

コミュニケーションの事前準備とワークショップ運営のポイント

顧客も知らない 真の要求を導き出す

Text=株式会社オージス総研 正木威寛 (PMP) MASAKI, Takehiro

日頃、要求定義に携わっている皆さんは、コミュニケーションの重要性を身をもって感じていらっしゃることでしょ。顧客から確かな要求を導き出し、仕様化する作業には、コミュニケーション能力はもちろん、調整力や折衝力、良い雰囲気を作る能力など、さまざまなスキルが求められます。本稿では、要求定義に先立って事前に準備しておくべきこと、そして要求定義を実践するうえで肝となるコミュニケーションの勘所を説明します。

確かな要求を導き出すために

多くの企業は、SIerにシステム開発を依頼します。その際、企業（顧客企業）側の担当者とSIer側のSEが最初にコミュニケーションをするのが要求定義です。この要求定義のなかで、確かな要求を上手く引き出せるかどうか、その後のシステム開発の成否を握る重要な鍵になります。

顧客企業は、システムに対する「あるべき要求」を一方的に述べてくれるわけではありません。SIerは顧客企業の事業戦略を把握し、それを踏まえて「新規ビジネスの立ち上げ」や「業務の改善」といったシステム開発の目的を理解する必要があります。そのうえで、ときには客観的な立場で要求を提案したり、顧客に対して新たな要求への気づきを与えたりしながら、双方

にとっての良い関係を築き、本当に必要な要求を導き出すことが求められます。

本稿では、SIerのプロジェクトマネージャーやプロジェクトリーダーなど、プロジェクトのなかで実際に要求定義に携わっている方を対象とし、より良い要求を導き出すための準備や、顧客とのコミュニケーションにおけるポイントを解説します。

要求定義とは

要求定義や要件定義という言葉をよく使いますが、これらに対して誤解があると、筆者と皆さんとのコミュニケーションも上手くいきません。そこで、まずは用語の意味を確認しておきましょう。

国際規格であるISO 12207では、システムの開発プロセスを

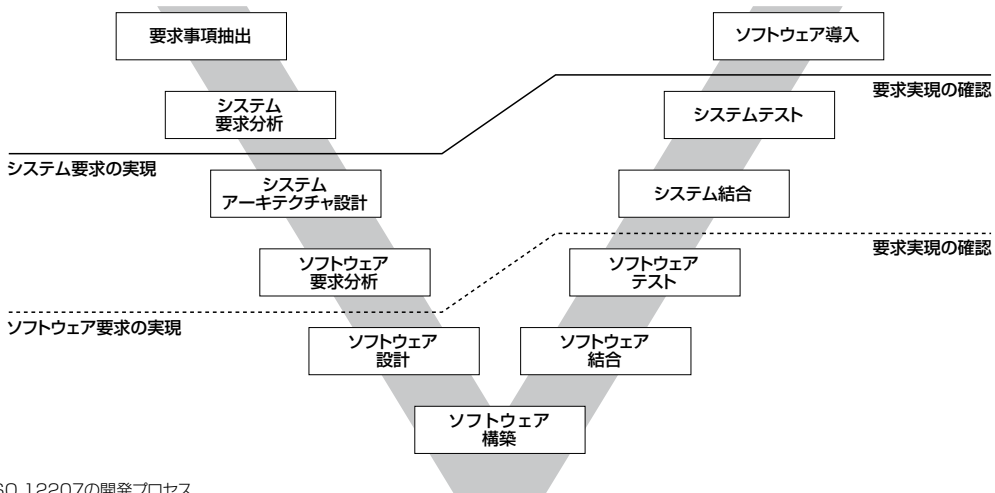


図1 ISO 12207の開発プロセス

	要求仕様書の構成	定義される内容
システム要求	システム開発の目的	システム開発の動機となった業務改善や新規ビジネスの概要。図や文章で記述することが多い。
	業務フロー一覧	システムが対象とする業務フローの一覧。業務フロー名とその概要を表形式にして記述する。
	業務フロー	業務のある目的のために連続して行われる作業を図で表したものを。業務フロー一覧の業務フローそれぞれについて作成する。フローチャートやUMLのアクティビティ図で記述することが多い。
	役割の定義	業務フローを実行するために分担された役割を定義したものを。業務フローに記載されている役割それぞれについて作成する。文章で記述することが多いが、クラス図を併用する場合もある。
	作業の定義	業務フローで行われる作業を定義したものを。業務フローに登場する作業それぞれについて作成する。文章で記述することが多いが、役割と一緒にクラス図を併用する場合もある。
	やりとりされるものの定義	作業と作業でやりとりされるものを定義したものを。業務フローでやりとりされているものそれぞれについて作成する。UMLのクラス図やER図で概念モデルとして図に表し、モデル要素を文章で記述することが多い。
	システムの非機能要求	システムのセキュリティ、信頼性、使用性、効率性、保守性、可搬性などの要求を定義したものを。
ソフトウェア要求	アクター一覧	ソフトウェアの利用者を分類したものを。アクター名とその定義を表形式にして記述することが多いが、ユースケース図を併用する場合もある。
	ソフトウェアの機能一覧	ソフトウェアがエンドユーザーへ提供する機能の一覧。機能名(ユースケース名)とその概要を表形式にして記述することが多いが、ユースケース図を併用する場合もある。
	ソフトウェアの機能要求	それぞれの機能について、アクターとの対話を定義したものを。イベントフローなどの文章や図(たとえば画面イメージ)で記述することが多い。
	ソフトウェアの非機能要求	ソフトウェアのセキュリティ、信頼性、使用性、効率性、保守性、可搬性などの要求を定義したものを。

※ システム要求仕様書とソフトウェア要求仕様書は、まとめて文書化されることもあれば、分けて文書化されることもある

図2 要求仕様書の例

図1に示すV字モデルで表しています。図の中で、V字上の前後関係にある工程^{注1}は、前の工程のアウトプットが後の工程のインプットとして連鎖していくことを示しています。たとえば、「要求事項抽出」のアウトプットは「システム要求分析」のインプットとなります。また、V字の平行(左右)関係にあるものは、右側の工程が左側の工程の実現を確認することを示しています。たとえば、「ソフトウェア設計」が正しく実現できたかどうかは、その右にある「ソフトウェアテスト」で確認されます。

3つの要求 ～業務要求、システム要求、ソフトウェア要求

図1のV字モデルでは、要求事項抽出、システム要求分析、ソフトウェア要求分析という3つの工程で「要求」という言葉が使われています。また、そのうちシステム要求分析とソフトウェア要求分析では、「要求分析」という言葉が使われています。これらについて簡単に説明しておきましょう。

まず、要求事項抽出とは、システム開発の目的(新規ビジネスの立ち上げや業務の改善など)の核となる要求を顧客から収集し、それらについて顧客と合意することを指します。このような、ITの範囲を決める前の業務上の要求を「業務要求」と呼びます。

要求分析とは、ある手法に基づいて要求を分析し、それを

仕様化することを指します。また、システム要求分析の「システム」とは、人手やIT(ソフトウェアやハードウェア)、設備といったシステムの構成要素が組み合わさった、ある目的を達成する集合を指します。たとえばコンビニエンスストアでは、販売業務のために次のようなものが組み合わさってシステムを構成します。すなわち、①棚に商品を並べる店員の手作業、②商品を売り上げる店員とそれを支援するレジ、③店長に代わって1日の売り上げを集計するバックヤードのコンピュータ、④商品を並べるための棚などの設備、といったものです。このように、システム要求とは「ある目的を達成するためのシステムへの要求」を指し、その一構成要素であるレジやバックヤードのコンピュータに搭載するソフトウェアへの要求を「ソフトウェア要求」と言います。

再び図1をご覧ください。システム要求分析の次の工程である「システムアーキテクチャ設計」とは、システムの構成要素の組み合わせを設計することです。この組み合わせによって、ソフトウェアが支援や代行する範囲が異なります^{注2}。したがって、厳密に言えば、システムアーキテクチャ設計がある程度具体的にならないと、ソフトウェア要求分析はできないと言えます。ただ

注1 ISO 12207ではV字モデルの各要素を「サブプロセス」と呼びますが、本稿では「工程」と表記します。

注2 販売業務でシステムの構成要素の組み合わせを変えた例として、店員ではなく買い物客が自ら売り上げる「セルフレジ」といったものがあります。これは、実際に大手スーパーで使われています。

し、実際の開発では、両者を「要求分析」というかたちでひとまとめにして、並行して行うことも少なくありません。

要求定義と要求仕様書

本題の要求定義に話を戻しましょう。

要求定義とは、前述のシステム要求分析やソフトウェア要求分析によって「仕様化された要求」のことを指します。あるいは、要求分析と同義語として使われ、「要求を仕様化する行為」を指すこともあります。本稿ではこれらを区別するために、前者、つまり仕様化された要求のことを要求定義と呼び、後者、つまり要求を仕様化する行為を要求分析と呼びます。

では、仕様化された要求である要求定義は、どのような作業成果物として存在するのでしょうか。それは、前頁図2に示す要求仕様書というドキュメントに、モデルや記述としてまとめられます。

要求分析の進め方

前述したように、要求を仕様化することを要求分析と呼びます。この要件分析を上手く進めるには、事前準備も必要です。し、顧客から要求を引き出すためのテクニックも必要です。ここでは、①要求分析に取りかかる前に認識しておくべきこと、②要求分析に先立って行っておくべきこと、③顧客から要求を引き出すためのテクニックである「ワークショップ(検討会)」の運営、について説明します。

要求分析、その前に

まずは、要求分析の前にきちんと認識しておくべきことを紹介します。ここで述べる事柄は、顧客とコミュニケーションする前提知識となるものです。

契約上のリスクを認識する

システム開発が顧客企業とSIerとのビジネスである以上、SIerは要求分析に入る前に、プロジェクトや収支上のリスクを認識しておく必要があります。ここで言うリスクとは、契約に関するものです。

図1に示したV字モデルをもう一度振り返ってみてください。システム要求分析やソフトウェア要求分析は、これから実現す

るもの(すなわちシステムやソフトウェアの開発)との境界に位置します。したがって、要求分析には次のような契約パターンがあります。

- ① システム開発の前に、提案というかたちでSIerが無償で行う
- ② システム開発に含めて、SIerが一括請負契約で行う
- ③ システム開発の前に、SIerが要求分析だけを準委任契約で行う

1つ目のパターンは、大型案件で受注を見越して行われることがあります。しかし、顧客ときちんと合意できるような素晴らしい要求仕様が出来たとしても、要求定義によって明らかになった開発費用・期間などのシステム開発の規模感や、それに対する達成可能な効果を基に、顧客が「開発を中止する」という結論を出した場合、SIerが働き損になるというリスクが伴います。

2つ目のパターンは、システム開発の範囲を示す要求定義ができていないにもかかわらず、開発の契約をしてしまうことになります。したがって、顧客とSIerとの間で、システムに対する規模感の予想が大幅に食い違い、赤字プロジェクトになるリスクがあります。ところが、これは日本では伝統的に多い契約形態で、システム開発のための時間やコストを要求分析のために食いつぶしてしまうケースも見られます。

3つ目の準委任契約(S&S契約)は、最近増えてきた形態で、システム開発の契約に先行し、要求分析の作業(作業時間など)に応じて対価を得る契約です。したがって、ここに挙げた3つのパターンの中で、収支上のリスクが最も少ないものだと言えます。たとえば、「要求分析を延長し、保留している要求についてもっとディスカッションしたい」という顧客からのリクエストに対しても、収支上の問題が少なく応えられます。

少しでも多く相手のことを知る

現行システムの再構築のように具体性を帯びたシステム開発の案件であっても、事前の情報収集は不可欠です。具体的には、要求分析に取り掛かる前に、顧客企業が打ち出している事業戦略や業務内容といった情報を可能な限り収集しておきましょう。たとえば、顧客企業のホームページには、業務内容の他に「企業情報」や「採用情報」が掲載してあるはず。企業情報には、ビジネスを今後どのように展開するのかという事業戦略や、経営者からのメッセージ、企業の沿革などが記載されており、その企業のカルチャーを感じ取れます。筆者の経験では、特に中途採用情報には、今期、来期といった短期計

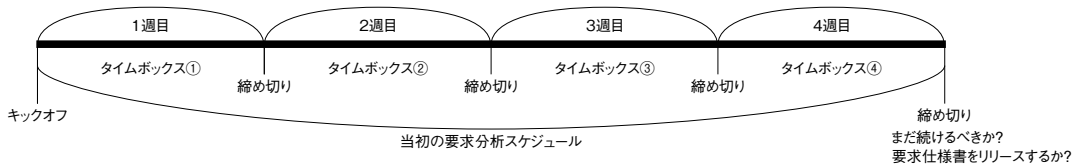


図3 タイムボックス手法

画が具体的に現れていることが多いようです。収集した情報と、システム開発の対象となる業務との関係の有無を問わず、「少しでも多く相手企業を知る」ということは、要求分析を進めるうえでコミュニケーションに少なからず役立つものです。

すべての要求を見つけることは不可能だと認識する

どんなに時間をかけて要求分析を行ったとしても、明らかになる要求と、明らかにならない要求、すなわち暗黙の要求とがあります。暗黙の要求には、この先の設計で明らかになるユーザーインターフェイスの詳細な振る舞いや、ユーザーが実際に使ってみて初めて気づく些細なビジネスルールなど、さまざまなものがあります。

要求分析の段階で、このような暗黙の要求を見つけ損ねてしまうことは、ある程度は仕方ありません。できるだけ要求を見つけて仕様化する努力をすることは必要ですが、プロジェクトの判断の切れ目として、要求分析に締め切りを設けることも必要です。さもなければ、いつまでも、極端に言えば開発期間よりも要求分析に時間をかけてしまうことさえあります。そして何よりも、要求分析のメンバーは終わりの見えない作業に段々と関心が薄れ、モチベーションを失っていきます。

タイムボックス手法を活用する

上で述べたようなことを避けるために、筆者はアジャイル開発で用いられているタイムボックス手法を要求分析を取り入れています。タイムボックス手法では、図3のように、全体の締め切りだけではなく、短期間(タイムボックス)の締め切り、たとえば1、2週間程度の期間で締め切りを設けます。そして、それぞれの締め切りに間に合わせるべき要求分析の範囲(中間マイルストーン)を設定して作業を進めます。1つのタイムボックスが終わると、中間マイルストーンに到達したかを確認し、必要であれば次以降のスケジュールを見直します。ただし、基本的には「もっとディスカッションすれば良い案が出るかもしれない要求定義の内容」よりも、締め切りのほうを厳守します。タイムボックスの目安は、利害関係者が集まる要求分析のワークショップの

間隔に合わせ、ワークショップ2~4回につき1回とするのがよいでしょう。ワークショップについては後ほど説明します。

顧客企業のパートナーを得る

要求定義には、システムのユーザーである業務部門のニーズ、運用担当者のニーズ、連携する他システムの技術的な制約などを反映させる必要があります。したがって、ある日突然、外部のSEである自分が要求分析を担当することになったとしても、自分1人だけでは不可能です。

要求分析を進めるには、次頁図4のような役割分担が必要です。よって、①顧客企業の業務部門や運用部門の担当者の招集、②そうした関係者への作業依頼の働きかけ、③既存システムの仕様の調査を手配してくれるお客さま、たとえば情報システム部門(以下、情シス部門)やシステム企画部門^{注3}のカウンターパートナーとなる要求分析者の協力、などが欠かせません。良いカウンターパートナーとは、SIerである自分たちの強み・弱みを理解してくれ、二人三脚で作業を進められる方のことです。決して役職が高い人とはかぎりません。

要求分析ワークショップの運営

次に、要求分析の中心となる作業であるワークショップについて説明します。これは、関係者が集まって要求を洗い出したり、仕様についてディスカッションしたりする検討会のことです。本稿では、これを「要求分析ワークショップ」と呼ぶことにします。

このワークショップには、次頁図4のように、システム開発に関係するさまざまな立場の方が参加します。推進役である要求分析者にとっては、ワークショップを取り仕切る力やコミュニケーションの力が試される場でもあります。ここでは、要求分析ワークショップを運営するうえで、筆者が実践している基本的な事柄について説明します。

.....
注3 企業によって、情シス部門やシステム企画部門が片方だけ、あるいは両方とも存在することがあります。本稿では両者をまとめて情シス部門と表記します。

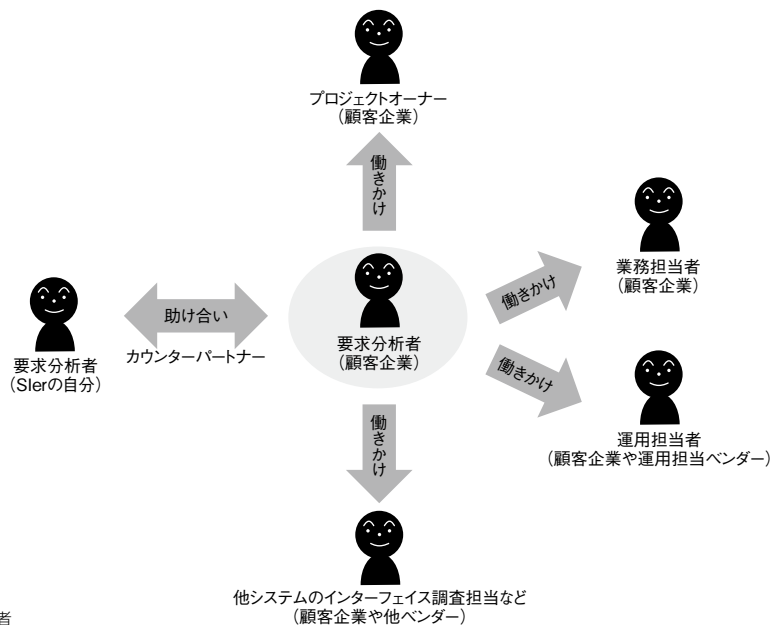


図4 要求分析の参加者

アジェンダを用意する

アジェンダとは、ワークショップのテーマを列記したものです。当日に何をディスカッションするのか、そのテーマをA4用紙で1枚程度に書き込んでいきます。こんな紙きれ1枚でも、ディスカッションが横道に逸れた場合に、参加者に「逸れたこと」を認識してもらい、話を本題に引き戻すのにとても威力を発揮します。できれば、開催前にメールなどで事前に配布しておきましょう。

時計を見ながら進める

ワークショップを仕切るうえで、時計は欠かせません。けれども、腕時計や携帯電話をあからさまに覗き込みながら進めると、ディスカッションに気持ちが入っていないように見られてしまい、印象が良くありません。腕時計を外して机の上に置き、参加者にわからないようにさりげなく見ながら、アジェンダの予定と調整するようにしましょう。

ゆっくり丁寧に話す

人は、自分の思いを伝えようと一生懸命になると、段々と早口になってしまいがちです。早口になると、参加者の中には話について来れない人も出てきてしまい、一部の人だけのディスカッションになり、参加者全員の理解が得られなくなってしまいます。まず、モデレーターである要求分析者は、ゆっくり丁寧に話すことを心がけましょう。そして、もしも誰かが早口で話し始め、さらに場全体が早口になり出したら、その内容を自分が確認す

るように(じつは皆が理解できるように)途中でさりげなく割り込み、ゆっくり復唱してペースを戻しましょう。

多少の脱線に耳を貸す

アジェンダで挙げたテーマに沿ってディスカッションをしていると、いつの間にか参加者の発言が他の話へ脱線することがあるでしょう。ですが、一概に脱線とは言えないこともあります。たとえば、筆者はこんな経験をしました。ある時、画面や帳票のデザイン、ファイル形式にしきりに関心を示す業務担当者がいたのですが、よくよく聞いてみると、「他社と取り引きするうえで、形式が決まっているので業務担当者としても要求しておきたい」といった意見でした。また別のケースでは、業務担当者が「あつ、そう言えばAさんのところでも同じような問題を言っていたな」と気づき、業務改善という部門単位での共通目標の観点から、フェーズ2(次フェーズ)として範囲を広げ、さらに要求分析の検討を行ったこともありました。

このように、脱線と思えるようなことでも、場合によっては新たな要求の気づき(発見)となることがあるので、雑談なのか発見なのかを見極められるまで、多少の脱線には耳を貸しましょう。雑談であれば、タイミングを見てアジェンダを示し、いま考えるべきことかどうか示唆を与えるようにしてください。

言葉をすぐに絵にする

まだ何も文書にしていないことについて、言葉のやり取りだけ

でディスカッションするよりも、言葉を絵に描いて見せることは効果的です。そうすると、参加者たちが要求を早く、正しく理解し合えることがあります。実際に、参加者それぞれが異なる要求を同じ要求のつもりでディスカッションしていたのが、絵に描いたら議論が深まったり、要求に矛盾があることに気づくといったこともあります。ホワイトボードにUMLなどの決まった様式で書いてもよいのですが、業務担当者が理解できなかつたり、UMLの書き方の善し悪しに話が脱線したりすることもあります。筆者は、業務担当者の方が参加するワークショップで描く絵は、様式にこだわらずにマンガで描いておき、要求定義として要求仕様書に記載する際に、UMLや文章できちんと仕様化するようになっています。

得意なことに夢中にならない

要求分析に慣れていないと、どうしても要求分析者が自ら技術的な話題に夢中になりがちです。SEとしてたくさん話せるネタがあるので、要求分析者という立場を忘れてデータベースのチューニングやプログラミングなどについてあれこれ語ってしまうのです。常に要求とアーキテクチャの区別を心がけ、冷静にワークショップを仕切ることを忘れないでください。

要求分析の実践

ここからは、前述した要求分析ワークショップを実際に運営し、顧客から要求を引き出していくためのポイントを、要求分析の手順に沿って紹介します。具体的には、次の4つの局面に分けて説明します。

- ①要求分析ワークショップのキックオフ
- ②序盤の要求分析ワークショップ
- ③中盤の要求分析ワークショップ
- ④終盤の要求分析ワークショップ

①要求分析ワークショップのキックオフ

キックオフは、**図1**に示した要求事項抽出を締めくくり、これから始まる要求分析という共同作業について、関係者が理解して共有する場です。したがって、できるだけプロジェクトオーナーも含め、プロジェクトの利害関係者全員に参加してもらうとよいでしょう。

いきなり“形”で進めない

キックオフは、参加者同士が初めて会う機会であることが多いと思います。また、参加者はこれから始まる要求分析に不慣れだったり、多くの人が「今から何が始まるのだろう……」と不安を抱いていたりします。このような場で、「あなたの業務で感じている問題は何ですか?」といった定型的なインタビューシートを使って、いきなり進めようとする要求分析者もいます。しかし、このような進め方はプロフェッショナルな良い印象を与えることもあります。一方で警察の取り調べのような緊張した場を作ってしまうこともあります。それよりも、参加者それぞれのシステムに対する期待や、システムが実現されたときのイメージをざっばらんに話し合い、これから行っていく要求分析の目的についての合意形成に努めましょう。

業務を理解する態度を見せる

「自分は言われた要求は文書化します。お客さまの業務のことは、そちらで要求を取りまとめてください」といったような、お客さまの業務を理解することを放棄するような態度は避けましょう。このような態度は、参加者を「ただ集められただけで何をすればよいかわからない」という不安な状態にしてしまいますし、このような場で要求を引き出すことなど到底できっこありません。要求分析者が、キックオフ時点ではお客さまの業務のことがまったくわからなくても、一生懸命勉強していく姿勢を見れば親切に教えてくれるものです。自分は「要求定義屋」として業務をできるだけ理解し、責任を持ってワークショップを導き、仕様化していくことを表明しましょう。そうすれば、お互いに教えられる対等の信頼関係を築いていけるものです。

みんなの役割を再確認する

システム開発の目的に対する、参加者それぞれの役割をキックオフの時に再確認しましょう。通常は、事前に参加者の上司から本人たちへの説明があるものですが、なかには日々忙しい業務をこなしており、突然上司から参加を命じられ、システム開発の目的やワークショップでの自分の役割を十分に理解していないことがあります。目的は何なのか、誰がどの役割なのか、どの業務はどの人がよく知っているのか、などを確認し合い、目的や役割に対して共通認識を持てるようにしましょう。

要求分析作業全体の流れを示す

参加者の不安をさらに軽減するために、これから始まる要

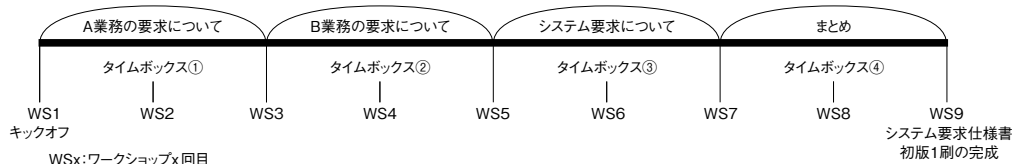


図5 マスタースケジュールの例

求分析の作業全体の流れを示し、参加者が合意することが必要です。合意すべき主な事柄には次のようなものがあります。

- ① 要求仕様書の最初(初版1刷)のリリース時期
- ② 要求分析の範囲(ディスカッションのテーマ)
- ③ ディスカッションのテーマに関連する要求仕様書の各パート(たたき台)の作成分担
- ④ 分担した各パートのディスカッションのスケジュール

たとえば、タイムボックス手法が進めるのであれば、図5のようなマスタースケジュールになります。この例では、AとBの2つの業務を対象とし、業務要求に2回のタイムボックスを割り当てています。次に、それらのシステム要求に1回、全体のまとめとして1回のタイムボックスを割り当てています。最後のまとめ以外のタイムボックスは、たたき台をベースにディスカッションする初回のワークショップ(たとえばWS1)、その修正版についての2回目のワークショップ(たとえばWS2)、そして締め切りである最終確認(たとえばWS3で、B業務の初回のワークショップでもある)といった進め方になります。このようなタイムボックス手法のスケジュールを不慣れた参加者が見ると、「締め切りばかりで辛そう、プレッシャーだなあ」というような誤解を与えてしまうことがあります。確かに、各タイムボックスで締め切りを設けていますし、それぞれのパートの締め切りまでにそれぞれのパートの最初のバージョンができるのは間違いありません。けれども、その後のディスカッション(たとえばWS4)で完成済みのパート(たとえばA業務)を再検討するほうがよいと判断した場合は、それ以降のタイムボックス(たとえばWS5)で完成済みのパートについても、再検討するように計画を変更することもできます。このように、タイムボックス手法は序盤から締め切りがあり、担当者を少し締め切りに追い込んでしまうかもしれませんが、一方でプロジェクトの変化に対して柔軟性のあるスケジューリング手法であることも忘れずに説明してください。

②序盤の要求分析ワークショップ

要求分析ワークショップの序盤では、システム開発の目的で

ある業務要求に焦点を当て、ディスカッションしながら深掘りしていきます。この時には、業務担当者(あるいは要求分析者)が作成した業務フローなど、対象業務を見える化(文書化)した「たたき台」をベースに、参加者は業務を理解し、新規ビジネスや業務改善といったシステム開発の目的と照らし合わせながらディスカッションを重ねます。

見えないアプリケーションを発掘する

ひと昔前からEUC(End User Computing)というキーワードの下に、業務部門でExcelやAccessが当たり前のように使われるようになりました。企業によっては、業務部門にExcelやAccessのプログラミングができる人がいて、情シス部門も把握していないExcelやAccessのアプリケーションが業務をサポートしていることが少なくありません。このようなアプリケーションが作られた背景には、もはや現行システムが現行業務に対して機能不足となっていることがあります。したがって、システム再構築ではたんに現行システムの機能を再検討するのではなく、このようなアプリケーションがサポートしている作業もひっくり返して業務全体を俯瞰し、業務フローやシステム化の範囲をディスカッションする必要があります。

いろいろな要求を管理する

業務要求に焦点を当ててディスカッションしているのに、いつの間にかソフトウェアの使い勝手の要求など、いまディスカッションすべきでないことも出てくるものです。特にこの序盤の要求分析ワークショップでは、日頃から業務やITに対して問題意識を持っている人は、話したくてウズウズしています。このような人の発言を排除したり聞き流したりすると、発言者には大きなストレスが溜まります。このようなストレスは、ときに爆発してワークショップに良くない空気を作るので、そうならないように受け皿を用意しましょう。筆者は、受け皿として表6のようなニーズ一覧と呼んでいる表に書き留めるようにしています。ここに記載するニーズとは、システム要求なのか、ソフトウェア要求なのか、開発範囲外の要求なのか、じつは要求ではないのか、などにはこだわら

No	大分類	小分類	ニーズ	登録日	要求者	ステータス	状況や経緯
1	仕入業務	発注	18時までしか発注をできないが、空いている夜間にも発注ができるようにしてほしい	2007/9/7	仕入担当 猪木	クローズ	9/7議事 / 情シスへ調査依頼(課題ID 1) 9/14議事 / 現行システムがEDIでデータ転送する時刻に合わせてエントリーをできない実装であった 9/21議事 / 要求仕様書へ反映
2	仕入業務	検収	検収時の発注商品との突き合わせ作業の効率が悪い。もっと早く検収ができるようにしてほしい	2007/9/7	仕入担当 青木	検討中	9/7議事 / 現行システムの使い勝手が悪い
3	仕入業務	仕入管理	発注残の照会で、発注残数が不正確で二重に発注してしまうことがある。もっと正確にほしい	2007/9/7	仕入担当 猪木	検討中	9/7議事 / 状況がよくわからないので業務部署へ再度ヒアリング(課題ID 2)
4	販売業務	商品管理	最近では価格変更が多いので棚札の変更が面倒。棚札を電子化してほしい	2007/9/7	フロア担当 馬場	保留	9/7議事 / 実現可能性を調査(課題ID 3) 9/21議事 / X社の棚で実現可能。棚の導入費用がかかるので別途予算を確保する必要がある 9/28議事 / フェーズ2として別途予算が取れるまで保留

図6 ニーズ一覧

No	大分類	小分類	課題	登録日	ステータス	担当者	完了予定日	状況や経緯
1	調査	現行システム	現行システムが18時までしか発注をできない理由と、夜間に発注の実現可能性の調査	2007/9/7	クローズ	情シス 正木	2007/9/14	9/10調査 / 現行システムがEDIでデータ転送する時刻に合わせてエントリーをできない実装である 9/13調査 / 特にエントリーだけなら制限せずに24時間できるように実現できる
2	調査	現行業務	発注残の照会で、発注残数が不正確といっている状況がよくわからないので、業務部署の報告者へ再度ヒアリングする	2007/9/7	検討中	業務担当 高田	2007/9/21	
3	調査	新技術	液晶表示などで、コンピュータと連動して表示が変わる電子化した棚札は実現可能か?	2007/9/7	クローズ	情シス 田村	2007/9/21	9/12調査 / X社で液晶表示のついた棚を販売していた 9/20調査 / X社営業とコンタクト。LAN接続で提供されるjavaのライブラリから表示の変更が可能。1台100万円であった

図7 課題一覧

ず、求められていることすべてを指します。ニーズ一覧に書き留めたニーズは、ワークショップや要求分析者によって分析され、開発範囲の要求であれば要求仕様書に反映します。それまでは、開発範囲であろうがなかろうがニーズ一覧でステータス管理をします。このような管理をすることを「要求管理」と言い、専用の要求管理ソフトもありますが、Excelなどでも十分に役目を果たせます。参加者の発言が要求管理されている状態というのは、参加者に安心感を与え、ディスカッションを活発にし、要求分析作業のスケジュール管理などにもとても役立ちます。

議事録を偏重しない

ワークショップが終わると、議事録の作成に多くの時間をかける人がいます。ワークショップを何度も何度も行くと、参加者が以前言った発言や要求仕様が再び話題になることがあります。こんな時にたくさんの議事録をひっくり返すのは効率が良くありませんし、なにより、言った・言わないといったコミュニケーション上のトラブルを引き起こしかねません。こうならないように、言った・言わないの証拠として、議事録を残すというスタンスではなく、公式に共有しているニーズ一覧を中心に要求分析を進め、

議事録はニーズ一覧に記載されないことの「メモ」であるというスタンスをお勧めします。ワークショップ後、あるいはその開催中に、ワークショップで挙げられた要求や意見は速やかにニーズ一覧へ記録し、常に参加者で要求を共有していきましょう。

課題を管理する

要求についてディスカッションをしていくと、次のような調査や確認が必要となることがあります。

- 参加していない他の業務担当者の意見収集
- 法的な要件についての法務や顧問弁護士への確認
- 連携する他システムの外部インターフェイス仕様の調査
- 新技術の実用性の調査

このようなワークショップ以外での作業が必要であることがわかったら、これも議事録に書くのではなく、**図7**のような課題一覧(ToDoリスト)として参加者で共有していきましょう。ニーズに関連する課題であれば、ニーズ一覧の対応するニーズに「必要となった調査や確認の概要」と「課題一覧に追加した課題のID」を記載しておきます。

③中盤の要求分析ワークショップ

中盤の要求分析ワークショップでは、序盤で掘り下げた業務要求を基に、業務担当者あるいは要求分析者がシステム開発に必要な情報を収集し、システム要求に展開していきます。この段階ぐらいいから、システムの品質面や技術的な制約となる要求定義のために、現行システムや連携する他システムの情報が必要になってきます。また、ニーズ一覧に集まったニーズの整理や課題の状況確認に多くの力を割く必要があります。ここでのポイントは以下のとおりです。

ワークショップの状況を共有する

2回目以降のワークショップ、少なくともタイムボックスの締め切りでは、アジェンダに挙げる作業事項として、次のような要求分析ワークショップの状況確認を載せましょう。

(1) ニーズ一覧の状況

1. 追加されたニーズ
2. 引き続きディスカッションするニーズ
3. 要求仕様書に記載されクローズしたニーズ
4. 開発の範囲外や保留にされたニーズ

(2) 課題一覧の状況

1. 追加された課題
2. まだ調査中の課題
3. 前回調査結果の報告があり、クローズした課題
4. 関連するニーズが開発範囲から外れ、調査の必要性がなくなった課題

3) スケジュールの状況

1. 更新済みの最新のスケジュール
2. 現在、マイルストーンのどこまで来たか(実績)
3. 次のマイルストーンは何か

4) ニーズ一覧から反映された要求仕様書のドラフト版

ニーズ一覧にせよ課題一覧にせよ、一覧性がありますし、追加や消し込み(クローズや保留)で要求分析の状況がよくわかります。このような、アジャイル開発のダウンチャートのように消し込むということや、要求仕様書の空欄が埋まっていくことには、参加者のモチベーションを高める効果があります。

また、業務が多忙で、だんだんと不参加者が出てきていると

感じたら、タイムボックスの締め切りのワークショップで、短い時間でもかまわないので一度全員に参加してもらい、ワークショップの状況確認だけでもしてもらいましょう。これは、ワークショップを再活性化するのに有効です。

現行システムの発想を捨てる

システム再構築の場合、現行システムのアーキテクチャや設計が、業務上の不便さを招いていることがあります。たとえば、現行システムではバッチ処理のため、ある作業を行ううえで必要な情報の反映が翌日になってしまい、業務効率を落としている場合、新システムではバッチ処理をやめてリアルタイム処理にすることによって解決できます。このようなケースでは、現行システムがバッチ処理だから、新システムもバッチ処理で行うことを前提として業務上の問題解決を考えてしまいがちです。しかし、新システムではそのような制限を受けずにリアルタイムでも実現可能かもしれません。まずは、現行のアーキテクチャの発想を捨て、業務上の問題を解決できる可能性がないか考えてみましょう。

業務上の頻度や量を収集する

業務フローが実行される頻度や作業の頻度は、システム要求のトランザクション量と関連します。業務によって扱われる伝票などは、システム要求のデータ容量と関連します。業務フローに現れる作業担当者の実際の人数は、同時利用者数やアカウント数と関連し、時にはパッケージのライセンス料と関連します。可用性、使用性、効率性といったシステム要求の非機能要求は、いきなりシステム要求として収集するのではなく、業務担当者からこのような業務上の計数(メトリクス)として収集することにより、上手く導き出すことができます。

他システムとの連携に注意する

現行システムが連携する他のシステムがある時に注意しなければいけないのが、連携のための外部インターフェイス仕様書がきちんと最新の状態の文書として存在するかどうかです。さらには、「その外部インターフェイス仕様書は変更可能か」という点も重要です。レガシーなシステムと連携している場合、そのレガシーシステムは仕様書もアップデートされておらず、ブラックボックス化しているかもしれません。変更不能であれば、システム要求で技術的な制約として挙げておく必要がありますし、不明なことをリスクとして挙げ、現行システムとのインターフェイス

仕様を調査してもらう必要があります。現行システムに詳しい情シス部門の方がいればまだマシで、連携先のシステムの保守担当会社へ調査を依頼することが必要になるかもしれません。このようなケースでは、要求分析の当初の参加者だけでなく、必要に応じて関連するシステムの担当者も要求分析へ巻き込んでいくことが必要になります。

技術的な話題をフォローする

この局面では、イメージをすり合わせるために、これまで述べたような技術的な内容へ話が及ぶことがあります。WebサービスやAjaxのような、業務担当者にはさっぱりわからない話も出てしまうことがあります。このようなときには、とろとろで何を話しているのかを業務担当者にもわかるように噛み砕いて丁寧に説明し、取り残されてしまったような気持ちにさせないように気遣いしましょう。もし長い話になるようなら、別の機会に情シス部門の人とのディスカッションの場を設けることを提案すべきです。

④終盤の要求分析ワークショップ

システム開発がプロジェクトとして成立するように、要求定義を取捨選択するのが終盤のワークショップです。この段階から概算見積もりや実現可能性の検証が必要になってきます。ここでのポイントは以下のとおりです。

開発範囲を再確認し、共有する

業務フローや作業内容を要求定義として明らかにし、非機能要求やシステム連携など関連する技術的な制約もわかってきたら、業務フロー上のITが支援や代行する開発範囲を、今度はプロジェクトの観点で再確認する必要があります。ITが支援や代行する範囲によって、開発費用や開発期間は2倍、3倍の違いが出てしまい、プロジェクトとして成立しなくなることがあるからです。したがって、ワークショップの終盤では、それまでに挙がった要求定義を、業務改善といったそもそもの開発目的とQCD (Quality: 品質、Cost: 開発費用やランニングコスト、Delivery: 納期)を踏まえ、次のような選別にかかる必要があります。

- 本当にそのまま要求として要求仕様書に載せるのか
- 要求定義の要求レベルを落として要求仕様書に載せるのか
- 開発範囲から外す(要求仕様書から削除する)のか

筆者の経験では、ニーズ一覧のすべての要求を満たすと、大抵の場合は過剰品質になります。実際、24時間365日使えることを業務担当者が要求していても、「そんなに開発費用がかかるなら、そこまでの可用性はいらない」とか、「業務がなんとかできればいいから、他の要求を外してでも、ビジネスの開始に間に合うよう納期を厳守してほしい」とか、システム開発の目的と照らし合わせて参加者に問いかけてみると、ふるい落とせる要求は多いものです。

さらにこの時、システム再構築を前提とする要求定義でない場合は、新システムで実現するには費用対効果が思わしくないと予想されるのであれば、要求に対して十分ではないとしても、現行システムの保守である程度の目的を達成することに方針を変更することもあります。

このようにワークショップの終盤では、システム開発の目的とQCDが上手くバランスした開発範囲の着地点を見つけることにディスカッションの時間が割かれます。この間、ニーズ一覧とともに要求仕様書のドラフト版が頻繁に更新されていきます。

要求しないことを共有する

終盤のワークショップでも、これまでどおり、①ニーズ一覧、②課題一覧、③スケジュール、④開発範囲となる要求仕様書、などを使ってワークショップの状況確認を行います。ただし、特に最後のワークショップでは、参加者が確実に合意しておくべきことがあります。その1つが、ニーズ一覧に載せた要求の中で、前述の要求仕様の確認で「開発範囲から外す」と決めたものの確認です。参加者は、これまで「要求する(要求仕様書に記載する)」という思考でいたので、逆に「要求しない」ニーズの集まりを目の当たりにすると、合意しがたい心境になることもあるからです。ニーズ一覧上で保留にされ、要求仕様書から外した1つ1つの要求について「要求しない」ことを共有しましょう。

放ったらかしの課題を共有する

上で述べた確認すべきことの2つ目が、課題一覧に載せた調査などで「やり残した課題」を共有することです。要求仕様書から外された要求と関係する調査や、要求仕様書に記載された要求と関係があっても、調査結果が報告されていない場合は、課題のステータスは「調査中」のままです。このような「調査中」の課題が予定どおりに調査中であればよいのですが、往々にして「放ったらかし」になることがあります。もしもこの調査内容が、連携する他システムとのインターフェイス仕様の調

査のように、システム開発が必要とする時期までに入手できなければ、開発そのものが頓挫するようなリスクの高いものである場合、要求分析の参加者は現時点でそのような重大なリスクがあることを認識すべきです。課題一覧上で「調査中」の課題1つ1つについて「放ったらかしになっていないか」「放ったらかしにしておく(リスクを受容する)のか」「何か手を打つ(リスクを軽減・迂回・転嫁する)のか^{注4}」を共有しましょう。そして残ったリスクは、要求仕様書とともにシステム開発の部隊へ伝えるべきです。

みんなで創り上げたものの共有する

最後のワークショップでぜひ実践していただきたいのが、「要求仕様書 初版1刷」を印刷し、参加者全員で“みんなで創り上げたもの”を一緒に共有することです。それまで、何度も何度も皆でディスカッションを繰り返し、知恵を絞り、要求定義として仕上げた公式な成果が「要求仕様書 初版1刷」だからです。「要求仕様書はWordでメールに添付して送っているからいいや」と思わず、皆さんと一緒に共有してください。筆者の経験では、たとえ1か月程度の短期間の要求分析であっても、これを行うといつも感慨深いものがあります。

要求定義はアップデートされていく

以上、本稿では、要求分析で顧客とさまざまなコミュニケーションをするなかで、要求を上手く導き出し、要件定義として要求仕様書の完成にまでこぎ着けるために、筆者が日頃実践しているポイントについて解説しました。ただ、この後に続くシステム開発で注意していただきたいことがあります。最後にそのことを述べて本稿を締め括りたいと思います。

前述したように、要求には「要求分析中に要求仕様書に要求定義として明らかにされる要求」と「まだ明らかにされていない暗黙の要求」、さらに「ユーザーが開発途中のアプリケーションを触ってみて、『少し変えて欲しい』というような変化する要求」もあります。実際のシステム開発では、このような些細な要求の追加・変更が開発中に起きるものですが、システム開

発が始まってしまうと、要求定義を放ったらかしにしてしまうプロジェクトも少なくありません。しかし要求定義は、システムテストやユーザーの受け入れテスト(図1の「ソフトウェア導入」)の根拠となるものです。このときに「要求定義どおりのシステムか」「システム開発の目的を達成したか」を評価する根拠が不明瞭では、コミュニケーション上のトラブルが起きることは明らかです。システム開発中も、本稿で紹介したニーズ一覧のような要求管理の手続きを確立しておき、忘れずに要求仕様書をアップデートしていきましょう。✍

参考文献

「ISO/IEC 12207:1995、修正案1:2002、修正案2:2004
Information technology - Software life cycle process」

「プロジェクト知識体系ガイド 第3版」
(著者・刊行:PMI[Project Management Institute])

PROFILE

正木威寛(まさき たけひろ)

システムエンジニアやプロジェクトマネジャーを経験し、現在は企業のシステム開発における調達プロセスや開発プロセスのコンサルティングに従事。ユーザー企業とSierとの両方が幸せになれるような健全な開発プロジェクトを実現するために、日夜プロセス改善に取り組んでいる。与えられた制約の中で、最大限の知恵を絞り、良いアイデアを出すことが求められるプロセスコンサルティングの仕事は自分に合っていると自認している。

注4 プロジェクトマネジメントの標準であるPMBOKでは、リスクの対応策を以下の4つに分類しています。

- 受容:リスクを認識しているが、あえて何もしない
- 軽減:リスクの影響を少なくする
- 迂回:リスクが起きないようにする
- 転嫁:リスクが起きた時の責任をプロジェクト以外にする