

### 3. インターネットを利用した複数企業間でのソフトウェア開発

大貫一志 三菱電機(株)  
 加藤潤三 コムニック創研(株)  
 石塚英弘 図書館情報大学  
 鮫島千尋 (株)ノス

#### ■コミュニケーションが鍵

電子化とネットワーク化が進展する現状において、歴史・文化のまったく異なる企業同士が、ソフトウェアの開発、運用、保守において総合的な生産性向上、開発のスピードアップ、作業プロセスの維持向上を図るためには、円滑なコミュニケーションのための情報共有と、適切なタイミングで情報交換を可能とする環境が求められる。

一般的に、企業内で常識化していて、普段あまり会話されない情報は企業間の情報交換でも欠如されやすい。具体的には、各社で常識となっている制度や標準、プロセスに関連した情報や用語などが不足することが多く、この不足情報をめぐって、多くのコミュニケーションのロスと不具合が発生している。CALSでは従来からあるEDIの範疇に加えて、データ定義が曖昧な領域である技術情報の共有や交換が求められている。そしてこれらに対し、文化・歴史や言語などが異なる企業や地域、国を越えて行うソフ

トウェア開発でコミュニケーションのロスを極小化する試みが、CALS実証実験として進められてきた。

実証実験では、実際の開発に沿った、開発から保守に至る一連のプロセスに則って、インターネットを介した企業間情報交換を実際に行った。上流工程における情報定義が、継続する工程の情報交換の円滑さに影響を与えることが多いため、ここでは上流工程での情報交換の問題を取り上げ詳述する。

#### 実証実験のシナリオ

実証実験は、図-1のように発注社1社、SI社1社、製造社4社がネットワーク上で情報交換を行いながら、開発を進めた。発注社の要求するシステムを、SI社が取りまとめ、製造社4社が分担開発を行うシナリオである。この他に、実験には実験評価だけを行う評価企業も加え、合計7社20名で、約19,000ステップのプログラム開発を通じて評価を進めた。

実験で対象としたソフトウェア開発プロセスは、図-2に示す流れを想定しており、この中で、SI社が製造社にソフトウェア設計とプログラム開発を発注<sup>1)</sup>するため、要求仕様をまとめるところから実験を開始した。開発するソフトウェアはWebを利用したクライアント/サーバ型の「販売支援システム」である。

#### 実験環境と企業間ルールの設定

分散した環境、企業間で連携してソフトウェアの開発を実現するためには、物理的に情報を共有したり、交換するための条件とともに、書式などの文書形式、内容の検討や、業務上の手順の問題をルール化することが必要であった。このため、実験に先立って、ソフトウェアの開発・保守のプロセスを対象とし、図-3に示す「仕様書」「報告書」などのプロダクトに関する情報と、円滑なプロジェクト推進に必要な管理情報に関する「作成要領」と「開発実施手順書」、およびこれらの実践に必要なツール群を整備した。

ツール群の中核は統合技術情報サービス(CITIS: Contractor Integrated Technical Information Service)である。統合技術情報サービスは、企業間

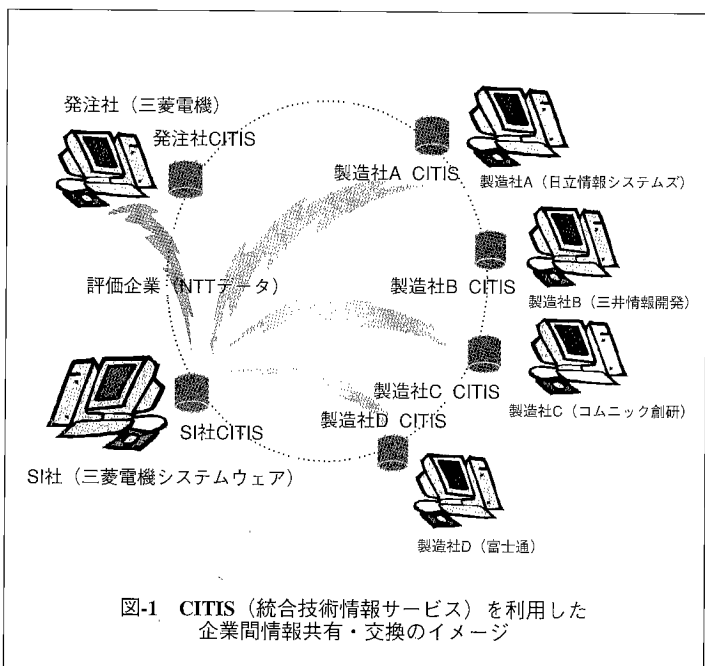


図-1 CITIS (統合技術情報サービス) を利用した企業間情報共有・交換のイメージ

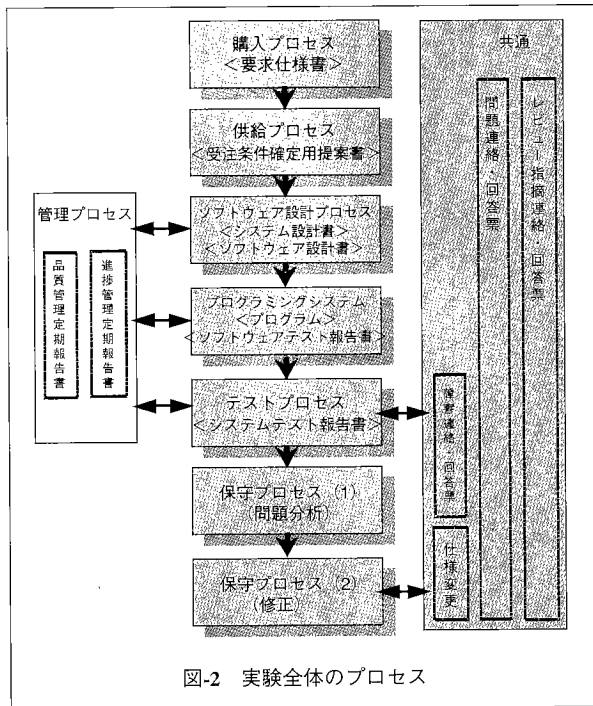


図-2 実験全体のプロセス

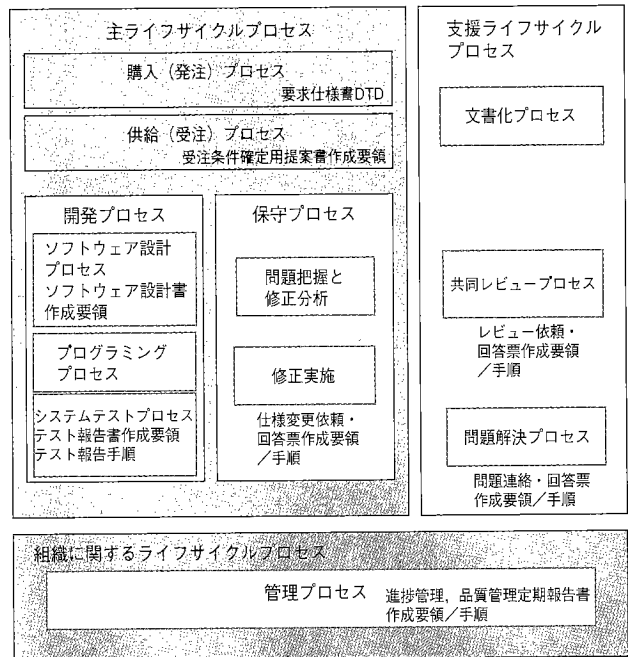


図-3 各プロセスと作成したCALS実施ガイド

での情報を共有・交換するためのサービス群であり、データベースシステムとして実装した。各企業はCITISに対し、文書や連絡票の登録を行う。文書、連絡票などの形式はSGML (Standard Generalized Markup Language)を採用している。また、CITISは登録などのイベントに対し、CITISが関係企業にメールを発信する機能も備えている。

この他、企業間連携を円滑に進めるため「管理データ交換支援機能」「ワープロ文書SGML変換機能」「CASEデータ変換機能」などを開発しながら実験を進めた。

本稿では、図-3における購入プロセスと供給プロセスに関連した要求仕様書から提案書提出に至るまでの情報交換の状況をさらに分析して、実験結果を報告することにしたい。

〔要求仕様書の提示から提案書作成までの実験シナリオ〕

開発の初期段階でSI社が製造社に対して、要求仕様書を提示するところから実証実験をスタートした。この段階では、要求仕様書の内容が各製造社に等しく伝達されることが必要であり、「標準DTD」<sup>2)</sup>を使用して要求仕様書を作成した。また、実際の場面では提案書の準備に際し、要求仕様書に対して質疑応答が発生した。企業間においてこの質疑応答が、的確かつ情報として管理されるよう、この時点から、「問題連絡・回答票作成要領」と「開発実施手順書(問題処理編)」に基づいて実証実験を進めた。

次の段階では、製造社からSI社に対して、要求仕様書に対する提案書を提出する。ここでは複数の企業の提案が考えられる場面であり、プロジェクトでは提案

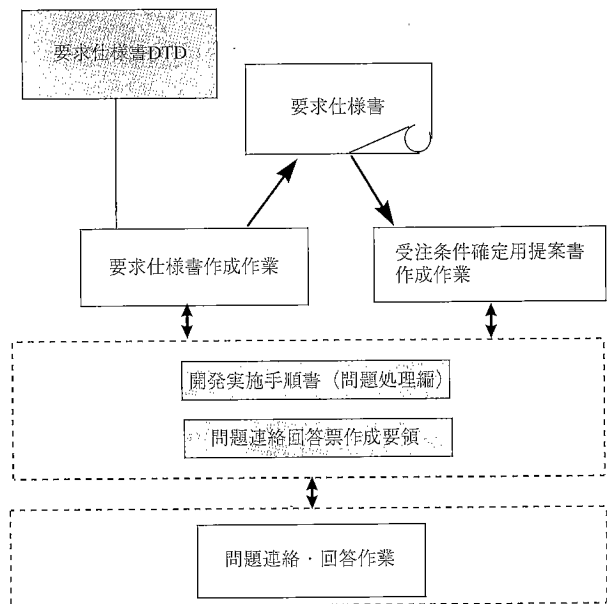


図-4 実施ガイドと実証シナリオ

書の記述方法、形式が統一できるよう「受注条件を確定するための提案書作成要領」を適用した。この場面の「問題連絡・回答」については先述の要領・手順と同様のことがいえる。

関連するプロセスの主な作業内容と実施ガイドの関係を図-4に示す。

〔要求仕様書のDTD化〕

今回の実験では、ドキュメントのSGML化についても評価を行った。要求仕様書についてはDTDの開発を行う必要があった。要求仕様書の文書の内容と構造についてはIEEE Std830 (A7) -1993<sup>3)</sup>で規定され

表-1 要求仕様書の構成

目次
1. 序
1.1 目的
1.2 適用範囲
1.3 定義、頭字語および略語
1.4 参考文献
1.5 概要
2. 全体説明
2.1 製品の見通し
2.2 製品機能
2.3 利用者特性
2.4 制約
2.5 過程と依存性
3. 仕様要件
付録/索引

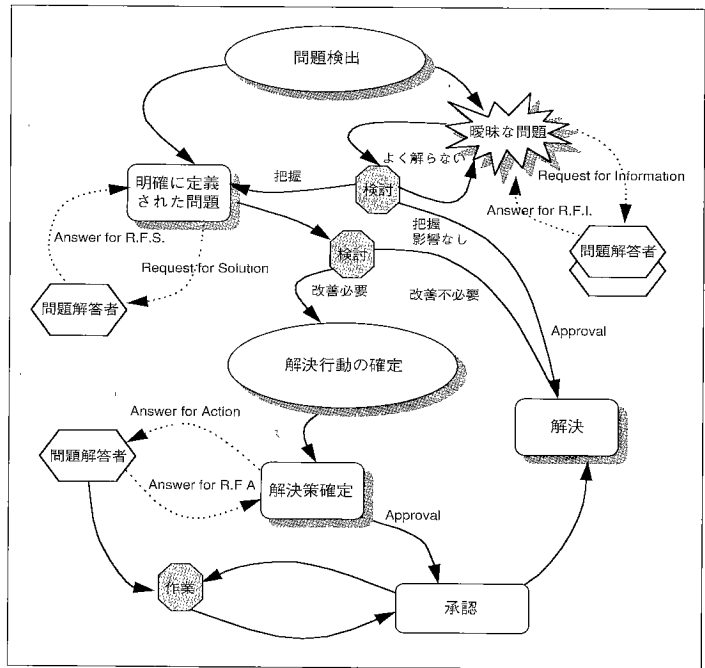


図-5 問題処理概念

たものを採用し、これをもとにDTDを開発した。

要求仕様書の構成概要は表-1のとおりである。

**[問題処理手続きのルール化]**

プロジェクト期間中に発生した問題をスムーズに解決へと導き、また、過去に発生したり、他の場所で発生した問題を参照することで、問題の解決時間を短縮するために問題処理の手続きをルール化した。問題の提起から解決までは、図-5に示すように、「曖昧な問題」・「明確な問題」・「改修する問題」3つのフェーズで複数企業間で、複数回のデータ交換が行われることを想定して、一連のデータのやり取りを1つの問題グループとしてまとめて扱えるようにした。

各フェーズで交換されるデータは、それぞれ情報収集手段 (Information) ・ 解決策収集手段 (Solution) ・ 改修依頼手段 (Action) の意味を持たせ、また問題を帰結させるための承認 (Approval) データを設定した。

**実験データの解析**

実験では、SI社による要求仕様書の作成、製造社への提示、その後の問題連絡・回答票による問題点・疑問点の指摘ならびに回答を経て、承認すべき版の要求仕様書を作成、さらにこれに基づく提案書を作成した。その後のソフトウェア設計、製造の結果が要求仕様書を正しく反映していたため、要求仕様書が「標準DTD」の記述項目/内容としてSI社の要求を製造社へ正しく伝達する上で有効であったと評価される。

しかし、被験者に対して実施したアンケートと問題連絡・回答票を分析してみると、承認された要求仕様書の作成に至る過程でさまざまな問題点がみられた。要求仕様書を作るSI社側と、要求仕様書から内容を把握しようとする製造者との双方の指摘内容、指摘件数のアンケート集計結果を表-2に、また具体的な指摘項

目を表-3に示す。これらの結果からは、インターネットを活用した情報交換においては、対話型で情報交換しながら相手の理解を深めたり、記述内容を修正していくことが不足する。また、提示すべき文書を一気に文書化をしようと試みるため、提示する文書を書く人の思い込みや状況判断の影響が出やすい。結果として、文書を受ける側は理解しにくいという問題が浮き彫りになった。

**[作成者から見た要求仕様書の問題点]**

要求仕様書についてSI社側からは、容易性・理解難易度・実施難易度いずれに対しても困難が指摘された。さらに、ほとんどすべての章、節が標準DTDの理解が困難であった箇所と要求仕様書作成が困難であった箇所としてあげられている。

今回の実験で要求仕様書の作成は、IEEE Std830 (A7) -1993をベースにしたものである。アンケートの回答では、SI社担当者がこれに基づいて作成する要求仕様書を各項目で、何をどのように書くべきかわからないまま作成した経緯があったようである。形式や情報項目だけ決めても対応できないという問題は、今後ソフトウェアCALCを普及させて行く上でしばしば発生する問題であると考えられる。

また、要求仕様書の下記項目はSI社自身が曖昧だと認め、製造社側からも内容が不足している、曖昧である、理解しづらいと指摘された。下記項目に関しては、作成要領などを制定する場合は、特に具体的な作成要領を盛り込むことが重要であろう。

- 製品見通し
- ユーザインタフェース
- データエンティティ
- アルゴリズム/処理方式

表-2 SI社／製造社の要求仕様書評価

評価項目	評価データ名	測定内容	SI社	製造社
網羅性	項目の不足件数	データ項目不足に関する指摘件数	1	7
	内容の不足件数	データ項目の内容不足に対する指摘件数	0	19
冗長性	項目と内容の重複件数	データ項目の重複に関する指摘件数	0	0
	不要項目と内容に関する件数	データ項目を不要とする指摘件数	0	3
均一性	記述レベル	データ項目の記述レベルに対する指摘件数	1	2
正確性	記述の誤り	データ項目の記述の誤りに関する指摘件数	5	3
	曖昧な記述	データ項目の記述の曖昧性に対する指摘件数	4	10
一貫性	矛盾した記述	データ項目の矛盾に関する指摘件数	3	5
明確性	変更履歴の検索性	変更履歴情報の検索性に関する指摘件数	0	3
	関連情報の検索性	関連する情報の検索不可能に関する指摘件数	0	3
容易性	理解の難易度	理解の困難性に関する指摘件数	1	3
	実施の容易性	作成実施の困難性に関する指摘件数	1	0

アンケート対象はSI社1名、製造社3社3名であった。上表は「標準DTD」に基づいて作成された要求仕様書で、SI社の要求内容として製造社へ正確に伝わったか表している。指摘内容は、「標準DTD」の評価項目としてその網羅性、冗長性、均一性、正確性、一貫性、明確性、容易性によって分類した。

さらに、SI社は網羅性の項目について、実験の要件に関わる内容を記述する事例やテンプレートが存在しなかったため、要求仕様としてどのような条件や要求をすべきかが不明確であったことを問題点として指摘している。はじめてソフトウェアCALSの思想で開発を実施した企業・組織では、CALS要件に関わる章について何を記述してよいかわからないという問題が生じたのは容易に想像できることである。実験の成果を基に、記述内容について検討を加え、テンプレートを備えた作成要領を整備する必要がある。

[内容理解しようとする人から見た  
要求仕様書の問題点]

製造社に対するアンケートの分析結果からは、要求仕様書に関して以下のような指摘があった。これらの項目は、要求仕様書についてほぼすべての項目に及んでおり、この要求仕様書を提示された製造社は、容易性－理解難易度の項目からもわかるように3社とも理解が困難であったと指摘している。さらに、SI社が意図しているシステムの範囲は不明確で読み取れなかったとも指摘している。

このような事態が発生した背景として、製造社側から次のようなコメントがあがってきている。

- \* システムに対応する業務シナリオの記述が欠けている。
- \* 全体的に情報量が少ない。

表-3 要求仕様書評価で指摘された問題点

1) 網羅性－項目の不足が指摘された部分	<ul style="list-style-type: none"> <li>・ サービスフロー（業務フロー）</li> <li>・ 運用者に関する情報</li> <li>・ 品質に関する記述</li> <li>・ 性能に関する記述</li> <li>・ プロジェクト全体の体制と日程</li> <li>・ 各工程の納品物と納期</li> <li>・ 発注社およびSI社の体制</li> </ul>
2) 網羅性－内容の不足が指摘された項目	<ul style="list-style-type: none"> <li>・ 製品見通し</li> <li>・ 製品機能</li> <li>・ 外部インタフェース</li> <li>・ 情報フロー</li> <li>・ データ構造の仕様</li> <li>・ 処理概要</li> <li>・ 性能要求</li> <li>・ 設計制約条件</li> <li>・ ソフトウェアシステム属性</li> </ul>
3) 冗長性－不要な項目と内容が指摘された部分	<ul style="list-style-type: none"> <li>・ 製品見通し</li> <li>・ データ構造仕様</li> <li>・ データ辞書</li> </ul>
4) 正確性－記述の誤りが指摘された項目	<ul style="list-style-type: none"> <li>・ データ構造の仕様</li> <li>・ トポロジー</li> <li>・ 処理概要</li> </ul>
5) 正確性－曖昧な箇所が指摘された項目	<ul style="list-style-type: none"> <li>・ 製品見通し</li> <li>・ データエンティティ</li> <li>・ 関連処理</li> <li>・ アルゴリズム／処理方式</li> <li>・ データの構造の仕様</li> <li>・ 構造フィールド</li> <li>・ データ辞書</li> <li>・ レコードタイプ</li> <li>・ トポロジー</li> <li>・ ユーザインタフェース</li> </ul>
6) 一貫性－記述内容の矛盾が指摘された項目	<ul style="list-style-type: none"> <li>・ アルゴリズム／処理方式</li> <li>・ データ構造の仕様</li> <li>・ トポロジー</li> <li>・ ユーザインタフェース</li> <li>・ データ・フローとそのダイアグラム</li> </ul>

- \* 購入者側の業務特性が見えない。
- \* 発注社側の情報が少ない（現状がわからない）。
- \* データ構造の仕様、データ辞書などかなり詳細な記述があるが、なぜそこに至ったかの理由がつかめない。

また、このような背景でプロジェクトが進められた結果として、「SI社の要望が本当に正しく理解できているかわからない。根本的な部分を読み違えていないかを質疑応答で確認したが、それでも必ず正しいと

いう確証が得られない」と同アンケートでは述べている。

一般的に今回の実験で対象としたようなビジネスアプリケーションシステムの開発では、下記のようなプロセスで開発が行われる。

(a) 業務分析

- システム化の対象と考えている業務の現状分析
- システム導入の狙い、システム導入によって解決すべき課題の明確化
- システム導入後の業務プロセスの明確化

(b) システム要求仕様書の作成

- システム構成、性能条件、信頼性条件、制約事項、開発スケジュールの条件と運用の条件記述。

(c) ソフトウェア要求仕様書の作成

- システム要求仕様書の中から、ソフトウェアに関する要求部分をより詳細に記述したもの。

ソフトウェア要求仕様書ではあくまでもソフトウェアに対する要求条件のみが記載されるべきである。ここで「仕様」とは、試験によって確認可能な事項であり、一般的なシステムの説明は含まれない。実験では、ソフトウェア要求仕様書だけではシステムの全体概要を理解することが困難で、システムを把握できなくなることから、結果として上述 (a) (b) (c) のような先行するプロセスに関連した問題点が指摘されるに至ったと考えられる。したがって、ソフトウェア要求仕様書では、IEEE Std830 (A7) -1993をベースに、実践的な作成要領に、以下の項目を盛り込んでおけば、製造社からの上記 (a), (b) に関連するようなさまざまな指摘は出されるには至らなかったと考えられる。

- \* サービスフロー（業務フロー）：業務分析の結果を盛り込む。
- \* 運用者に関する情報：システム導入後の業務プロセスを想定して記述を盛り込む。
- \* 品質に関する記述：業務の特性などから、システム全体としての品質要求を明らかにし、これを受けたソフトウェアの品質要求を記述する。
- \* 性能に関する記述：業務量、業務プロセスおよび運用者からのシステムに対する性能要求、ソフトウェアに対する性能要求を盛り込む。
- \* 製品見通し、製品機能、情報フロー、処理概要、設計制約（上流工程の業務分析の報告書、システム要求仕様書を基に記述する）。

ソフトウェアCALSはビジネスアプリケーションシステムだけを対象と考えているわけではないので、通常行うシステム分析といったフェーズやシステム要求仕様書が存在しない場合がある。ソフトウェアCALSに対する標準、実施ガイドは、現実のシステム開発への適用を考えた場合、当該システムの特性を見極め、より上位の仕様書も含め臨機応変にそのプロジェクトとしてのドキュメント体系を決定することが重要であ

ると考えられる。

また、容易性－理解難易度の項に関連して、ER図が理解できなかったとの指摘が存在した。

今回の実験で使用した要求仕様書に対する標準DTDではER図を使ってデータ定義を実施し、要求仕様書にもER図を記載している。一般にER図は比較的広く普及しているが、表記法の流儀は何通りも存在し、ER図のみを示されても理解できないという事態が生じた。この問題はER図に限らず、図を用いて説明する場合はその図の表記法と意味をどこかに記述しておくことが必要であることをあわせて実証する結果となった。

[対面会議が欲しい]

ドキュメントだけでは理解しにくい、見るより直接聞いた方が早いなどの意見もあったが、対面会議の必要性を調査した結果では、以下の指摘があげられた。

- 1) 文書として書くことを習慣化していない企業がある場合、各社の標準や前後プロセスとの関係を理解させ、意図している部分（重要度など）を明確に伝達する。この場合対話型でないと相互の理解に、非常に時間がかかる。
- 2) 用語の意味の差異があつて、要求仕様書から文章として読み取れない部分を微妙なニュアンスとして理解する。
- 3) 要求仕様書から推測したシステムの概要が正しいかを会話で確認、確証を得る。思い込みの防止。
- 4) ネットワークという媒体による情報交換では、顔が見えない、声が聞こえないこともあつて記述追加（改善）の要望を拒否したり、要望を容れないという回答を出すことが比較的容易に行われるので、問題が解決しないのを抑制する。
- 5) 他製造社との連絡が皆無とならないよう調整することで、SI社と製造社間だけの調整では解決に手間がかかる状況の回避。
- 6) プロジェクト全体の状況把握や、共通的な問題、課題に対する効率的対応や合意の形成。

このような点で対面会議の必要性が求められる結果となったが、一方では海外の製造社も含めたグローバルな調達、グローバルスタンダードに則ったビジネスプロセスの導入を推進する上では電子メールなども利用した文書による意志の伝達に慣れることが必須の課題で、かつソフトウェアCALSを普及させる上ではどうしても克服しなければならない課題である。安易に対面会議を導入することは、結果としてCALSの推進を遅らせ、効果を半減させる要因となるので、情報技術でこれらを解決することが望まれる。

面と向かつて行う会議の場では簡単に断れないが、相手の顔が見えないネットワークによる議論では簡単に断れるという問題は、文書上に明確かつ論理的に意志や要求を表現し、これに基づく議論をネットワーク

表-4

主要文書 (成果物名)	版変更回数	レビュー	問題連絡
要求仕様書	4版/1	実験ナシ	58票
受注条件確定用提案書	10版/3	実験ナシ	41票
システム設計書	4版/1	9票	43票
ソフトウェア設計書	16版/6	36票	実験ナシ
ソフトウェア・テスト報告	8版/5	6票	実験ナシ
システム・テスト報告書	2版/1	3票	実験ナシ

上でできるような方式の確立とスキルを身につけることが重要といえる。この他、製造社相互間の議論も認める運用の必要性も生じている。

**[用語定義が合っていない]**

用語の定義の問題では、「業務表現に関する問題」「システムの仕様表現に関する問題」の2つがあった。後者は、仕様書作成段階でできるだけわかりやすく誤解を招かない表現にするよう心がけ、用語定義を加えるなどの工夫が必要であることが明確となった。前者の問題は、ビジネスアプリケーションシステムの開発の際には常に問題となる事項である。

一般にビジネスアプリケーションシステムにおいては、当該組織で日常的に使われている用語がデータ名称などとして仕様書に頻繁に出現する。これら、用語の意味はそれを使っている組織ごとに微妙にニュアンスが異なっていることが多く、一見してその意味が推測できる用語であっても、その仕様書上で厳密な意味を正しく理解できるようにする必要がある。特に業務に関する用語は、誰が見てもその意味が一応は推測できるが厳密な定義はわからない用語が多く、要求仕様書の用語定義欄に詳細に記載すべきであると推察する。

**実験全体の総合評価**

今回の実験ではISO標準のソフトウェアライフサイクルプロセス (SLCP)<sup>4)</sup> をベースとして実験の流れを取り決め、実際の「販売支援システム」の開発・保守のプロセスに適用した。この結果、インターネットを通じた情報共有、とその交換は、企業間での情報共有と交換の中で、書類 (文書) が不明であったり、伝達されるまでに数日かかるような問題は発生せず、スピーディな情報開示を可能とした点が評価できた。円滑な運用という点で、ワークフローの機能を充実した形態で、大きな開発の流れ (プロセス) に適用できれば、一層信頼のおける情報共有と交換が行えるであろう。

しかし、表-4で見ると、記述内容を固定できるまでの改版数はかなり多く、これらが記述内容の充実にかかなりの時間を費やしている実態や、一方において、

上流工程ほど仕様の詰めがあまい傾向が見られたのも否めない。企業にまたがる情報の交換では、この仕様の詰めが、仕様書を作成する人と読む人の間の「問題連絡・回答」の量に大きく影響していると推察される。系列を離れ、企業間で協調してソフトウェアを開発する場合には、仕様の正確性や詳細記述がかなり要求され、文化や歴史的背景が異なることの影響は、この上流工程の記述レベルの差 (文書化、情報化) として表れてくることを問題視する必要がある。

**■系列を越えた企業連携のために**

系列内の企業同士の場合には、標準や技術基盤が同一であったり、日頃のコミュニケーションや電話などの曖昧さがあっても解決しやすい状況にある。通常あまり表面化しない問題も、複数企業間で開発連携を実現しようとする、情報の曖昧さが、継続する工程で大きくコミュニケーションのロスとして影響することが証明されたといえる。

この曖昧さの大部分は決して難しいことばかりではない。各社内で常識化していること、一見「あたりまえと考えられていること」を情報化し、情報開示へ対応していればコミュニケーションのロスを最小限にできると考えられる。

CALSに象徴される系列を超えた企業連携においては、技術や著作権、特許などの知的財産権の問題、データ責任などの重要な問題もあるが、まずはこの常識化している部分の情報化に各社が注力していけば、ソフトウェア開発がCALSで実現できるといえる。

**参考文献**

- 1) 社団法人情報サービス産業協会: ソフトウェア開発取得モデル契約と解説。
- 2) 日本規格協会ソフトウェアCALSに関する調査研究委員会WG3: 要求仕様書DJD (案) (1998)。
- 3) IEEE Std830-1993: The Parts of an SRS (Software Requirement Specification)。
- 4) ISO/IEC 12207 Information Technology Software Life Cycle Processes。

(平成10年7月14日受付)