

# マイルストーン計画と断面評価による段階的な要件定義の進め方

## How to Proceed the Graded Requirement Definition by a Milestone Plan and the Sectional Evaluation

宮 脇 亨

**要 約** 要件定義に起因する問題は、システム開発全体に影響を及ぼす。要件定義が終わらない、要件が膨れ上がってしまう、主要な要件の評価が不十分、といった問題は、コストオーバーの原因と強く関係する。要件定義工程の主要な活動は、スコープマネジメントである。システム化要件のベースラインを確立する役割を担う。

本論は、ベースラインの確立に向けたスコープマネジメントの実施方法を、プロセスと評価の観点から示す。具体的には、マイルストーン計画と進捗の断面を捉える評価方法によって段階的に要件を固めながら、内容の実現性と全体として要件の整合性に着目して進め、リスクの可視化と評価を行うことを提唱する。

**Abstract** The problem due to definition of requirements has an influence on the whole system development. Problems, such as the never-ending requirement definition, a ballooning requirements, and insufficient evaluation of major requirements are strongly-related to the cause of a cost overrun. The requirement definition process undertakes a role to establish the baseline of systematization requirements. And its main activity is the scope management.

This paper describes how to execute the scope management for the establishment of the baseline from the viewpoint of process and evaluation. Specifically, while hardening the requirements by an evaluation method that captures a section of a milestone plan and progress, proceed with the requirement definition by focusing attention on overall consistency and possibility of requirements. And also we suggest that visualization and evaluation of the risk should be performed.

### 1. はじめに

要件定義がシステム開発に与える影響は大きい。コストオーバーを引き起こす原因の多くがこの工程にあるとも指摘されている<sup>[1]</sup>。そのため、要求分析の方法や整理の仕方、ヒアリング技術、成果物の抜け漏れ防止策などの対策が実施される。しかし、成果物の品質向上を目指すあまり、要件が膨れ上がって工程が終わらないなど、別の問題に直面することもある。

要件定義工程<sup>\*1</sup>の主な活動は、スコープマネジメント<sup>\*2</sup>である。システム開発の発注者と受注者<sup>\*3</sup>を合意形成に導き、システム化要件のベースラインを確立する役割を担う<sup>\*4</sup>。企画プロセス等で検討された業務要求やビジネス要求、現状システムからの技術要求などを受けて、技術・コスト・期間を踏まえた実現可能性を評価し、システム化要件を決定する<sup>\*5</sup>。

本論では、ベースラインの確立に向けたスコープマネジメントの実施方法を、プロセスと評価の観点から示す。まず、2章では、要件定義工程の位置づけと特性について確認し、どういった問題が発生しやすいかを示す。次に、3章では、要件定義の成果物をどのように定め、評価していくかについて論じる。そして4章では、実際に要件定義をするにあたりどのような計画

や評価を必要とするかについて示す。最後に、その実行においては、プロジェクトマネージャを補佐する企画人材やコンサルタントの活用が今後必要になることを提起する。

## 2. 要件定義工程の特徴と問題点

要件定義工程は、他の開発工程と違った特徴を持つ。工程の位置づけ、関係者、視点の違いなどである。そのためが発生する問題点も多い。

### 2.1 要件定義工程の特徴

要件定義工程は、二つの視点への対応が必要である。一つは、企画者（発注者）の視点であり、もう一つは、エンジニアリング（開発者）の視点である。企画者の視点とは、システム化の目的であるビジネス要求や業務要求に対してシステム化要件が対応できているか、エンジニアリングの視点とは、決定したシステム化要件が次工程の設計への入力情報として十分か、である。そのため、要件定義工程では、違う視点の違った立場の関係者の間で、要件に関する内容やコスト・期間などの合意形成が求められる（図1）。

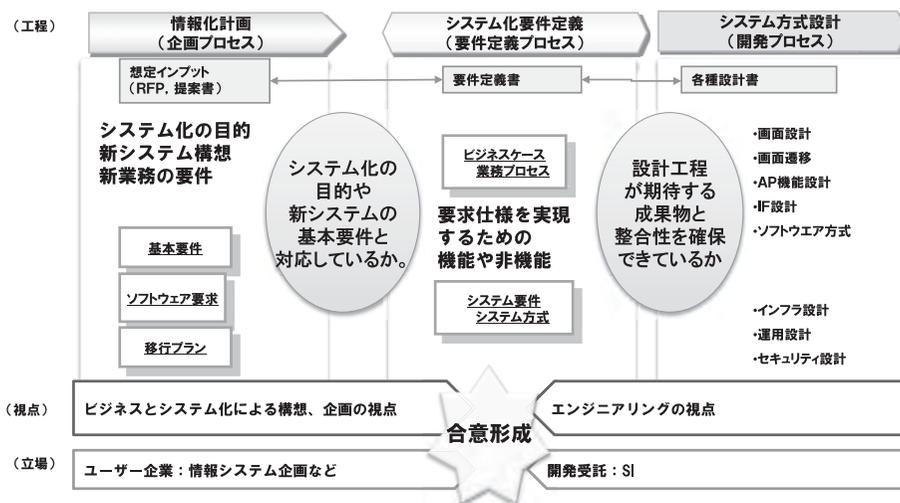


図1 要件定義工程に関する二つの視点

また、二つの視点には、図2に示すような抽象度のレベルや関心の違いがある。こうありたいという要望やニーズへの関心といった抽象度の高いものと、どのように解決するかという具体的な方式への関心である。

こういった視点と抽象度の違いが発生するのが、要件定義工程が持つ特徴である。

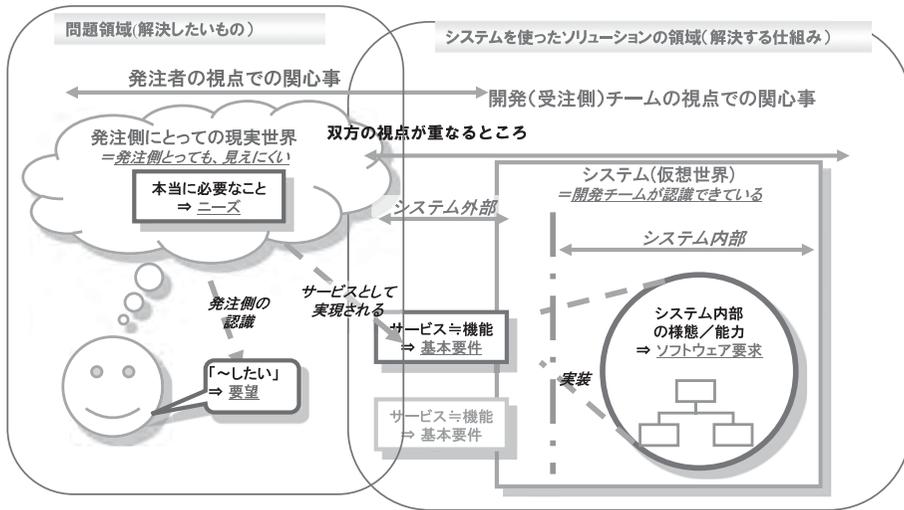


図2 企画者（発注者）とエンジニアリング（開発チーム）の関心の違い

## 2.2 要件定義で起きやすい問題点の追求にあたって

要件定義に起因する問題を追求していくと、成果物の作成を中心とした対策に帰着しがちである。たとえば、業務知識が不足していたため業務を詳しく定義する、要件漏れがないように網羅的に記述する、できるだけ詳細に書く、あいまいな記述がないようにしていくなど、いずれも成果物を作成するという視点が強い（図3）。

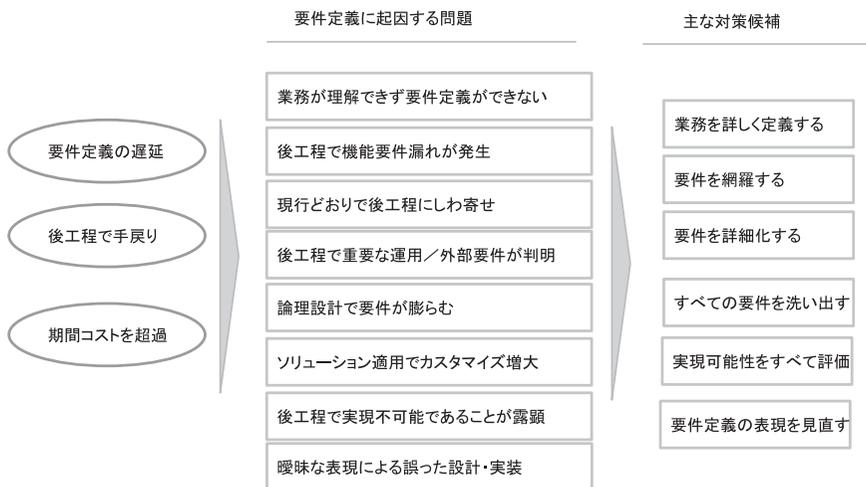


図3 要件定義に起因する問題と主な失敗原因抽出例

こういった成果物を中心とした対策を進めると、そのこと自体が目的化され、新たな問題を誘発することになる。たとえば、以下のような成果物の深さに関する基準や取扱い方に関する問題である。

- 1) 要件漏れをなくそうとして網羅的に詳細化を進めてしまい、論理設計と内容が重複す

るレベルになってしまう。

- 2) 同様に詳細化に時間がかかり、要件定義が終わらない。
- 3) 成果物を完成させたいがあまり、要件の重要性について考慮せず、一定の量が作成できれば、一部を先送りして、終了とする。(結果、後工程で重要な要件が漏れたことが発覚し、大きな手戻りになる)

こういったケースは、成果物はできているが、どのようなレベルで成果物の作成を終了すべきか、という指標がないために発生する。

一方、要件定義の特徴として、要件の妥当性について判断が必要になることが挙げられる。システムの機能や非機能要件を具体化する中では、業務要求との整合性、運用面と非機能要件、システム利用者からみた機能や非機能の充足度などを踏まえて、要件の妥当性を判断する。こういった判断には、システム全体の視点が重要である。また、要件として何が重要で落とせないのか、後工程で定義できるものはあるのかなど、優先度や重要性の判断には、ビジネス要求との整合、実現可能性などの観点が重要である。

以上のことから、要件定義工程では、成果物の終了基準の設定と、システム全体の視点で要件の整合性を確認し、システム化要求や業務要件との合目的性を確保することが必要である。

### 3. 要件定義の特性を踏まえた進め方

前章では、要件定義を進める上では、企画視点とエンジニアリング視点が重要であり、システム全体の観点での評価が必要であることを示した。本章では、要件定義の特性を踏まえた上で、成果物の関係をどう捉えて評価していくべきか、そのためには、どのような進め方が必要になるかについて示す。

#### 3.1 成果物の関係と評価について

要件定義工程で、スコープ調整し、ベースラインを確立するには、利害関係者の視点を意識した成果物を規定することが重要となる。通常、設計工程以降では、システム化の対象が絞られており、成果物も予め規定されたものを基に進めることができる。つまり、成果物の範囲が明確であり成果物の達成度合いが進捗として利用できる。しかし、要件定義工程では、システム化に関わる人や組織、視点の違いがあるため、成果物をどのように決めるかが重要となる。要件が妥当で、かつ、承認を得るには、関係者の関心や重視したいことを成果物として定義する。たとえば、業務の立場では、担当業務への影響が関心事であり、システムを利用した結果どのように業務が実現されるかが重要である。実装方式や機能は、結果的に、業務にどのような影響があるかを示す必要がある。

こういった成果物の規定については、それぞれシステムで取り扱う業務や利用特性によって違うため、定義にあたっては工夫を必要とする。たとえば、以下のような個々の成果物の関係性に着目したものを要件定義の成果目標とすることができる。

- システム化の目的と要件の整合性に関する評価を示したもの
- システムの機能と業務の関係を明示したもの
- 非機能要件とシステム化後の運用の関係

- 各機能間の関係と重要な機能の見極め
- システムで保全するデータとビジネストランザクションの関係
- 既存システムと新システムを比較して運用要件の違いを示したもの

こういった成果物間の関係性に加え、前章でも示したように、企画視点とエンジニアリングの視点に着目し、妥当であることを確認していく。たとえば、開発者の関心事である設計要件との整合性については、モデル契約書<sup>[2]</sup>などでも示されているように、エンジニアリングプロセスとして標準的な成果物が存在する。例として日本ユニシスのエンジニアリングプロセス<sup>\*6</sup>で定義した成果物を基に、要件定義と論理設計間の成果物関連を示す(図4)。

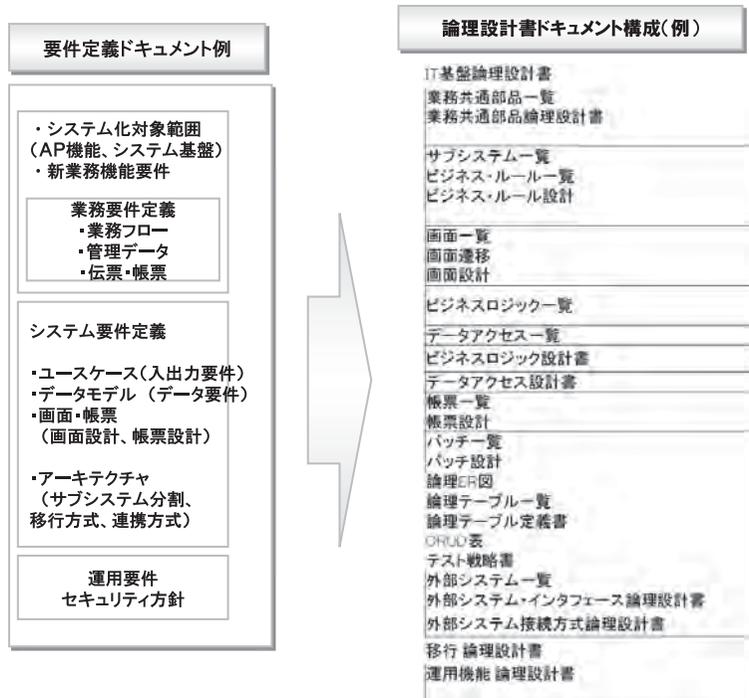


図4 日本ユニシスのエンジニアリングプロセスにおけるドキュメント例

一方、企画視点では、個々のビジネスや業務、前提とするシステムが違うために、対象を絞り込まず、成果物の定義も難しい。まず、企画段階での要求間の整合やシステムへの要求事項などをまとめ、対象範囲を明らかにしてからでないと要件定義が難しくなる。そのため、コンサルタントが入って情報化計画やITプランニングという形で検討され、RFP (Request for Proposal) を整備していくことも多い<sup>[3]</sup>。

本論では、要件定義工程の入力情報の観点から想定して、企画プロセスでこういった成果物があればいいのかをフレームワークとして例示する(図5)。このフレームワークでは、システムで実現したい機能を要求という形で整理することを前提とする。事前にどのようなことを実現したいのか、ビジネス用語や業務、現行システムからの要望として整理し、特にシステム化の背景や要望・ニーズを踏まえたビジネス要求・システム化要求を文書化し、基本要件、ソフトウェア要求仕様として定義する。

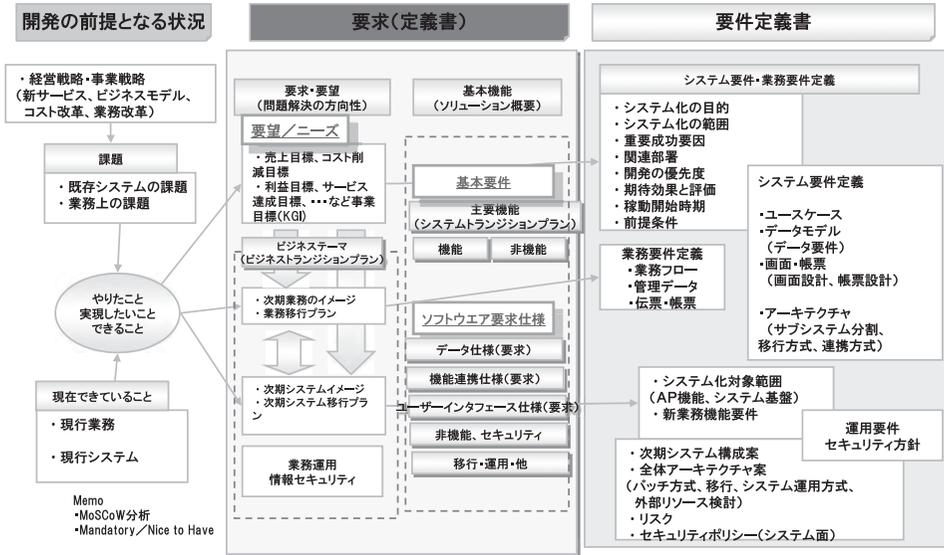


図5 開発の前提から要求・要件までのフレームワーク

### 3.2 評価を繰り返すプロセス

発注者側が情報化計画段階で考えていることと、実現可能性が十分評価されていることの整合を取っていくには、それぞれの妥当性だけではなく、繰り返し要求との関連を評価することが必要である。現実問題として、RFPに書かれている内容のレベルには差があることも多い。本来であれば、企画プロセス等で対象となるシステムの範囲や開発の方針、システム化後のビジネスゴールなどが整理され、要件定義工程でスコープを明確にすればよい。しかし、実現したい要望に対して、開発者視点を加えて評価したとき、期間やコスト・技術的なレベルであいまいさが残っていることも多い。実際、トレードオフが発生し、計画段階からの見直しに及ぶケースもある。そのため、どの程度固まっているかによって進め方を決めるよう、要件を決めながら、あいまいな要求があるのかどうかを確認する。要件を決めようとする要求事項が一見矛盾しているように見えることもある。そこで本当に必要な要件とは何かを検討し、要求の

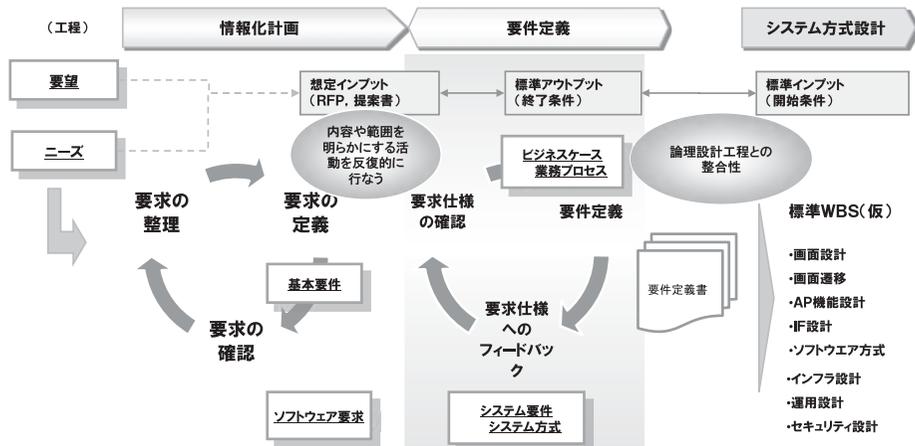


図6 反復的プロセスによる要件定義の進め方イメージ

意図を踏まえて妥当性のある要件に着地させる。つまり、繰り返し要求と要件を確認するような反復型の進め方が求められる。図6に、そのイメージを示す。

#### 4. 実際の適用に向けて

前章で示したように、要求との関連を評価し、要件として落とし込むためには、反復型のプロセスが重要である。しかし、実際に反復型を進めるには、計画をどう立案し、進捗をどう評価していくかが課題となる。本章では、マイルストーン型の計画と、進捗の断面を捉えて段階的に要件を固めていく方法を提唱する。

##### 4.1 計画の立て方と評価ポイント

計画は、WBS型ではなく、マイルストーン型で立案する\*7。WBS型は、ブレークダウンした作業項目をワークパッケージ化し、それぞれのタスクを完成させる形で進めるが、マイルストーン型計画は、期間で区切り、それぞれに目標を設定する。単に、月次や週次の進捗を評価するのではなく、内容がどのレベルに達している必要があるかまで踏み込み評価する。

実際の計画例として、3ヶ月が要件定義期間として与えられた場合を想定し説明する。まず、要件定義工程を小さなプロジェクトという形で捉え、最初に、プロジェクト立ち上げと要求確認ステップの設定を行う。ここで何を成果目標とするか、重要な機能や要件をどのように想定しているかを確認する。その後、それぞれの要件定義作業では段階的に要件を固めていく。常に全体感を意識できるように配慮する。最後に見積りやリスクに関するまとめを実施する。表1に、3ヶ月の実施プランの例を示す。

表1 要件定義のマイルストーン計画例

No	マイルストーン	期間 W (Week)	内容	評価ポイント
1	プロジェクト立ち上げ	1W	要件定義の成果目標の確認 意識あわせの打ち合わせ ビジネスゴールとシステムの持つ特性の確認	企画者、開発者でどのようなことが達成できていればいいか 何が重要かについて共有できているか
2	要求確認ステップ	1W	何が重要な機能なのかについての検討と確認 根幹となる業務やシステムの機能の確認 複雑な業務についての確認	主要な機能の抽出ができていないか 複雑な機能や業務への対応方法を 確認できているか
3	各要件定義計画	1W	期間の設定 セッションの組み立て	関係者とのセッションが現実的な計画となっているか
4	要件定義作業実施(各担当)	8W	ステップで分割 ・おおよそ2W単位 ・各レビュー	各要件の進捗、見直し 業務・システム機能、非機能相互の確認、影響の確認
5	クロージング	2W	定義できている内容に基づく 後工程での見積り 後工程で発生しそうなリスクの抽出	見積り評価 見積り幅 リスクと影響の確認 リスク対策案

## 4.2 進捗評価の方法

進捗は成果物の達成度合いを定量化したパーセンテージなどの評価方法だけではなく、各マイルストーンで、進捗全体を断面として切り取り、各タスクを横断して評価する。前節で例示したマイルストーンを計画図として設定すると図7のようになる。

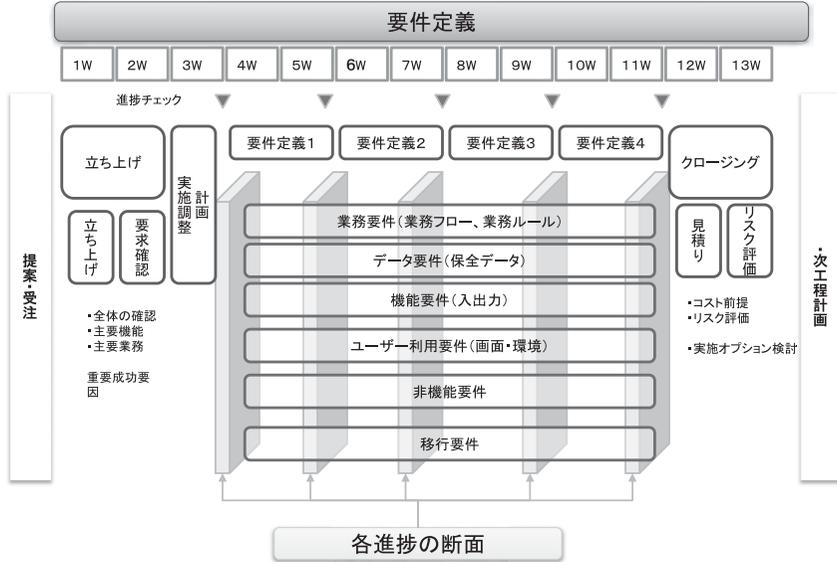


図7 要件定義の計画例

図7の要件定義1～4を反復的に実施する中で、どのように各断面を評価するかが重要である。単に実施進捗率では、反復型を計測できないため、実際の進め方にはすこし工夫が必要となる。

計画例では、六つのタスクが並行して走っている。通常、これらは完全に並列ではなく、ある程度、順序性がある。また、現実問題として各要件の関連性を確認しながら進めるにしても、順序性があつたほうが進めやすいし、進捗も測りやすい。

たとえば、保全データは、入出力機能がある程度出てきた段階のほうが定義しやすいケースもあるし、業務ルールがあつたほうがわかりやすいこともある。また、システムの入出力要件と業務フローは、関係が深いために調整しながら進めたい、といったことが発生する。

本論では進め方の事例として、事前に、ある程度の業務フローやビジネスルールが明らかになっていて、保全したいデータもおおよそ把握できている状態であると仮定する。そして、要件を決めるための順序性や前提資料を踏まえて計画を見直す。ポイントは、成果物をどう仕上げていくかという範囲の絞り込みと、段階的にフィードバックを反映させる繰り返しプロセスである。図8にプロセスの例を示す。

この計画では、前提文書を基にして、最初の段階で主要機能についての分析と検討作業を実施する。要件定義1の段階では、入出力情報やユーザー利用の要件に関して基本的な理解を共有する。最初にシステムの概略が共有できたあとは、調整を繰り返し、段階的に要件を固めながら、気づいていなかった点などを評価してフィードバックを実施する。

重要なのは、ある断面で各成果物を総合的に評価することである。総合的とは、関係者間の視点を加えることと、成果物間の関連性に着目し、システム全体の視点を加えることである。

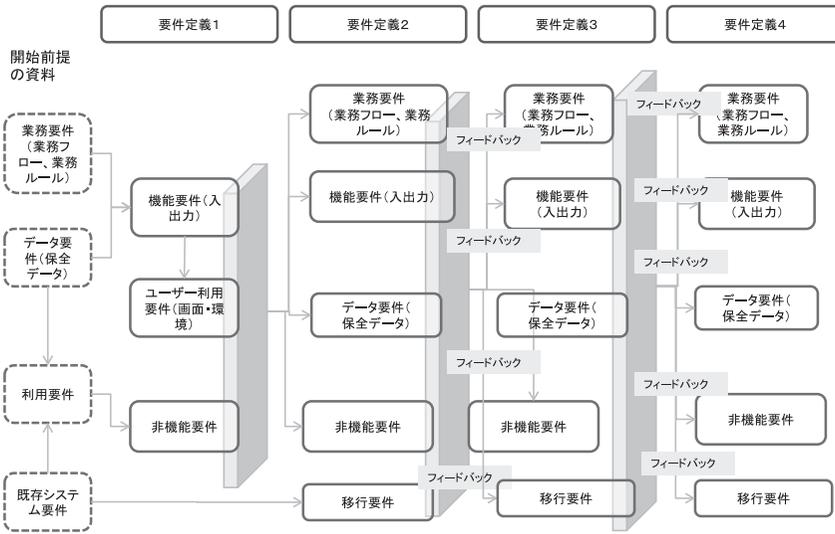


図8 断面ごとに段階的に精度を上げる計画

その上で、それぞれの成果物作成における課題も踏まえた進捗評価を実施する。その後、次の断面では課題への対応を考慮しながら、要件検討の範囲を拡張して定義していく。こういった段階的な進め方により、まず主要な要件を押さえることができる。仮に、特別な事情で時間切れになっても、枝葉部分が先送りされることになり、リスク対応がしやすくなるメリットがある。各断面における評価例は表2のとおりである。

表2 各断面での主要評価ポイント

段階	主な成果物の状況	評価ポイント
要件定義1終了	(前提資料) 業務フロー、ルール ・システム機能要件 (主要機能) ・ユーザー利用要件 (主要画面のみ) ・非機能要件 (初期構成案)	・前提資料を基に、システム化の概要を記述できているか ・システム要求やビジネス要求に対応づいた機能や画面イメージの確認と評価
要件定義2終了	・業務フローなどへのフィードバック ・システム機能要件 (主要機能の拡張) ・ユーザー利用要件 (画面の種類、操作性などに着目したサンプルの抽出) ・非機能要件 (機能要件を受けて、アクセス量やデータ量などの想定検討) ・データ要件 (機能要件を受けて想定するデータ項目の抽出と精査)	・業務、システム機能、システムの使用性、データ保全量、非機能要件などが連携してシステム構築のイメージができていないかを評価 ・設計工程に入る上で、主要な項目が落ちていないかを評価 ・サイジング要件などへの反映 移行がある場合は移行要件の評価
要件定義3終了	前段階での評価をフィードバックし、各要件の拡張の実施 ・システム機能、データ要件、非機能要件、移行要件、ユーザー利用要件などのファイナル版に向けた資料の作成 ・詳細にできていない部分の可視化	各要件に対する、内部レビュー、顧客レビューの実施 主要機能だけではなく、システム化実現後の運用や稼働イメージを共有した評価 詳細なルール、機能で記述できていないものの見通し
要件定義4終了	各要件の成果物のほぼファイナル版	決まっていないものの評価 スケモレの可能性評価 設計工程以降の実装上の課題など

### 4.3 ベースラインの確立に向けたポイント

要件定義工程では、スコープマネジメントを主たる活動として、成果物の範囲や深さに関する調整を行い、ベースラインを確立する役割を担っている。ベースラインの確立に向けては、合意形成と妥当性の確保が重要である。特に、業務と機能、運用と非機能、コストと開発期間などトレードオフの対象となりやすい要件は、早い段階からスコープ調整を必要とする。本論で示したマイルストーン計画と断面に着目した評価は、要件を段階的に定義していくことで、妥当性評価ができてきているか、合意形成に向けて利害関係者からの理解が得られそうか、などの定性的な状況を把握できる。工程の最後になって一気にトレードオフ調整や合意形成をするにはリスクが大きい。そのため、断面を評価する際に、合意形成できそうなポイントや、トレードオフとなる主要な要件についての検討を進め、リスクとして可視化しながらベースラインの確立を実現する<sup>\*8</sup>。

こういった進め方は、プロジェクト管理上は負荷がかかり、かつ、利害関係者の要求を聞きすぎてしまうなどの理由から適用を難しくしている。また、定性面に着目するため扱いにくいといったことも適用の障害となっている。実際、限られた期間で要件定義を終了したいというプレッシャーを感じているプロジェクトメンバにとっては、まず成果物の達成量という定量的な指標を重視したい。また、現実問題としてこういった進め方をできる人はいるのか、といった点が気になる。しかし、要件定義はシステム開発全体に影響を及ぼす重要な工程であり、開発の前提を決める工程でもある。管理のしやすさだけで量的な問題を重視するのではなく、合意形成や妥当性評価といった要件のベースライン確立に必要な課題を積極的に捉え、リスク定量化によるアプローチを試みるなど適用に向けた工夫をすることが重要である。

## 5. おわりに

本論では、要件定義工程の持つ特徴と発生しやすい問題点について確認し、ベースラインの確立に向けたスコープマネジメントの実施方法を、プロセスと評価の観点から示した。要件定義工程は、企画プロセスに続いて“何をシステム化するか”という点について主にスコープを決定する役割を持つ。そのためには、反復的で、かつ、段階的な進め方と、断面評価という仕組みが有効であることを提唱した。特に、利害関係者間の視点を踏まえた要件調整、トレードオフの選択、合意形成に向けたリスク可視化には、高度な進め方が必要になることも述べた。

実際、プロジェクトの運営については、新たなスキルセットが要求される。要件定義は、設計工程との繋がりから開発側のエンジニア<sup>\*9</sup>が実施することが多いが、企画プロセスとの繋がりも重要であり、企画視点をもって要件定義に臨まないと、プロジェクトのゴール設定からスコープ決定までの流れを実行することが厳しい。

本来、エンジニアは、どのように実現するかに対して責務を持ち、そのために技術力や創造性を発揮する。つまり、要件定義工程では、要件全体を捉えて、システムの実現性を評価し、設計工程との整合性に主たる責務を持つ。そのため、設計工程との関係には強い。

一方で、企画視点を実行するには、別のスキルセットを持った人材の登用が考えられる。たとえば、エンジニア以外の人材に目を向けると、コンサルタントや企画系人材であれば、段階的な進め方や反復、リスクの評価など、企画視点での進め方に対応できる。つまり、プロジェクトマネージャを支える形で新たな人材をプロジェクトに参画させ、企画者視点の活動を進めることで、要件定義の抱える問題点を解消する支援ができる<sup>\*10</sup>。

今後は、要件定義はエンジニアが担うという固定した考え方ではなく、エンジニアと協働して要件定義の抱える課題に対応できるスキルセットを持った人材の登用や育成、プロジェクト運営方法の検討が重要と考える。そのためには、組織としてのプロジェクト評価方法の見直しと、新しい要員体制の考え方に対応していかなければならない。

- 
- \* 1 ウォーターフォール開発の工程の一つ。SLCP2007では、企画プロセス、要件定義プロセスとして企画フェーズに含まれ、開発フェーズとは独立する。同様に、経済産業省が策定したモデル契約書(参考文献[2])でも、ユーザー側(企画側)が責任を持って実施するプロセスとして定義された。
  - \* 2 PMI (Project Management Institute) が発行するプロジェクトマネジメント知識体系ガイド (A Guide to the Project Management Body of Knowledge) で、スコープマネジメントは、PMBOKの知識エリアとして定義されている。スコープには、Project ScopeとProduct Scopeがあるが、本論では、Project Scopeに重点を置き、Project ScopeがProduct Scopeに影響を与えているという観点から論じている。
  - \* 3 受託開発におけるシステムの発注者は主に企画を担い、受注者が開発を担うものとする。
  - \* 4 確立されたベースラインは、スコープマネジメントとして継続して管理される。参考文献[8]では、新たな要求や変更についての管理方法について紹介している。
  - \* 5 本論では、要求と要件を区別する。要求とは、実現したいことであり、要件は、実現可能性などの評価を踏まえた備えるべき機能や仕組みを指す。システム開発においては、要求定義や要件定義といった用語の違いが見受けられるが、本論では、企画プロセスを要求定義に対応づける。
  - \* 6 日本ユニシスのエンジニアリングプロセスでは、ウォーターフォール型開発の工程とその成果物について規定している。工程は要件定義・論理設計・物理設計・統合テスト・導入である。
  - \* 7 要件定義でスコープマネジメントを実施した結果、WBSを作成する。設計工程以降はWBS主体で実行する。
  - \* 8 現実には、システム運用や業務運用、システムの最終利用者の視点など多様な視点に対応しなければいけないのでさらに複雑である。
  - \* 9 ここではSE(システムエンジニア)全般を指す。ITスキル標準の人材モデルでは、ITアーキテクトから各スペシャリストが含まれる。
  - \* 10 本来、企画・発注側に企画系人材やコンサルタントを含めた体制をつくり、協働して要件定義を進めるということも必要とするが本論では述べない。

- 参考文献**
- [1] 日本情報システム・ユーザー協会, “平成22年度産業技術研究開発委託費(中小企業システム基盤開発環境整備事業(企業のIT投資動向に関する調査研究及びソフトウェア開発管理基準に関する調査研究))”, 一般社団法人日本情報システム・ユーザー協会, 2011
  - [2] 経済産業省 商務情報政策局 情報処理振興課, “「情報システムの信頼性向上のための取引慣行・契約に関する研究会」～情報システム・モデル取引・契約書～(受託開発(一部企画を含む), 保守運用) 第一版”, 2007/4
  - [3] 濱本佳史, 大角祐司, 新谷将司, 鈴木努, “変革を成功させるITプランニング——プロジェクト・ファシリテーションを活用した計画づくり”, 別冊技報, 日本ユニシス, Vol.33 No.1 通巻116号, 2013年6月
  - [4] 神崎善司, “顧客の要求を確実に仕様にできる要件定義マニュアル”, 秀和システム, 2008/10/23
  - [5] 独立行政法人情報処理推進機構 ソフトウェア・エンジニアリング・センター, “ITプロジェクトの「見える化」上流工程編”, 日経BP社, 2007/5/1
  - [6] 独立行政法人情報処理推進機構 ソフトウェア・エンジニアリング・センター, “共通フレーム2007 第2版”, オーム社, 2009/10/1
  - [7] 吉田伸一, 山崎晋, “契約管理と契約書”, ユニシス技報, 日本ユニシス, Vol.20 No.3 通巻67号, 2000年11月
  - [8] 三宅晴夫, “要求管理のプロセス紹介”, ユニシス技報, 日本ユニシス, Vol.20 No.3 通巻67号, 2000年11月
  - [9] Adam Ding, “PMBOK 5th Edition Study Guide”, the 5th one in the new PMP exam cram series, 05: Scope (New PMP Exam Cram), 2013/8/1

[10] 岡村正司, “徹底解説！プロジェクトマネジメント”, 日経 BP 社, P44-45, 2003/6/28

**執筆者紹介** 宮 脇 亨 (Tohru Miyawaki)

日本ユニシス(株)入社後, 地域金融機関担当を経て, Java, Object 指向系の技術主管を担当. 開発手法の策定, 標準化コンサルティング, 情報化計画やシステム企画など顧客視点と開発者視点の両面から顧客サービスに従事. 現在, アドバンスド技術部に所属.

