの役割や機能を正しく認識してもらうことを目的としている。

また、システムの仕様と業務知識は明確に分離して記述を行う。これはシステムの仕様が、業務に予期できない影響を及ぼす可能性を極力減らすためである。

③ 要求定義作業手順を意識した構成

EXA-SRSの基本構成（3.2参照）は、作成者にわかりやすいように、要求定義作業の手順に沿って構成されている。作業手順に沿った構成は、ある作業の結果抽出した要求を即座に取得することができる。また、プロジェクト用にEXA-SRSをカスタマイズする場合には、この構成をカスタマイズすることになる。

④ ガイドラインドキュメントの提供

プロジェクトの開発者がSRSを作成するときのサポート資料として、テンプレート以外に、ガイドラインドキュメントを用意している。このガイドラインとは、SRSの作成者が守らなくてはならない規約を定義したものである。また、ガイドライン以外にも各項目で記述する内容についての詳細な説明や、記述例も記述されているので、初めてEXA-SRSを利用する人にとっての説明資料としても利用

できる。

⑤　チェックリストの提供

EXA-SRSをベースに作成した要求仕様書の妥当性を確認するための資料として、チェックリストが用意されている。このチェックリストは、SRSの作成者がセルフチェックを行うためのチェック項目の一覧であり、要求の品質や記述に関するチェックを行うことができる。

⑥　事例、サンプルの提供

EXA-SRSでは､過去のプロジェクトでの導入事例を、サンプルとして公開している(導入事例の内容については3.4節参照)。これらのサンプルは、EXA-SRSを導入時に、利用EXA-SRSの内容理解やEXA-SRS導入検討時のボリューム、コストの検討の材料に利用することができる。

3.4.　EXA-SRSの導入メリット

この節では、EXA-SRSをプロジェクトに導入することによるメリットを説明する。

①　要求分析者のスキルの違いによる成果物の品質のばらつきをふせぐ

ガイドライン･テンプレート･チェックリスト･サンプルといったサポート資料を利用することにより、経験が少ない要求分析者であっても、一定水準以上の品質の成果物が作成できる。

②　獲得しなければならない要求を把握できる。

テンプレートの構成や、ガイドラインの説明、サンプルは、仕様化しなければならない要求が何かを、あらかじめ把握するのに役立つ。

③　開発者が業務知識を理解しやすい

EXA-SRSでは業務知識をモデル化して、SRSに記述する。モデル化された業務知識は業務知識をより正確に開発者に伝達することができる。また、記載された業務知識は、システムが業務に適応できるかどうかを検証する場合に作成するテストケースの材料にすることができる。

④　知識の再利用を促進する

EXA-SRSというプロジェクトで共通に定義されたフォーマットを使用することにより、プロジェクト間での知識の再利用を促進することができる。

更に、UMLのモデルに関しては、過去に作成されたモデルを再利用することが可能である。例えば、クラス図で書かれた業務の概念モデルには、アナリシスパターンなどの分析に関するパターンを適用することが可能である。また、同業種の過去のプロジェクトで作成した業務モデルを利用することにより業務の理解を早めることが可能となる。

3.5.　EXA-SRSの導入事例

EXA-SRSをプロジェクトに導入する場合、プロジェクト毎に、カスタマイズして利用することになる。そこで、この節では、過去にEXA-SRSを導入した事例を紹介する。これらの導入事例は現在サンプルとして提供している。

①　オフショア開発プロジェクト

・　プロジェクト概要

小規模Webアプリケーションの開発案件である。要求定義をEXAが行い、設計･実装をインドのIT企業に委託した。

・　EXA-SRSのカスタマイズ

委託先の開発者に要求を正確に伝達するために、委託先の標準SRSと、EXA-SRSの構造のマッピングを行い、その結果から補足SRSを定義した（表３）。補足SRSには、EXA-SRSの基本構成には含めていないプロジェクトの管理に関する項目が記述される。

表３　オフショア開発プロジェクトのドキュメント構成

ドキュメント名	説明	記述形式
SRSドキュメント		テキスト
オフショア対応補足SRS	オフショア用補足SRS	テキスト
用語集	「1.3 用語定義」の参照資料	一覧表
業務フローモデル	「2.2 業務概要」の参照資料	アクティビティ図
概念モデル		クラス図
ビジネスルール		一覧表
コンテクスト図	「2.3.1システムの境界」の参照資料	図
ユースケースモデル		ユースケース図
ユースケース仕様書	「3.1.1 ユースケース」の参照資料	形式的なテキスト
ストーリーボード	「3.2.2 設計･実装に関する要求」の参照資料	図
UIサンプル（画面･帳票）		図
外部インターフェース仕様書		図･表
SRSマッピング一覧	マッピング資料	一覧表

・　結果の検証

日本語と英語という言語圏の違いによる、要求記述の理解のずれが懸念されていたが､UMLのモデルやストーリーボードなどの図を使った要求の記述により、翻訳の間違いによる理解のずれを最小限に食い止めることがで

きた。また、マッピング資料と補足SRSは、委託先の開発者がEXA-SRSの構成を理解するのに非常に効果的だった。今回作成したマッピング資料は、今後同じ委託先に開発を依頼する場合の標準となる。

② 業務パッケージ開発プロジェクト

・ プロジェクト概要

小規模Webアプリケーション案件である。当社で開発した業務システムをパッケージ化するプロジェクトである。

・ EXA-SRSのカスタマイズ

パッケージアプリケーションの要求を定義する時には、企業毎に異なるビジネスルールや要求を、どこまでアプリケーションの機能に含めるかという、開発スコープの検討が重要となる。そこで、基本要件をリスト化し、開発スコープを管理することにした（表４、表５）。

・ 結果の検証

基本要求リストを使って、スコープを管理することで、パッケージ化のスコープを正確に把握することができた。

また、EXA-SRSのガイドラインやテンプレートを確認しながら要求定義を行うことで、運用の要求などの普段見過ごしやすい要求に関しても意識することができた。そのため、要求の抽出漏れを防ぐための手助けとなった。

また、ユースケース仕様をテキスト形式で記述していたが、理解しにくいというコメントをもらった。システムの振る舞いをストーリーボードなどの視覚的な資料で表現し、それを使って議論するなどの工夫が必要だったと思われる。

3.6. EXA-SRSの課題

以下に、現在のバージョン（バージョン2.0）のEXA-SRSの課題について記述する。これらの課題は、今後のバージョンアップで対応する予定である。

① わかりやすさの向上

EXA-SRSでは、業務分析やスコープ定義の記述に、UMLの各ダイアグラムを利用する。UMLによるモデル化は、複雑な対象を正確に、かつわかりやすく表現することができるが、UMLを知らない読者にとっては理解が難しい。そこでUML以外のわかりやすいモデルを提供できる工夫が必要である。

② 多種類の導入事例

要求の再利用を促進するためには、多くの導入事例を知識として蓄積する必要がある。そのためには、EXA-SRS

表４　パッケージ開発プロジェクトのドキュメント構成

ドキュメント名	説明	記述形式
SRSドキュメント		テキスト
用語集	「1.3 用語定義」の参照資料	一覧表
業務フローモデル	「2.2 業務概要」の参照資料	アクティビティ図
概念モデル		クラス図
ビジネスルール		一覧表
コンテクスト図	「2.3.1システムの境界」の参照資料	図
ユースケースモデル		ユースケース図
基本要件リスト	「2.4 システムの基本要件」の参照資料	一覧表
ユースケース仕様書	「3.1.1 ユースケース」の参照資料	形式的なテキスト
UIサンプル（画面・帳票）	「3.2.2 設計・実装に関する要求」の参照資料	図
外部インターフェース仕様書		図・表

表５　基本要件リスト項目一覧

項目名	説明	記述形式
要件名	基本要件の概要	テキスト
内容	基本要件の詳細	テキスト
カテゴリ	要件の分類	業務/機能/実現方法など
スコープの種類	要件のスコープの分類 （開発スコープと優先順位に該当）	開発スコープ内/ カスタマイズで対応/ パッケージ想定外/ システムスコープ外
備考	備考	テキスト
他の仕様への影響	他の仕様へ影響分析結果	テキスト

をさまざまな種類のプロジェクトに導入する必要があるが、まだ導入事例が少ないため、要求の再利用の恩恵を得ることができない。

4. 今後の方針

2.2節で、要求定義の品質向上のためのアクションについて説明した。今後もこれらのアクションを継続することになるが、特に重要視しているのは、追跡可能性モデルの作成と要求の抽出である。

追跡可能性は要求管理を実現するために必要である。要求管理を実現できれば、要求間の矛盾や漏れの発見、開発スコープの拡大防止の効果を期待できる。また、変更依頼に対する影響の把握が容易になり、開発時だけではなく保守においてもその効果を期待できる。このように、システムの品質をシステムのライフサイクル全体で維持することが容易になる。追跡可能性モデルの作成は、低コストかつ高品質な要求管理を実現するための実現手段であると考えている。

また、要求抽出に関しては、インタビューなどの方法だけでは品質の高い要求を抽出することは難しい。利害関係者には、さまざまな人が存在する。エンドユーザは使いやすさを重視し、企業の上層部は投資対効果を重要視する。システムの導入に積極的な利害関係者もいれば、反対している利害関係者も存在する。また、各利害関係者から抽出した要求が、ニーズを実現する要求ではない場合や、抽出した要求が間違っている場合もある。品質の高い要求を抽出するには、お客様と共同で、現状の業務の問題点や戦略、ニーズを分析し、その中で真にシステムに求められているものを見つけ出さなければならない。そのためのノウハウが重要となる。

5. おわりに

本論文では、当社の要求定義に対する取り組みと、その一つであるEXA-SRSについて紹介した。EXA-SRSが利用者にとってより使いやすく、効果が期待されるものになるように、今後も検討を続けていきたい。そのために、EXA-SRSのプロジェクトへの適用結果や、要求定義を実践している現場の要求定義担当者の意見を募集している。

短納期･低コスト･高品質を実現するために、さまざまな要素技術や設計手法の調査･検証が行われているが、それ以前に、必要な要求を過不足なく抽出し、正しく仕様化できているだろうか。また、それらを開発者に正しく伝達できているだろうか。その問題を解決することこそ短納期･低コスト･高品質の実現の第一歩であると筆者は考える。

参考文献

1) IEEE Standards Board."IEEE Recommended Practice for Software Requirements Specifications",USA, IEEE Std 830-1998, 1998.
2) Ivar Jacobson, Grady Booch, James Rumbaugh "UML による統一ソフトウェア開発プロセス".東京,翔泳社,2000. P.125-196
3) Karl E. Wiegers."ソフトウェア要求 顧客が望むシステムとは".東京,日経BPソフトプレス,2003.
4) Pankaj Jalote,"実践CMM インフォシス社におけるソフトウェア･プロジェクトのプロセス".東京.ピアソン･エデュケーション,2002.P.43-64.
5) 大西　淳,郷 健太郎."ソフトウェアテクノロジーシリーズ9 要求工学".東京,共立出版,2002.
6) Dean Leffingwell, Don Widrig."ソフトウェア要求管理 新世代の統一アプローチ".東京,ピアソン･エデュケーション, 2002.
7) Suzanne Robertson, James Robertson."要件プロセス完全修得法".東京,三元社, 2002.
8) IEEE（http://www.ieee.org/）
9) Volere （http://www.volere.co.uk/）
10) OMG（http://www.omg.org）

＜問い合わせ先＞
技術部
開発技術チーム
Tel 044-540-2125　宮井 剣士郎
E-mail: kenshiro-miyai@exa-corp.co.jp

RUP(Rational Unified Process)は、IBM Corporation の登録商標である。
OMG(Object Management Group, UML(Unified Modeling Language) はObject Management Group, Inc. の登録商標である。
