

要件定義のあり方を考える

システム企画研修株式会社
代表取締役 上野則男

イントロダクション

(1) 最近の当テーマの関心動向

要件問題は、以下のように各方面で取りあげられています。

1. ITpro

2006/09/01から2006/10/30まで8回にわたって「要件定義」特集を実施した。そのテーマの一覧は図表1を参照。

2. 日経システムズ

2006. 11号特集「図解で極める要求定義」
（「漏れ」「過剰」「誤り」をなくそう）

3. 日経コンピュータ 2006. 4. 17～5. 1

「システム企画の勘所—事業の思いとこだわりを業務要件に具現化する—」「繰り返す要件定義事故、その防止策を企画化する」

4. 日経コンピュータ 2006. 5. 15

「内部統制を意識して要件定義を進める」

1

5. 株豆蔵殿主催の「要求開発アライアンス」

数十社が参加して要求開発（「要求は定義できるものではなく、開発しなければならない」というコンセプト）の方法を研究している。

6. 当社主催「要件不備に挑む事例研究セミナー」

定員20名に対して、38社64名のご参加があった（後述）。

(2) 「要件問題」が話題になる理由

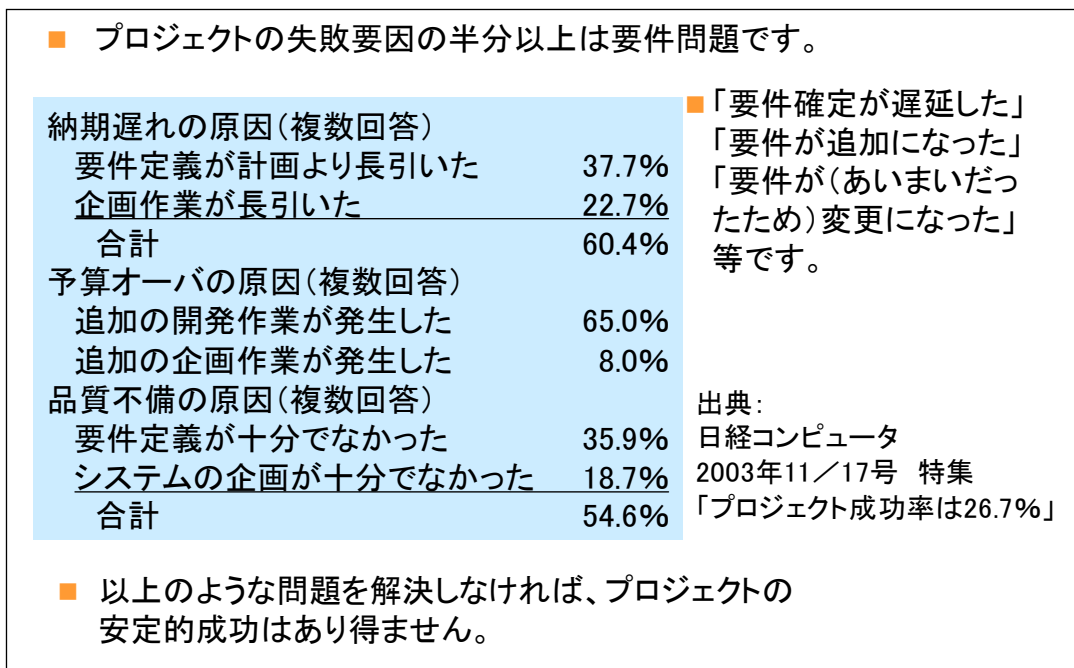
それは、情報サービス業の経営を揺るがせている「失敗プロジェクト」の大きな原因として認識されているからなのです。

- 図表2 日経コンピュータ2003. 11. 17号
- 特集「プロジェクト成功率は26. 7%」参照。
- 図表3 JUAS調査「要件定義不十分などのユーザーの責任は40%」参照。

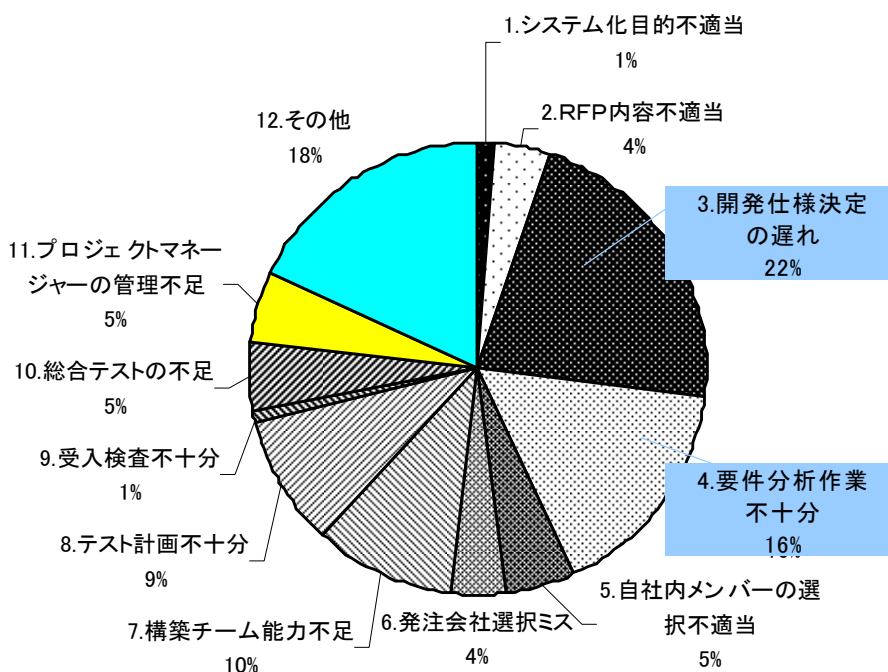
図表1 「要件定義」特集テーマ一覧

●要件定義の方法論を知る
【前編】DOA型要件定義の実際 [2006/09/01]
【後編】日本標準的な作業手順とは？ [2006/09/11]
●非機能要件を見極める
【前編】ヒアリングでは不十分 [2006/09/19]
【後編】ひな型を使い漏れ防止 [2006/09/25]
●上流工程の問題解決 要求定義編
【前編】要求と機能の関係は？ [2006/10/03]
【後編】うまく合意するには？ [2006/10/16]
●上流工程の問題解決 仕様変更編
【前編】仕様変更が出たときの「禁句」は？ [2006/10/23]
【後編】後でもめないための対策は？ [2006/10/30]

図表2 プロジェクトの失敗要因



図表3 遅延理由分析



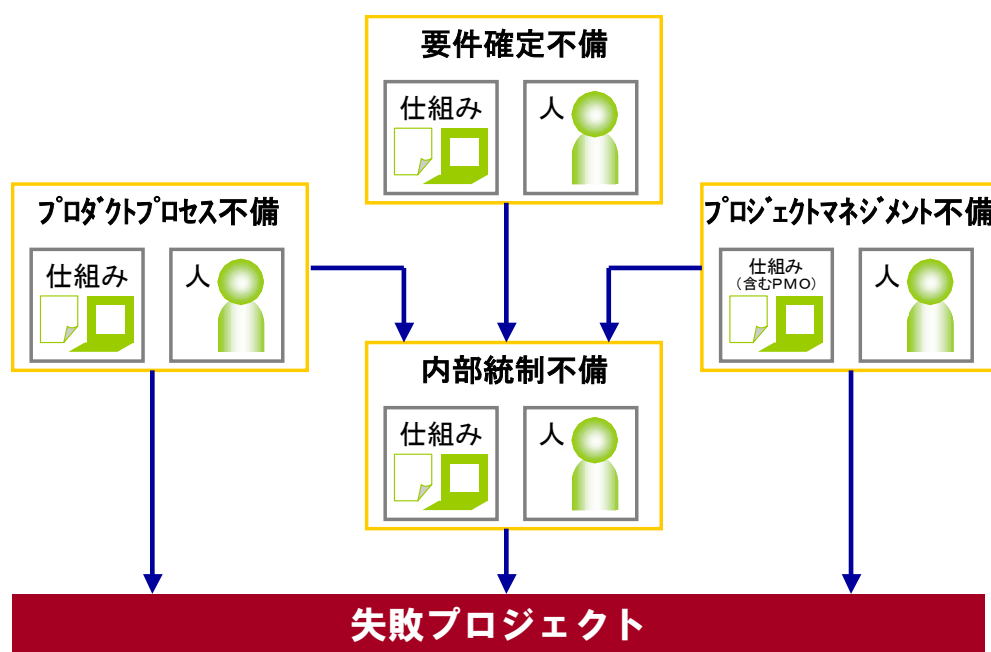
(3)失敗プロジェクトの要因全体の関連

- ここで、プロジェクトの失敗原因をマクロ的に整理しておきましょう。それは図表4のようになります。逆に言えば、これを改善することが失敗プロジェクトの削減につながるのです。
- プロジェクトですから、プロジェクトマネジメントの不備は当然失敗につながります。優秀なプロマネが担当したプロジェクトは失敗がなく、ダメなプロマネが担当したプロジェクトは失敗するというデータを確認された大手SIベンダもおられません。
- 現在は、適用技術・適用開発手法が多様化したため、開発プロセス自体が不確実になっています。「プロダクトプロセスの標準化崩壊」現象です。これの不備による失敗もありますが、過負荷勤務・納期延長等で

切り抜けて大事に至らずに済んでいるケースが多いようです。しかし、実態は大きな問題含みです。この点に気づき、開発プロセスの整備に乗り出している大手ベンダが多くなっています。

- 要件不備に対する受注側の「緊急」対応は、受注審査による受注拒絶や条件改善です。これは「内部統制」としての改善であると見ることができます。
- 2006. 11. 10日経産業新聞「TIS業績回復に薄日」の中で「昨年6月から受注前の提案段階で案件の採算性を確認する『提案レビュー制度』を開始した。(中略)これまで40件以上を審査した。『受注を我が社のほうから断った案件も2、3件ある』(岡本社長)という。」とあります。

図表4 プロジェクトの発生原因



- 同じく2006. 11. 10日経産業新聞に「受注前に見積もり審査」の見出しで「情報システムを受注する前に見積もりを審査する専門部署を設置する。(中略)4人の専任者を配置するほか、自治体や流通など業種ごとの事業部や支社から兼任スタッフを集め、計15人体制とする」とあります。
- しかし、要件不備を内部統制でブロックするのは消極的な対応です。現在のような需要超過状態であればその方策も有効ですが、いずれ需要は減退します。そのときにはどうするのでしょうか。「あいまいな要件でもいいから仕事をください」というわけにはいかないはずです。
- 「要件不備に挑む事例研究 세미나」のように、発注側と一緒に要件を固めるという前向きなアプローチに取り組まな

ればならないのではないのでしょうか。これについては別稿「要件不備に挑む事例研究 세미나大盛況!!」をご参照ください。

- 要件不備の改善が、何といても根本的な解決になるのです。そこで、以下に、要件を確定させるプロセスである要件定義のあり方について研究したいと思います。

1. 要件定義とは何か

要件定義という言葉は、使う人によってずいぶん異なる意味を持っています。外部設計と同様の意味で使う人もいますし、単に開発要求事項をまとめる作業ととらえている人もいます。このテーマの検討を行うには、まず言葉の定義が必要です。

(1) 一般の定義

- 公的な機関の定義として、IPA（情報処理推進機構）のITスキル標準センターの定義は、以下のようになっています。
 - 要求定義：
ビジネスやシステムの課題を明らかにし、あるべき姿（要求仕様）を列挙すること。

- 要件定義：

要求仕様を実際のシステムに実装するために、システム仕様を再定義すること。

- これで言葉の定義にはなっていますが、どこまでシステム仕様を定義するかなど、実質的なことはよく分かりません。

(2) 「要件定義」の実質的な規定

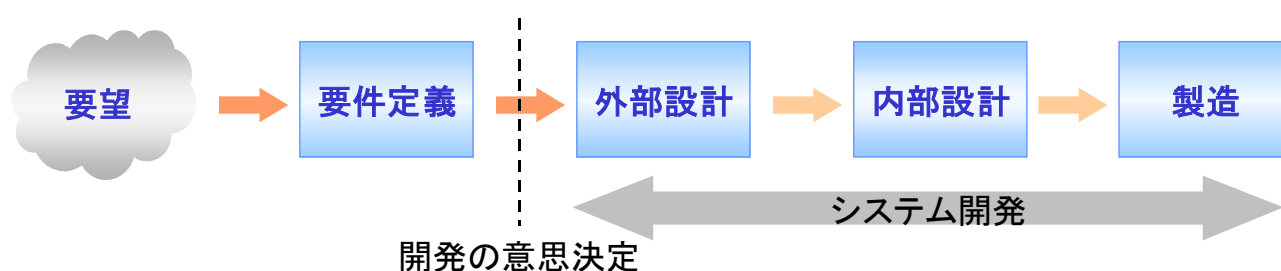
- 本稿では、「要件定義」は「この結果でシステム開発を実施することの意思決定を行う行為または工程」と定義します（図表5参照）。
- そのように定義してはじめて、この工程の実質的な位置づけやあり方を議論できることになります。
- 単に連続的な開発作業の一工程と見ると（概要設計、基本設計、詳細設計の中の概

要設計を行う工程である、というような定義)、実質的な内容規定の決め手が見つからないこととなります。「どうとでも言える」ことになってしまいます。

- ただし、定義ですから別の実質的な定義を行って、そのあり方を検討することがあってもかまわないことです。

- 以下は、ここでの定義を前提に議論を進めることとします。

図表5 要件定義の位置づけ



2. 要件定義の必要条件

(1) 必要条件の内容

- 要件定義の目的が「システム開発の意思決定」であり、その目標が「的確な意思決定ができること」であるとすると、要件定義で達成すべき必要条件は以下のようになります（図表6参照）。
- 「的確な意思決定」の結果は、「後戻りしない、試行錯誤しない」「満足のいくシステムができる」「見積りが狂わない」ことです。
- 「後戻りしない、試行錯誤しない」ためには、システム利用者側では「意思決定者およびシステム利用者の判断が確実であること」が必要です。そのためには、開発の意思決定者が「システム開発の目的・意義

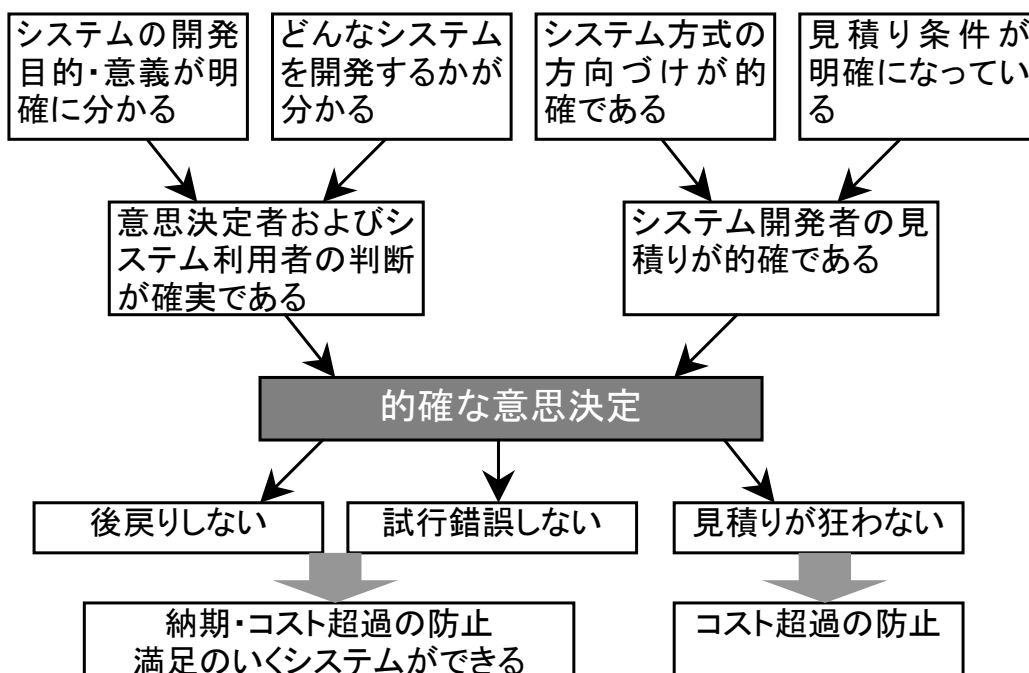
が明確に分かること」と、システム利用者が「どんなシステムを開発するかが分かる」ことが必要です。

- 「後戻りしない、試行錯誤しない」ために、システム開発者側では「的確なシステム方式（アーキテクチャ）の方向づけをする」ことが必要です。この点は、「見積りが狂わない」にもつながります。
- 「見積りが狂わない」ためには、「システム開発者の見積りが的確であること」が必要です。そのためには「見積り条件が明確になっていること」が必要です。
- したがって要件定義の最終成果は、「システム開発の目的・意義が明確に分かる」「どんなシステムを開発するかが分かる」「見積り条件が明確になっている」ことが必要である、ということになります。
- この必要条件が、「要件定義」の実質的な

規定を行うものです。

- この関係を図示すると、図表6のようになります。

図表6 システム開発の的確な意思決定の必要条件



(2)内容の掘り下げレベル

- (1)の必要条件からすると、要件定義の成果物である要件定義書でその内容をどこまで詰める(詳細化する)べきか、については、まず、開発する「システムの内容」の観点から、以下のようになります。

「どんなシステムを開発するかが分かる」と「的確なシステム方式(アーキテクチャ)の方向づけをする」の目標が実現できるレベルでよく、「要件定義以降では、システム利用者との打ち合わせが不要の精度まで記述する」必要はない。

- なぜならば、要件定義段階の検討体制は、まだ正式に開発の意思決定を行っていないので、少数精鋭型のはずです。要件定義後に意思決定が行われると、場合によって開発の委託契約も行われ、開発体制が強化されます。
- したがって、要件定義段階ではなるべく必要最低限度の作業を実施すべきなのです。
- また、「システム利用者との打ち合わせが不要の精度まで記述する」のは外部設計であり、もし要件定義段階でそこまで検討するなら、外部設計で行うことは(ユーザ要件に関しては)ない、ことになってしまいます。
- 次いで、開発に必要な諸資源の「見積りが的確である」ための条件の保証は、以下のようになります。

見積り単位

→システム単位（場合によってサブシステム単位）を明示する。

**機能数（内部ファイル、外部ファイル、
入力処理、出力処理、検索処理）**が分かる。

→「画面・帳票・エンティティ一覧」「データ項目定義」「インタフェース機能一覧」等を作成・明示するが、要件確定段階の「（システム利用者にとって）どんなシステムを開発するかが分かる」というレベルでは、完全洗い出しはできない。そこで洗い出されている範囲から、残りを想定して見積りに加える方式をとらざるを得ない。

全体的な難易度

→総合判断

見積り工数変動要因

→「非機能要件品質目標」「内部統制実施方針」「セキュリティ実施方針」等を作成する。

概略データ量

→「画面・帳票・エンティティ一覧」「インタフェース機能一覧」等に記述する。

3. 要件定義「標準」の作成目的

要件定義の「標準」（以下この「 」付きの標準は、標準書や指針、ガイド等を含んだ総称を言うものとします）の作成レベルは、大きく分けると以下の2つが存在します。

(1)レベル1 MUSTの標準として作成

- 多くの場合の「標準」は、このレベル1から始まります。「決めないとまずい」という判断からこの「標準」を作成します。
- この場合は、要件定義工程の最終成果物である「要件定義書」の必要資料・必要項目のみを規定します。検討プロセスはMUSTには馴染みません。
- ISO9000やCMMI対応で「標準」を設定する場合は、このレベルが主流です。

「方法が決まっていないと困る」のですから、それをクリアするためです。

- 必要資料・必要項目の規定は、必要な資料や項目を洩らすことは防止できますが、内容の質は別問題となります。
- また、「CMMIレベル2では工程標準を前提としているが、レベル3ではテーラリングを要求している」ように、一律の標準化には限界があるのも事実です。

(2)レベル2 BETTERな結果を得るためのガイドとして作成

- このためには、2. 項で挙げた必要条件が充足されるように検討プロセスをガイドし、成果物に関しても要件定義の目的を達成するようにガイドする必要があります。
- かなりのノウハウが蓄積されていないと作成できません。

4. 要件定義「標準」の作成内容

要件定義「標準」の作成内容は、3. のレベルに応じて、以下のようになります。

(1)レベル1

MUSTの標準として作成の場合の内容

- この場合は、要件定義書の成果物を決めればよいのです。例として図表7をご参照ください。
- 一般には、作成する成果物の「様式」「作成要領」「作成例」の3点セットを用意します。
- この方式の限界は、どうやってそこに至ったらよいか不明であることです。
- 様式しか決めていない場合は、記述レベルは作成者の判断・スキルに依存することになってしまいます。

- この記述内容だけでは、第三者がそれではよいかを検証しにくいものです。したがって、レベル1の資料に基づく意思決定はいい加減になる可能性があります。記述されている文言だけで表面的・形式的に判断しかねないからです。
- 特に従来方式の要件定義書は、開発するシステムの内容を中心に記述されていますので（開発目的が明示されていない）、完全性・網羅性には限界があります。「概ねよいのではないか」というような判断になってしまいます。「概ねよい」は後で問題を起こすこととなります。
- それでも、「標準」化の第一歩として、何も決めないよりはましですが、実態はあまり改善されないでしょう。よほど強制指導をしないと、守られない「標準」になってしまうからです。そうなっている会社はかなりあります（当社の調査結果）。

図表7 要件定義書の項目例

<p>1.システム開発の目的・ねらい</p> <p>2.システム化課題と実現目標</p> <p>3.システム化対象範囲</p> <p>3.1 対象組織一覧</p> <p>3.2 対象利用者一覧</p> <p>3.3 対象取引先一覧</p> <p>3.4 対象業務一覧</p> <p>3.5 対象DB範囲</p> <p>3.6 対象データ量</p> <p>3.7 インタフェース対象既存システム一覧</p> <p>3.8 サービス時間帯</p> <p>3.9 その他</p> <p>4.システム開発方針</p> <p>4.1 全般</p> <p>4.2 アプリケーション開発方針</p> <p>(1)業務要件(機能要件)</p> <p>(2)品質・性能要件(非機能要件)</p> <p>4.3 インフラ構築方針(ハードウェア、ソフトウェア、ネットワーク)</p> <p>4.4 移行要件(移行方式、移行時間、並行ラン要件、移行データ作成・教育など移行前準備要件)</p> <p>4.5 運用・保守要件</p> <p>(1)運用要件</p> <p>(2)保守要件</p>	<p>5.開発対象システム</p> <p>5.1 全体システム概要</p> <p>(1)全体システム関連図</p> <p>(2)システム・アーキテクチャ構成図</p> <p>(3)全体システム概念図</p> <p>(4)全体システムDB構造図</p> <p>(5)CRUD全体図</p> <p>(6)ユーザインタフェース標準</p> <p>5.2 個別システム概要(個別システムごとに作成)</p> <p>(1)システム概要</p> <p>(2)業務プロセスフロー</p> <p>(3)機能一覧</p> <p>(4)入出力一覧</p> <p>(5)ファイル一覧</p> <p>(6)コード体系説明書</p> <p>(7)データ項目一覧</p> <p>(8)保留・未決要件</p> <p>6.用語集</p> <p>7.添付資料</p> <p>・議事録</p> <p>・調査・検討資料</p>
---	--

(2)レベル2 BETTERな結果を得るための ガイドとして作成の場合の内容

- 「結果を出す」という点からすると、「こうしなさい」と言わなくても、これに従って作業を実施すると「役に立つ。楽だし、好ましい結果が得られるから進んで利用する」ようになっていなければなりません。
- 検討の役に立つ有効なガイドになるためには、以下のような内容が必要でしょう。

①検討プロセスのガイド

- 要件定義書に至る検討プロセスを設定し、その検討方法をガイドします。
- それを、当社では図表8のような「タスク定義書」と「タスク作業ガイド」というような形で作成しています。

②開発パターン別のガイド

- 現在の開発案件は、件数的に言えばまっさらな新規開発はそう多くありませ

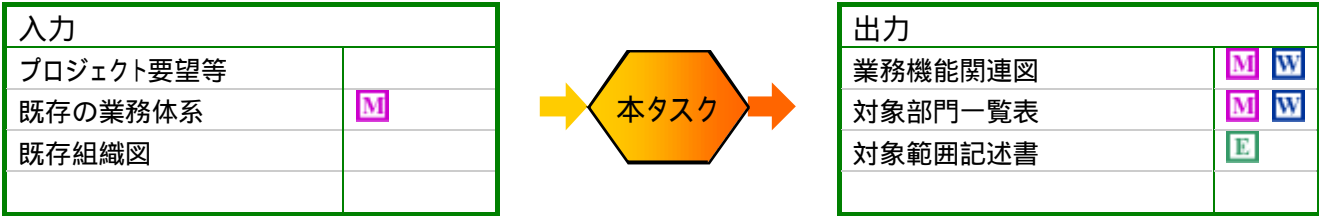
ん。パッケージ利用・再構築などが主流です。これらの案件に対して、「新規開発の方法をベースに、適宜テーラリングしなさい」と言うのでは、役に立つガイドとは言えないでしょう。

- パッケージ利用の場合、再構築の場合に固有の重要な検討内容があるはずですが。たとえば、パッケージの場合の基本的な検討方法は「フィットギャップ分析」です。それをどう行ったらよいかのガイドが必要です。それらをガイドに盛り込む必要があります。
- 図表8にパッケージ利用の場合のガイドが付加されています。
- 図表9は、図表8に対応するパッケージ利用の場合向けのガイドです。

図表8 タスク定義書

要件定義タスク定義書 基本業務系・情報系

タスクの目的
 要件定義を行う対象範囲を規定する。要件定義後のシステム対象範囲の設定でもある。
 開発目的の設定と合わせて、プロジェクト・スコープの決定となる。



作業内容・作業手順	参照手法
1.検討単位の設定	
<ul style="list-style-type: none"> どの範囲をどういう単位で検討するかを決定する。 通常は「システム単位」(仮設定でもよい)であるが、場合によってそれを細分して検討単位とする。 検討メンバの違いで検討単位を判断する。 	
2.プロジェクト要望の確認	
<ul style="list-style-type: none"> 担当部門から提示されているシステム検討要望またはシステム開発要望を確認する。 	1.1(2) 1)プロジェクト要望の確認
3.対象システムの確認	
<ul style="list-style-type: none"> 既存システムがある場合、そのシステム資料を入手し、整理する。 システム関連図があれば、それをを用いて検討対象範囲を明確にする。 	
4. 対象業務の確認	
<ul style="list-style-type: none"> プロジェクトが検討対象とすべき対象業務を確認する。 情報系の場合、情報を提供する対象業務を確認する。 何らかの業務マップを利用することが対象範囲を明確化する上で好ましい(「見える化」の観点から)。 	1.1(3.4)2)c.
4. 対象業務の確認(パッケージ利用の場合)	
<ul style="list-style-type: none"> 対象業務について、その業務が対象事業にとってのコア・プロセスであるか否かを判定する。 この判定は、今後のカスタマイズ対応方針を決定する際等に利用する。 	MIND-ERP2-2-2.2.コア・プロセスの確認
5.対象部門・対象者の確認	
<ul style="list-style-type: none"> 対象業務を実施している部門と担当(職種等)を対象業務ごとに明確化する(「対象部門一覧表」の作成)。 業務マップを用いて記述することが好ましい。 	
6.その他の対象範囲の確認	
<ul style="list-style-type: none"> 当プロジェクトの守備範囲を明確化する上で必要と思われる範囲規定を行う。 可能性のある範囲規定項目は1.1(3.4)2) 対象範囲の項目a.参照のこと。 	1.1(3.4)2)

図表9 コア・プロセスの確認

<p>i. コア・プロセスの確認目的</p> <ul style="list-style-type: none"> • パッケージのカスタマイズを認めるか否かの判定に用いる。 • 新システム検討の際の、力点の置き方の判断材料とする。 <p>ii. コア・プロセスの判定内容</p> <ul style="list-style-type: none"> • 業務機能関連図のマイクロ・レベル単位(場合によってはその下の業務プロセス・レベル単位)で判定する。 • A、B、Cの3ランクとする。 <ul style="list-style-type: none"> • A…事業競争力の源泉となっている業務 • B…それに準ずる業務、もしくはAを支援している業務 • C…事業競争力を産み出しているとは考えにくい業務 <p>iii. コア・プロセスの判定方法</p> <ul style="list-style-type: none"> • プロジェクトの代表メンバ(プロジェクトの人数が少ない場合は全員)が集まって検討する。 • マクロ・レベルから検討を行う。「当社の強みはどこか」→営業と製品開発である。 	<ul style="list-style-type: none"> • 次いで、そのミドル・レベルを選定する。「では営業の強みはどこか」→営業事務、顧客の営業支援、販売管理である。 • 次いで、そのマイクロ・レベルを特定する。「営業事務の強みはどこか」→受注処理である。 • マイクロ・レベルで特定されたものはAとなるが、マクロ・レベルまたはミドル・レベルで選定されている機能の中からBが選ばれることとなろう。例外的に、選ばれていないマクロ・レベルやミドル・レベルからBが選ばれてもよい。 • このような方法で検討すれば、堂々巡りや議論の硬着を避けることができる。結論に異論が出たら、どこから食い違ったかを確認して、そこから戻ればよい。 • 導入パターンA「目標システム実現型」とB「ERPパッケージ準拠型」の場合はこの判定の実施は必須であるが、C「現状システム改善型」の場合にも実施したほうがよい。 <p>iv. コア・プロセス一覧表の作成</p> <ul style="list-style-type: none"> • 検討結果を「コア・プロセス一覧表」として取りまとめる。
---	---

19

③具体的手法からの独立(含むオブジェクト指向)

- データモデリング(データ分析)の手法、ビジネスモデリング(業務フロー・業務プロセスの分析・記述方法)の手法は、数多くのものが存在します。それ以外にも各種定義ドキュメントが必要ですが、その組織が明確に定めているもの以外は、どのような手法・ツールを用いても差し支えないようにする必要があります。
- 受注者の立場ですと、発注者が要求する手法・ツール、部門や対象案件特性に応じた手法・ツールの選択が可能となっていなければ、現実の業務が遂行できません。
- 図表8のタスク定義書の出力欄に、使用するツールをリンクさせて利用できるようにすると便利です。

- オブジェクト指向への対応は行いません。その理由は以下のとおりです。

要件定義は「どんなシステムを作りたいか」をシステム利用者の視点でまとめるものです。オブジェクト指向は、基本的には「どのように作るか」を開発者の視点で検討するものです。オブジェクト指向の要件検討段階で使用する表記法は、いわゆるユーザにとって分かりやすいものではありません。オブジェクト指向対応は、外部設計以降でよいのです。

20

5. 要件定義の進め方

これまで要件定義ガイドのあり方を研究してまいりましたが、そもそも要件定義はどのように進めればよいのでしょうか。私は以下のように考えています。

(1)何かを核に

- 要件定義を繰り返し実施してそれなりの結果を出している人は、何か自分が得意な検討手法を持っていて、それを使って要件定義作業を進めているのです。広い意味の方法論です。

① DFD

- たとえば、DFD（データフローダイアグラム）方式です（図表10参照）。

日本IBM社では標準的な手法になっているようです。

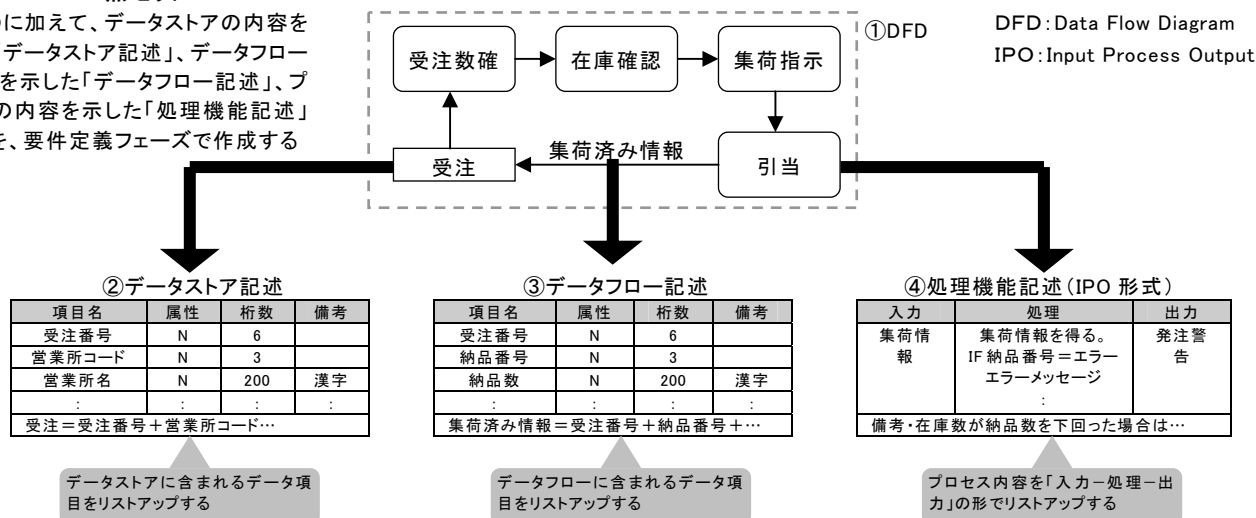
- たしかにこの方法は、情報処理の内容を記述するのにいろいろな利点があるようです。現状を記述する場合や、新しいシステムが考案されてそれを記述する（デザイン表記ツール）にはよいのかもしれませんが。しかし現状（いわゆるAs Is）をこの方法で記述した後、では新しいシステムではどのようにしたらよいか（To Be）は、現状のDFDをいくら眺めても出てこないでしょう。それでは、新システムの方角づけをしようという要件定義の道具だてとしては力不足です。
- 業務の機械化時代には、現状を記述するとその多少の簡素化・改良で新システムは考案できました。現状のDF

DFDは、かなり有効な新システムのデザインツールだったでしょう。しかし、まともなDFDを書けるようになるにはかなりの「年季」が必要だと言いますから、わざわざこれから習得して使

わなくてもよさそうです。極論すると、「DFDは（デザインツールとしては）業務機械化時代の遺物」とでも言えそうです。

図表10 DFD4点セット

DFDに加えて、データストアの内容を示した「データストア記述」、データフローの内容を示した「データフロー記述」、プロセスの内容を示した「処理機能記述」の4つを、要件定義フェーズで作成する



② E R 図

- では、E R 図（エンティティ・リレーションシップ・ダイアグラム）はどのようなのでしょうか。情報システムの核はデータ（データベース）ですから、データの基をE R 図の形にして整理することは、すっきりした構造のシステムを検討するのに極めて有効です。D F Dよりは、新システムのデザインに貢献できそうです。
- しかし、「そもそもこのエンティティは必要なの？」とか、「その項目は必要なの？」「そのエンティティにどんな項目が含まれていればよいのか？」に関しては、積極的な役割を果たしてはくれません。
- 日経コンピュータ2006. 10. 30号「脱“ITメタボリック”大作戦」に富士

フィルム殿の事例として以下の記事が載っていました。

「コールセンターのCRMシステムを検討した際に『1年近く議論した末、既存のグループウェアのシステムを使い続けることを決めた』」

- どんな情報システム（データベース）が必要か？はE R 図を作成すると判明するのではなく、何が必要かを真剣に検討するところから生まれてくるのです。真剣に検討した結果、開発しないという結論も一つの見識です。いい加減な検討で、役に立たないCRMシステムを作ってしまった事例が多いのですから。

【参考】

- 大和総研殿では、ユーザの要求事項（要件）はなかなか決まらないので、システム開発の初めの工程ではE R 図やデータ項目の定義をして、そろそろという後半になってはじめてユーザの処理要件を聞くのだと言われています。
- 「要件定義は下流工程で」（日経ITプロフェッショナル誌2005. 6号池田大造氏「拡張性向上のカギは企業活動全体の把握とDB設計」）
- なかなか刺激的な内容で多くの示唆が得られる発表です。しかし、この方法にも限界がありそうです。基本になるエンティティ自体はビジネスに従属するものですから、新システムとは無関係に作れるし存在するものです。ですからそのレベルの準備をするのは容易なことですが、

データ項目が決まらなければ、それほどの開発負荷の前倒しにはならないのではないのでしょうか。

- 新しい料金算定方式を導入し、その対応システムを開発しようという場合を考えてみましょう。その料金算定方式に応じて、「通話実績LOG」から「料金算定データ」に持ってくるデータ項目が変わります。料金算定データというエンティティ（ファイル）を用意しても、その中身が決まらなければシステム検討は何も前進しません。新方式によっては、「料金算定データ」に関連するエンティティは、当月のメールの使用実績や過去の通話実績やメール使用実績に関係するかもしれません。やはり、新料金算定方式（＝ユーザ要求事項）が決まらなければ多くのことはできないでしょう。

③ビジネスモデリング・ツール

- 業務プロセスを記述できるビジネスモデリング・ツールが、内部統制対応もあり、多数発表されています。これらも、基本的にはDFDと同じ性格です。新しいプロセスを発想・デザインすること自体には直結しない、という限界があります。
- 詳細の議論は省略しますが、CRUD図ファンのベテラン技術者がおられます。CRUD図はシステムの不整合を点検したり、適切なサブシステム分割をしたりするには有効ですが、やはり積極的にシステムの方向づけを検討することには貢献してくれそうにありません。
- しかし、何かは必要なのです。

(2)目的指向のアプローチ

- では何が、どんな指導原理が、システムの方向性を導いてくれるのでしょうか。システムの検討の入り口からずっとシステム検討に筋道をつけてくれるものがあるのです。それは「システム開発目的」です。システム開発目的は、無の状態から検討を始めることができます。「何のためにこのシステムを開発しようとするのか？」この質問はシステム検討の冒頭で、システムの開発をしようとしている人に投げかけることができます。突飛なことではありません。何らかの目的や意図があって、システムを考えるのですから。
- まともな思考順序では、システムの構想（手段）が先にあって「このシステムを作ったらどんな効果（目的）があるか」と考えるではありません。少なくともマネジ

25

メント層の人間はそう考えないでしょう。マネジメント層の考える目的は、自分のビジネスや業務に貢献するというのを目的にするはずです。業務に貢献するとは、業務成果が向上することです。

- ゴールドラット博士の「チェンジ・ザ・ルール」にも、その点が明快に述べられています。ご参照ください。
- そのような目的を我々は「（経営的）価値目標」と言って、業務の「早い、うまい、安い」など10個の切り口で考えることをお勧めしています。
- 開発目的の検討から始める要件定義のアプローチは、大まかに言うと、以下のような順番で検討を進めることになります。

- このシステムの開発目的は何か（それを完全定義する）。
- その目的を達成するための実現手段（解決策）は何か。
- その解決策は、どの業務プロセスに影響を及ぼすか。
- 該当する業務プロセスはどういう変化（アウトプットの変化）を求められるか。
- その業務プロセスはどのような処理方式にならなければならないか（新システムのデザイン。）
- かたや、どのようなデータが必要とされるか（エンティティやデータ項目の整理）。
- 業務プロセスを支えるためのインフラ条件はどうなるか。

26

- この方法は「目的指向のアプローチ」ですが、目的は、オブジェクトではなく、英語で言えばPurposeです（さしずめPOAでしょう）。目的にかかわらないことには手を出さない、極めて効率的な検討手順です。
- このアプローチ法では、現状調査から始めることはしません（現状調査から開始する方法も業務機械化時代の名残です）。現状調査・確認は、目的検討段階あるいは実現手段（解決策）検討段階で必要に応じて行うこととなります。その場合は調査・確認の目的が明確ですから、余計な手間ひまはかかりません。
- このプロセスを実行するキーになるのは、目的を探求する方法です。弊社では「問題点関連図手法」を利用しています。通常の問題分析手法では、問題の因果関係分析や問題の層別整理を行うだけで、目的設定を

行うことはできません。

- 当社では、20数年前にQCの関連図法からスタートして、目的設定までできる手法に育てあげました。この手法は衆知を集めるにも適しています。衆知の結果で合意形成が得られません（以下のURL参照）。
<http://www-cont.newspt.co.jp/manual/yes/oc/mkoc.html>
- 問題分析だけを行う手法は、これも業務機械化や現状改善型の案件の場合には有効ですが、改革型で新しいシステムを開発する場合の目的は見つけ出してくれません。
- 現在要求されているのは、（新規開発案件では）ほとんどが改革型なのです。「新春特集3」で野村総研の荒生氏が主張されている仮設検証型のシステム開発は、やはり改革型です。
- 大手ITベンダの提供しているセッション

ン型のサービスの多くは、問題整理型のアプローチです。改革型の目的探求には力不足です。

- 有名なナドラ氏の考案された「ワークデザイン法」は、目的を探求しますが、その場合の「何のために？」は、以下のような機能の目的階層を追求するものです。「～する」は機能であって、価値目標としての目的のとらえ方ではありません。価値目標は好ましい「状態」の表現（早い、うまい、安い）であって、「～する」ではないのです。

会議の運営原則にのっとり会議を運営する。

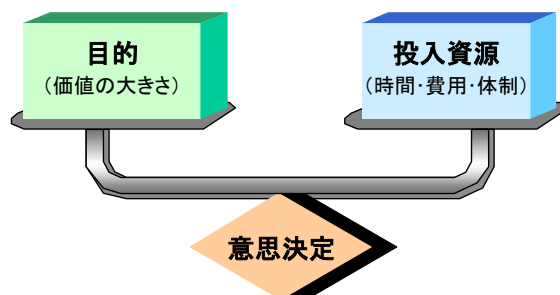
↑
会議の運営原則を出席者が認識する。

↑
会議の運営原則を会議室に掲示する。

↑
会議の運営原則を決める。

【システム開発目的の完全定義】

- 「システム開発目的」は、意思決定者の重要な判断項目です。開発の意思決定は以下の図のように、開発目的（＝効果）と投入資源のバランスで行われます。開発目的が定量化・具体化されていなければ、的確な判断はできません。どの程度の効果があるのかが「適当」「概ね」では駄目でしょう。



- したがって開発目的は、お題目ではなく、内容が具体化され定量化された完全定義であるべきなのです。
- 「このような完全定義を行うと開発結果の満足度が飛躍的に高まる」ということがERP研究推進フォーラムの調査結果で発表されています。
- 開発目的の完全定義についての詳細な説明は本号別稿「要件確定の基本解説（抜粋版）」を参照してください。例えば以下のような内容です。

【受注・出荷システム開発の目的】

- (1) 受注業務の工数を削減する。
 - 受注業務の工数を現状の10分の2程度に削減する。(10分の2=平均受注業務時間1件当たり2分以内)
- (2) 受注業務の精度を向上させる。

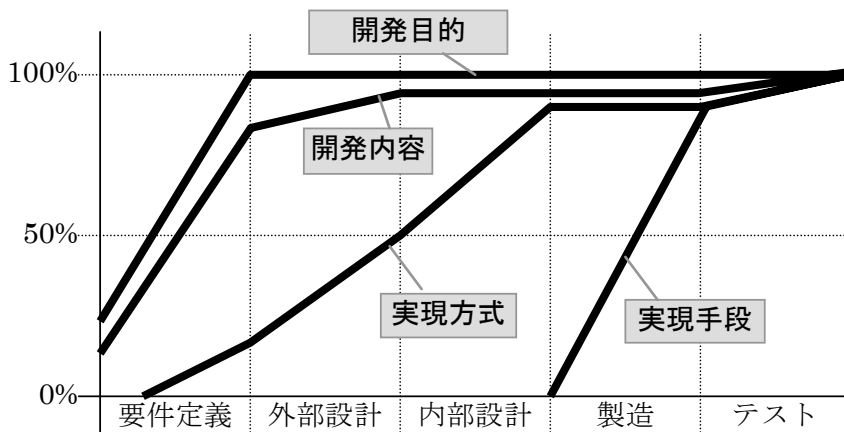
- 受注業務の誤りをほぼゼロに近いレベルに高める。
- (3) 出荷手配と製造予約の工数を削減する。
 - 出荷手配・製造予約の工数を半減する。
 - (4) 納期回答を迅速化する。
 - 納期回答を通常品は即時回答、それ以外も当日回答とする。

- 以上は、「開発目的」は意思決定の重要な評価要素であるから完全定義をすべきだ、という主張です。
- 一般には、可能なかぎり「開発内容」を決めようとしていました。前に述べましたように、開発内容の完全定義は、要件定義段階では無理ですし、すべきでもないでしょう。要件定義工程では、開発内容は80%程度固まっていればよいと思われます。

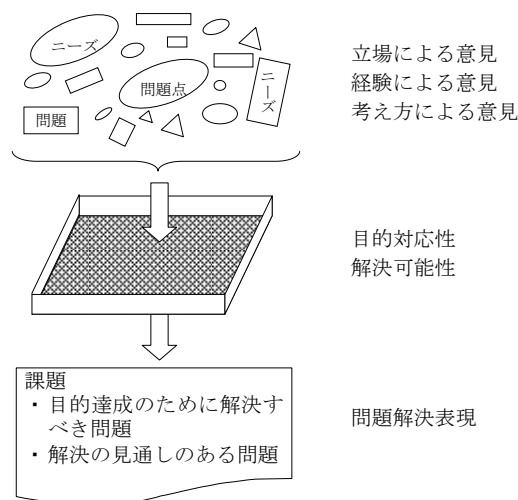
- 開発内容の完全定義は、外部設計工程でするものです。開発内容を実際の実現方式・実現手段（IT環境、プログラム、業務マニュアル）に落とすのが、内部設計・製造工程です。
- しかし、実現方式の検討は要件定義工程から始め、基本方針は固めなければ

- なりません。そうしないと、要件定義段階の見積りに狂いが生じたり、後工程での手戻りが発生したりしてしまいます。
- 開発工程と決定項目の関係を図表11に示しました。

図表11 開発工程と決定項目の関係



- 「目的指向のアプローチ」は、後工程でも効果を発揮します。要件定義段階で開発目的を完全に決定しておきますと、外部設計工程でシステムの内容を具体化する際に、システム利用者が持ち出す要望に対して、「開発目的」がフルイの役割を果たしてくれます。目的に合致しない要望は取り上げないのです。開発目的があいまいですとフルイがザルになってしまい、声の大きい人の要望が通り、混乱が始まるのです。
- 皆様も、要求事項が次第に膨らんで、難儀をされたご経験がおありだと思います。それは有効なフルイがないからなのです。
- もしどうしても目的に合致しない要求を取り上げたいのなら、要件定義に戻って目的を変更することになります。



(3) テンプレートの利用

- 何も無いところから、新しいシステムを考案することは、たとえ開発目的を明確化しても大変困難な作業です。斬新なシステムほど非常に高い能力が必要となります。高い能力ではない人が作るとダメなシステムになってしまうのです。一般にはこのケースが多いのではないのでしょうか。
- 低い能力や短い時間を救ってくれるのが、先人の経験・ノウハウの活用です。自分が過去に経験したシステムを参考にすることはどなたもなさっているでしょう。他人の経験はなかなか活用しにくいものですが、その気になれば役立てることが出来ます。
- 様式や作成例も小さなレベルでのテンプレートだと言えます。そういうものがあると、少なくともその成果物を作成するとき

には役立ちます。

- テンプレートの最大のもは、パッケージソフトです。パッケージの持っている機能から当方の開発目的に照らし合わせて選択し、どうしても合わない部分は追加（アドオン）していけばよいのですから、それなりのものができあがります。
- 日経コンピュータ誌2006. 10. 2号「企業のITランキング」「システム企画」の部で4位に選ばれた富士フイルム殿では、グループ企業を挙げてSAP社のR/3を導入されています。
- パッケージの機能選択をされる際に使われているツール（ドイツ製）は、R/3のモジュールごとに200～300の質問があり、それに対してYes、Noで答えていくようになっています。たとえば「購買の定期支払いは1か所で行っているか」

とか、です。それに全部答えると、それに対応したパラメータセットのガイドが示されるので、それに従うとただちにプロトタイプ版のシステムができあがり、その段階のシステムを確認することができるのだそうです。

- このツールを使って、部門長、部門長スタッフ、システム部門のR/3担当等が参加するワークショップを行って基本的な要件固めをされています。一つひとつの質問に対してどうしようかと検討するので、1モジュール当たり1週間程度の日数がかかるのだそうです。たとえば「定期支払いを1か所で行うとはどういうことだろうか、それはどんなメリットがあるのだろうか、やれるのだろうか」などを議論します。ERPパッケージの初期のころよく言われた「ベストプラクティスの採用」です。

- 1週間で基本的な要件確定ができれば、かなり楽です。パッケージ利用は適切なパッケージがあれば、要件定義のかなりの圧縮が可能となります。
- 他社のシステムをモデルにして検討するのもこの類です。気をつけるべきは、常に「それでよいのか」と検討することです。その際に判断基準になるのが「システム開発目的」なのです。

6. 要件定義成功の条件

それでは最後に、どうすれば要件定義は成功するのかを整理してみます。要件定義の実施責任は、そのシステムを使う各部門にあります。自分たちの業務成果向上、事業競争力強化をどうしたら実現できるかを具体化するのが要件定義ですから、他の部門が責任部門になるということは考えられません。

成功の条件の第1は、その責任部門の参画です。それは2つに分かれます。

(1)意思決定者の参画

- システム開発目的の決定は、経営・業務の改善目標の決定です。これは該当部門の意思決定者（いわゆるプロジェクト・オーナー）の専決事項です。意思決定者が積極的に参画し、意思決定者の意向を反映したシステム開発目的でなければならないのは当然のことです。意思決定者の意思が要件定義段階で反映されていないと、いずれ後工程でその意志が表れてくるか（手戻り・修正となる）、経営の役に立たないシステムができあがってしまうこととなります。
- 従来のシステム開発の意思決定は、「システムの内容を決めるのだから」という間違った考えで「任せるよ」と部下に決定を委譲していることが多かったようです。これでは経営やマネジメントに有効なシステムが生まれることは困難です。

(2)当事者の参画

- 当事者とはシステムを利用することになる人たちのことです。現在のシステムはそのほとんどが「事業競争力強化」のためのシステムですから、他人任せ・システム部門任せでは、よいシステムができるわけがないのです。
- システム部門任せ・他人任せがあるとすると、それも「業務機械化時代の名残り」です。
- 該当部門の優れた方々の（部門のNo. 1が担当すべきだという意見も聞かれます）積極的な参画が望まれます。
- 重要なシステムであれば、部門内に検討体制を作って臨むべきです。
- この点は当然のことですが、実態はなかなかそうならないために、要件不備が頻発しているのです。

- ところが、今回ヒアリングさせていただいた中で住友電気工業殿が、以下のような形で目覚しく前進されていました。真剣に継続的に取り組んでいけば、「当事者の参画」は実現できるのです。

- システム予算はシステム部門で一括管理し、償却は部門が行う。
- システム開発の成果・効果を部門が厳しく追求される。
- したがって、いい加減なシステムを作れない。
- その結果、システム部門の協力は受けるが、要件定義書を各部門が主体的に（完成まで）作成している。

35

(3)システム部門のリーダーシップ

- 次いで、要件定義にかかわるのはシステム部門です。
- システム部門は、子会社化やアウトソーシングで体制が弱体化しているところが多くなっています。
- 職務分掌上は、システム企画や要件定義までは担当することになっていても、実態は単なる「発注窓口」に墮してしまっている場合も多いようです。
- 要件定義の実施責任は各部門にあります。業務の専門家であっても、ITの専門家ではありません。あるいはシステム開発の検討プロセスの経験が豊富なわけでもありません。
- そこで、システム部門は、ITの専門家として、検討プロセスの専門家として、各部門を支援する必要があります。あるいは、

複数部門が関連する場合は、コーディネーション役も引き受ける必要があります。

- この点に関して、今回ヒアリングさせていただいた中では、「空洞化」しつつある実態を自覚された協和発酵工業殿が素晴らしい前進をされていました。

- 開発方法論（UP手法）・開発ツールを整備され、
- 部員教育を進め、
- 開発期間の短縮、
- 手戻り減少 を実現されている。

36

(4)情報サービス業の参画

- 要件定義は、まずは各部門の責任、それを支援するのはシステム部門の役割、です。
- しかし現状では住友電気工業殿、協和発酵工業殿のようなケースは例外に近く、多くの企業で全く不十分な要件書（あるいはRFP）がまかり通っています。ある調査では、RFPを発行している企業は2割弱でした。
- 現状の情報サービス業（受注者）は、その不備な要件書を受けて、要件変更等に悩まされているのです。要件がもう少し確定することをあきらめてリスクを見て受注するか、受注を拒否する対応をしています。
- これでは問題の基本解決になりません。要件定義は本来どちらの責任だと言っているても埒があきませんので、「できないのならお手伝いしましょう」というスタンスが

- 必要です。また、その力を養成することも必要です。
- このアプローチが成功すると、発注者から喜ばれ、信頼度向上になりますし、受注者のビジネス強化、ビジネス領域拡大になります。
- 大手ITベンダ殿は一斉にその方向に動き出しています。
- このアプローチにつきましては、別稿「要件不備に挑む事例研究セミナー大盛況！！」をご参照ください。

37

(5)適切なアプローチ法の採用

- 今は世の中の動きが早く、「時は金なり」の時代です。多忙の中で効率的な検討を実施しなければなりません。
- 5. 要件定義の進め方で述べたような、なるべく試行錯誤をしないですむ、本質を突いた検討方法が必要です。

むすび

以上、要件定義および要件定義ガイドのあるべき論として私見を述べさせていただきました。異論がおありの方も多いのではないかと思います。

要件定義が現在の状況では、このレポートを契機に反論・異論取り混ぜ、真剣に議論・検討を行うことが大事である気がいたします。ご意見をいただければ幸甚でございます。 ㊦

[目次に戻る](#)

38