

# メソドロジー整理によるシステム開発計画立案の精度向上

## Improving the Accuracy of System Development Planning by Organizing Methodologies

中 村 誠

**要 約** システム開発プロジェクトの計画立案段階において、先達の経験知や技術の継承による開発計画の精度向上と次世代の人材育成を目的に、メソドロジー（方法論）を活用した「全体マップ」と「プロジェクト実施計画書、プロジェクト管理計画書」のガイドライン及び「開発実行計画」を作成した。「全体マップ」では、開発計画の本質的な意味の正しい理解と開発全体の網羅的かつ俯瞰的な把握が容易となり、「全領域」「全工程」について、目的・構成要素・主要タスクを明確化することにより、抜け漏れのない開発計画の立案に寄与する。「プロジェクト実施計画書」及び「プロジェクト管理計画書」のガイドラインでは、顧客要求をシステム要件として、見積可能な機能や作業レベルに分解する手順を説明し、「開発実行計画」では、ゴールに至るプロセスを整理し、遂行フィジビリティを担保した。

**Abstract** At the planning stage of a system development project, we created an “Overall Map” using methodology, guidelines for “Project Implementation Plan and Project Management Plan” and “Development Execution Plan”, with the aim of improving the accuracy of development plans and fostering the next generation of human resources by inheriting the experience and technology of our predecessors. The “Overall Map” makes it easy to correctly understand the essential meaning of the development plan and to get a comprehensive and bird’s-eye view of the entire development, and clearly identifies the objectives, components, and main tasks for “all areas” and “all processes”. This contributes to the formulation of development plans without omissions. The guidelines for “Project Implementation Plan and Project Management Plan” explain the steps to break down customer requests into system requirements and into estimable functions and work levels, and the “Development Execution Plan” describes the process to reach the goal, and ensured the feasibility of execution.

### 1. はじめに

社会全体のデジタル化が進み、大量データを分析して傾向を把握するビッグデータ解析や、AI（人工知能）の画像認識やコミュニケーション支援への利用などの技術革新が続いている。

ユーザ企業や組織においては、老朽化や複雑化、ブラックボックス化したレガシーシステムの存在が経営・事業戦略上の足かせ、高コスト構造の原因となる、いわゆる「2025年の崖」が示す現状の課題対応が急務である<sup>[1]</sup>。あらゆる産業において、先端技術を駆使したビジネスモデルが台頭し、事業環境や社会要請が変化する中、各企業では持続的成長に向けた、情報システム刷新や新規構築が不可欠となっている。他方、少子高齢化により我が国の労働人口は減少が見込まれ、特に若年人口の減少が顕著である。IT人材については、2030年に最大で約78.7万人が不足する可能性があるとの試算もある<sup>[2]</sup>。

BIPROGY 株式会社（以降 BIPROGY）においても、開発スキルとリーダーシップを兼ね備

えた人材の不足が顕在化している。大規模/スクラッチ開発の経験者が、すでに現場から離れつつあることから、先達の経験知や技術の継承と次世代の人材育成が急がれる。

こうした課題の解決に向けた施策として、既存の制度・規約・先人の経験知や技術などを実践的に適用可能なナレッジ（明文化した業務知識）として整備する取り組みに着手した。この取り組みでは、ウォーターフォールモデルによる1,000人月規模のスクラッチ開発を前提とした。様々な開発形態<sup>\*1</sup>がある中で、スクラッチ開発を採用した理由は、1) データベース構造、アーキテクチャ、基盤などを一から全て設計・実装する形態であり、開発難易度が非常に高いこと、2) スクラッチ開発のベースモデルを作成することで、他の開発形態への展開も比較的容易になること、3) 基幹システムなどの大規模開発における適用頻度が依然として高い傾向にあること、などである。

本稿では、本取り組みで作成したナレッジについて説明する。まず2章でシステム開発の問題点と解決の基本方針を述べた後、3章でナレッジとして作成した「全体マップ」を、4章で「プロジェクト実施計画書」と「プロジェクト管理計画書」を紹介する。なおこれらのナレッジは社内利用を前提としており、社外への提供や関与は予定していない。本稿の目的は、BIPROGYの取り組みや考え方を共有し、お客様が安心してBIPROGYをパートナーにして頂けることと、同業者やユーザ企業に、計画立案のヒントにして問題を解決してもらうことである。

## 2. システム開発プロジェクトの問題点と解決に向けた基本方針

本章では、一般的なシステム開発プロジェクトの現状を概観したうえで、BIPROGYが受託した個々の開発プロジェクトの計画立案段階における問題点とその原因、問題解決に向けた基本方針と施策を述べる。

### 2.1 システム開発プロジェクトの現状と問題点

システムの複雑化・大規模化や、品質・コスト・納期に関する要求の高度化を背景に、今日におけるシステム開発プロジェクトの遂行難度は極めて高くなっている。

日経BP総研の調査<sup>[3]</sup>では、プロジェクトの成功率は、2008年の31.1%に対して2018年は52.8%と向上傾向であり、その要因は、プロジェクトにおける定量管理の普及としている。しかし裏を返せば、2018年では47.2%が失敗しており、失敗の具体的な内容と原因は表1のとおりである。

表1 失敗の具体的な内容と主要原因

失敗の具体的な内容	主要原因
コストを順守できなかった	<ul style="list-style-type: none"> <li>・追加の開発作業が発生した</li> <li>・追加の設計作業が発生した</li> <li>・見積りが甘かった</li> </ul>
スケジュールを順守できなかった	<ul style="list-style-type: none"> <li>・システムの仕様変更が相次いだ</li> <li>・各工程の見積りが甘かった</li> <li>・当初のスケジュールにそもそも無理があった</li> </ul>
品質に満足が得られなかった	<ul style="list-style-type: none"> <li>・要件定義が不十分であった</li> <li>・テストが不十分であった</li> <li>・移行作業に問題があった</li> </ul>

また、苦勞した工程は「要件定義」が筆頭であり、これは、2003年以降の15年間ほぼ変わらない。

日本情報システム・ユーザー協会の調査<sup>[4]</sup>では、当初予定に対して、予算超過が20.7%、工期遅延が27.4%、品質満足度（不満）が15.7%であり、500人月以上の場合は、その割合は2倍程度とさらに悪化する。当初予定通りにならなかった主な要因は、工期・予算・計画時の考慮不足、現行業務・システムの複雑さ、仕様変更多発であり、ここでもプロジェクト準備と上流工程が問題となっている。

## 2.2 BIPROGYでのプロジェクトの問題点と発生原因

BIPROGYでは、プロジェクト全体の開発工数規模が一定以上の全プロジェクトで工程完了時に品質保証レビューを実施している。このレビューで問題（次工程への進行または現工程の継続に支障をきたしていること）があると判定された割合は、2018年から2022年の期間で、1.7～3.6%の範囲で推移しており、低水準ではあるものの、撲滅には至っていない。主な原因はプロジェクトの準備段階や上流工程にあり、日本情報システム・ユーザー協会の統計とも一致する。

BIPROGYにおける受託開発プロジェクトの事故を社内で分析した結果、主な問題は次の3点に集約されることがわかった。

- 1) 過去の事例を流用してプロジェクト計画を立案したが、前提条件の差異の見逃し等により、エンジニアリングプロセスに抜け・漏れが発生した
- 2) ビジネススコープに対するタスクの洗い出しが不十分で、当初見積りに対し、実績コストが大幅に増加し、本稼働時期の延伸を余儀なくされた
- 3) 顧客要求の把握が不十分で、ゴール像が定まらない状態で開発に着手した結果、要件定義が当初予定の工期で終了しない、次工程以降での要件変更が多発する、という状況に陥った

これらの問題はすべて、計画立案段階で予見し改善されていれば回避できたはずであった。これらの問題事象が発生した原因としては、次の3点が挙げられる。

### 【原因1】計画立案作業の形骸化

計画立案に関する既存のナレッジは、計画書の建付けや構成として、計画書をどのように作ればよいかを中心に説明したガイドになっており、利用者が計画の本質的な意味を理解しないまま活用しているケースがある。

### 【原因2】タスク抽出・検証のスキル不足

開発計画に盛り込むべき主要なタスクを整理したナレッジが散在しているため、開発全体を網羅したスキルの習得が難しい。適切な開発計画を立案するための教育も不足しており、タスクの抽出時に発生した抜け漏れについて、開発計画の検証工程で気づくことができない。

### 【原因3】計画段階における要求（要件）分析スキルの不足

計画段階における要求分析にフォーカスしたナレッジが不足している。また、要求分析を経験する機会も少ないため、スキルの習得が難しく、結果として要求分析が不十分となる。

### 2.3 問題解決の基本方針

2.2節で述べた、【原因1】開発計画の本質の理解不足の対策は、メソドロジー（方法論）を網羅的に整理したナレッジを整備し、開発の各領域（AP、インフラなど）・工程の持つ目的・主要構成要素・主要タスクに関する本質的な意味を理解させる方針とした。

メソドロジーとは、開発の各領域・工程の総称であり、次の四つの要素で構成される。

- 1) コンセプト（目的）
- 2) メソッド（主要構成要素）
- 3) プロセス（主要タスク）
- 4) ツール

これらの要素を、縦軸を領域・横軸を開発工程としてマトリクス状に整理したものを、「全体マップ」と呼ぶことにした。この「全体マップ」を理解することで、【原因2】タスクの抽出・検証不足も回避できるものと考えた。

【原因3】計画段階における要求分析スキル不足については、計画段階における要求の分析・整理方法や分析の深さ、開発実施計画を作成するためのガイドラインを整備し、その活用を推進する方針とした。

ここで掲げた具体的施策については、3章以降でより詳しく説明する。

## 3. メソドロジーを活用した全体マップの整理

本章では、2.2節で挙げた【原因1】開発計画の本質の理解不足の対策、【原因2】タスクの抽出・検証不足の対策として整備することにした「全体マップ」を紹介する。

### 3.1 全体マップの位置づけと概要

システム開発では、「全領域」「全工程」について、その目的・構成要素・主要タスクを把握・理解していなければ、精度の高い開発計画を立案することはできない。「全体マップ」は、システム開発全体を網羅的かつ俯瞰的に把握できるよう、各領域/工程ごとにメソドロジーとして整理したシステム開発の鳥瞰図である。

システム開発プロジェクトでは、「全体マップ」に定義されているレベルの開発プロセスを各領域の担当者が把握し、また、プロジェクトマネージャーは開発時の抜け・漏れを見抜くために全体を把握しておくことが求められる。「全体マップ」は、そのために利用できる資料として位置づけている。

「全体マップ」を整理するうえで重視したのは、次の2点である。

- 1) 読み手が開発計画の本質的な意味を正しく理解できること

「全体マップ」は、各領域や工程ごとに目的（コンセプト）、主要な構成要素（メソッド）、主要なタスク（プロセス）をExcelシートに整理したものである。それぞれの工程について、その本質的な目的と管理視点が、また、各領域についてもその本質的な目的が簡潔明瞭に示されている。これにより、読み手が、各工程や領域の目的を理解した上で、その構成要素や主要タスクを理解できるようにした。

- 2) 開発全体を網羅的かつ俯瞰的に把握できること

各シートは、メソドロジーにより開発の全構成要素を抜け・漏れなく、明快に分かりやす

く表現することを追求し、縦軸に領域を二階層で並べ、横軸に工程を配置した二次元マトリクスの一覧表とした。

コンセプト一覧を例に、「全体マップ」の構成イメージを図1に示す。

工程名		立ち上げ準備	要件定義	論理設計	物理設計	プログラム開発	結合テスト	システムテスト	受入検査	移行・導入	保守
マネジメント	統合マネジメント	■全体マップ(コンセプト一覧)									
	品質管理	【開発】 開発スコア: 2.5									
	財務管理	【開発】 開発スコア: 2.5									
	進捗管理	【開発】 開発スコア: 2.5									
	リスク管理	【開発】 開発スコア: 2.5									
	要員管理	【開発】 開発スコア: 2.5									
AP	情報セキュリティ管理	【開発】 開発スコア: 2.5									
	機能(スクラッチ)	【開発】 開発スコア: 2.5									
	機能(リライト)	【開発】 開発スコア: 2.5									
アーキテクチャ	共通処理	【開発】 開発スコア: 2.5									
	処理方式	【開発】 開発スコア: 2.5									
テスト	標準化	【開発】 開発スコア: 2.5									
	テスト	【開発】 開発スコア: 2.5									
インフラ	環境	【開発】 開発スコア: 2.5									
	運用	【開発】 開発スコア: 2.5									
移行	移行共通	【開発】 開発スコア: 2.5									
	業務移行	【開発】 開発スコア: 2.5									
保守	システム移行	【開発】 開発スコア: 2.5									
	データ移行	【開発】 開発スコア: 2.5									
保守	保守	【開発】 開発スコア: 2.5									
	保守	【開発】 開発スコア: 2.5									

図1 全体マップ構成のイメージ

縦列の領域は、メソドロジーを表現する単位・粒度の検討に際しては、社内の開発標準や業界の標準的な手法と、ガイドラインであるPMBOK<sup>\*2</sup>、ITIL<sup>\*3</sup>を参考にした。第一階層は、マネジメント、AP、インフラ、移行をベースに、APやインフラに含まれる構成要素の内、共通と考えられるアーキテクチャとテストを括り出し、保守体制や保守方針などを事前に描き、プロセス面や要員リソース面などを明確化するために、保守を追加した七つを領域として定義した。

マネジメントの第二階層は、社内標準の「プロジェクト管理計画書」とPMBOKを目次レベルで比較し、補完して定義した。AP領域は、当初、第二階層を機能のみとしていたが、スクラッチ、リライト、ストコン(ストレートコンバージョン)の定義内容に相違点が多いことから別階層に分けた。アーキテクチャ、インフラは、コンセプトの異なる領域を第二階層とした。移行は、開発者がデータ移行、システム移行、業務移行の区別がついていないことが多いため、この切り口を新たに設け、第二階層として表現している。第二階層には、利用者が領域の範囲と目的を把握し易くするため、「構成要素」と「実施目的」を定義している。

横列の開発工程は、プロジェクトの立ち上げ準備に加えて、要件定義、論理設計、物理設計、プログラム開発、結合テスト、システムテスト、受入検査、移行・導入、保守とした。また、当該工程の「実施目的」、「ポリシー」、「管理視点」の3点を定義した。

### 3.2 メソドロジーを活用した全体マップの定義

ここでは、AP領域の機能（スクラッチ）を例に定義内容を述べる。当領域の実施目的は、「業務要求からシステム化すべき機能を実現する」であり、構成要素は、「業務、機能、インタフェース、データモデル」としている。

AP領域では、コンセプト一覧を、工程ごとに次のとおり定義している。

- ・要件定義：業務とシステムの流れの整理と、機能・インタフェース要件の明確化
- ・論理設計：機能とデータの流れの整理と、機能・インタフェースの具体的な処理の明確化
- ・物理設計：システムの具体的な処理を物理的に配置変換
- ・プログラム開発：物理的に配置変換された処理をコードに変換

要件定義、論理設計のメソッド一覧、プロセス一覧は、図2に示すとおり。定義内容は、工程間のつながりや同一工程のメソッド・プロセスの関連性を視覚的に確認できる粒度とした。メソッドは、各工程の主要構成要素を一覧化しており、プロセスは、主要タスクを業務分析などの分析単位、業務システム分析などの設計単位で階層化して定義している。

<メソッド一覧の定義内容>		<プロセス一覧の定義内容>	
要件定義	論理設計	要件定義	論理設計
<ul style="list-style-type: none"> <li>・業務/システム関連</li> <li>・アクター,ビジネスルール</li> <li>・用語,業務フロー</li> <li>・システム化業務</li> <li>・システム化業務説明</li> <li>・機能要件,画面要件</li> <li>・帳票要件,他シスIF要件</li> <li>・概念データモデル</li> </ul>	<ul style="list-style-type: none"> <li>・機能フロー,論理機能</li> <li>・論理バッチ,業務共通処理</li> <li>・論理画面,論理帳票</li> <li>・論理他シスIF,データフロー</li> <li>・共通業務仕様</li> <li>・論理データモデル</li> <li>・論理テーブル定義</li> <li>・CRUD</li> </ul>	<ul style="list-style-type: none"> <li>・業務分析</li> <li>・業務システム分析</li> <li>・アクター分析</li> <li>・業務フロー分析</li> <li>・業務ルール分析</li> <li>・用語整理</li> <li>・機能分析</li> <li>・システム化業務分析</li> <li>・トランザクション分析</li> <li>・インタフェース分析</li> <li>・機能要件分析</li> <li>・機能関連分析</li> <li>・概念データモデル分析</li> </ul>	<ul style="list-style-type: none"> <li>・全体設計</li> <li>・機能フロー分析</li> <li>・データフロー分析</li> <li>・共通業仕様分析</li> <li>・機能設計</li> <li>・論理機能設計</li> <li>・画面設計</li> <li>・帳票設計</li> <li>・バッチ設計</li> <li>・共通機能設計</li> <li>・システム共通処理設計</li> <li>・業務共通処理設計</li> <li>・一覧設計</li> <li>・論理データモデル分析</li> </ul>

図2 機能（スクラッチ）のメソッド一覧、プロセス一覧定義

AP領域以外についても、社内外のナレッジに基づき定義しているが、インフラについては、全てのプロセスを網羅したナレッジが存在しておらず、非機能要求に関する項目とその項目ごとに求められる要求レベルを網羅的にリストアップしたIPA（情報処理推進機構）の「非機能要求グレード活用シート」<sup>[5]</sup>やITIL、および優良プロジェクトの情報を組み合わせて定義した。

### 4. 開発計画の策定プロセスと作成すべきドキュメント

本章では、2.2節の【原因3】計画段階における要求（要件）分析スキルの不足の対策として、計画段階で明確化すべき事項と開発計画の策定プロセスを整理した上で、開発計画として整備することにしたドキュメントに関するガイドラインやテンプレートを紹介する。

#### 4.1 開発計画策定プロセスの整理

開発計画策定段階で明確化し、取り纏めるべき項目を洗い出し、各項目の記載内容やその粒度・注意事項条件を整理することで、開発計画ドキュメントとして盛り込むべき事項を漏れなく抽出した(表2)。

表2 計画段階で明確化すべきこと

明確化の対象	作業タスクや条件	作成対象文書
Why (何のために作るか)	・ 構想書として整理	・ 構想書
What (何を作るか)	・ 要求に対するシステム要件が明確なこと ・ 見積可能なレベルに詳細化されていること	・ プロジェクト実施計画書
How (どう作るか)	・ エンジニアリングプロセスを工程定義で整理 ・ 設計書等の作成ドキュメント定義	・ 工程定義
How (どのように動作保証するか)	・ テスト戦略の整理	・ プロジェクト実施計画書
How (どのように旧システムから新システムへもってくるか)	・ 移行方針の整理	・ プロジェクト実施計画書
How (開発環境として適しているものは何か)	・ 開発, テスト等の実施環境の利用数を整理	・ プロジェクト実施計画書
Who (誰が作るか)	・ 体制図/役割定義	・ 体制図
When (スケジュールは)	・ マスタスケジュール	・ マスタスケジュール
		・ 中日程
Where (どこで作るか)	・ 開発場所, 機器等の設置場所	・ プロジェクト実施計画書

計画の精度や遂行フィジビリティの担保には、計画策定に関するガイドラインを整備することが先決である。本取り組みでは、開発計画ドキュメント類の目次構成や記載粒度(=何を、どこまで分析しておくべきか)の理解を促すべく、サンプルやガイドを整備し、分析スキル不足への対策としている。

計画立案プロセスと作成すべきドキュメントとの関係を含む、計画策定の全体像を図3に示す。計画策定段階では、明確化すべき事項を整理した後、開発計画として取り纏める。計画策定プロセスは、次の二つの段階で構成される。

- ・ 事実認識と仮説を置いて「ビジョンや開発戦略を可視化」する構想段階
- ・ 前提事項を置いて「スコープとゴール」「開発実行プロセス」を整理する計画段階

構想段階では、顧客と共同でシステム化の目的と開発戦略を検討・整理し、「構想書」として取り纏める。

続くプロジェクト計画段階では、「構想書」をもとに、機能面・非機能面におけるスコープとゴールを「プロジェクト実施計画書」として、それを実現するためのマネジメント計画を「プロジェクト管理計画書」として、それぞれ整理する。その上で、顧客特性やプロジェクト特性

等を考慮し、ゴールに至るプロセスを「開発実行計画」として整理する。

「構想書テンプレート」「プロジェクト実施計画書作成ガイド」「プロジェクト管理計画書作成ガイド」は、メソッドロジーとして整理した「全体マップ」をもとに作成することで、開発タスクの網羅性を担保した。

この章の後続する各節では、開発計画として策定すべきドキュメントと、その標準化を図るべく整備したガイド・テンプレート類を紹介する。

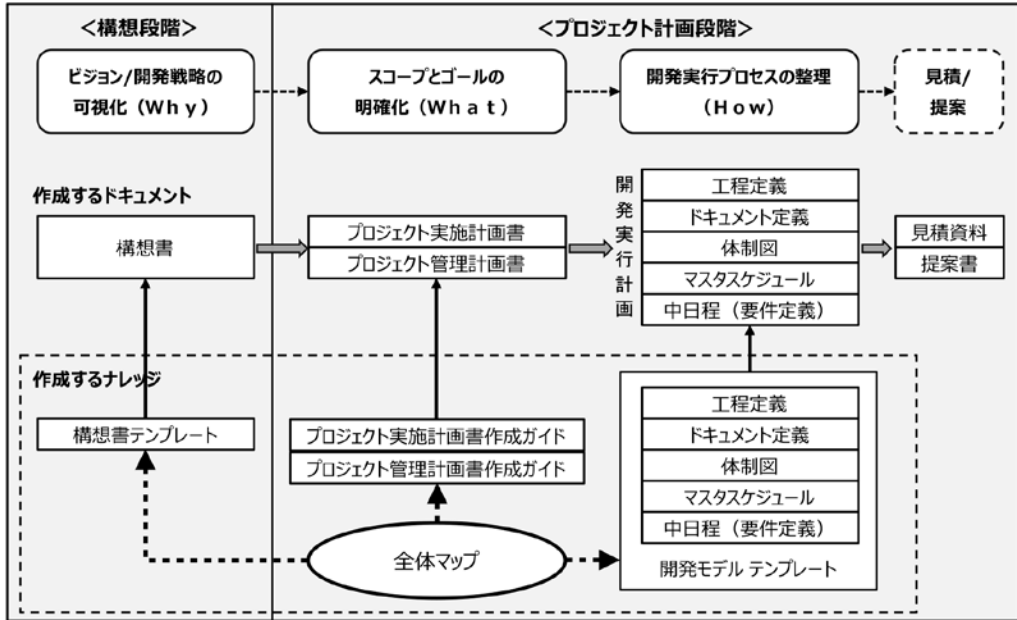


図3 開発計画策定のプロセス

#### 4.2 構想書の位置づけと概要

構想段階の目的は、社会課題や顧客要求などに対し、これを解決するためのシステム像を描き、実現に向けた開発戦略を検討整理することにある。「構想書」とは、課題を認識する前の社会課題の段階、IT 構想を持つ前の段階や実現方法検討前の段階において、いち早く課題を認識してシステム化の目的や開発戦略を取り纏め、顧客を含めた関係者間で Why (ビジョン)・What (スコープとゴール)・How (戦略) を確実に共有するためのものである。

「構想書」では、Why として課題認識、実現したいこと、および顧客要求を、What として

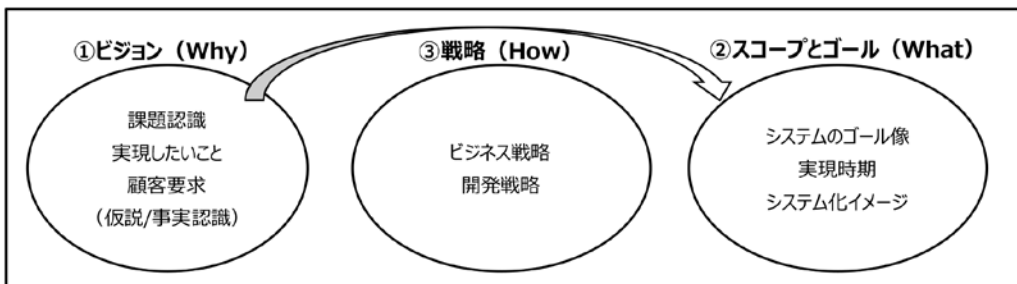


図4 構想書の基本構造



実現すべきシステム像，対象領域，実現時期，システム化イメージ等を，それぞれ整理する。その上で，Whyで整理した顧客課題や要求に基づき，Whatで整理したシステム像を実現するためのビジネス戦略，開発戦略などをHowとして整理し，「構想書」として取り纏める。

「構想書」の標準化を図るべく整備した「構想書テンプレート」では，基本的な検討要素を網羅しており，「構想書」に記載すべき一般的な項目とその記載粒度・記載レベル感に関する指針を取り纏めた（図4）。

### 4.3 プロジェクト実施計画書の位置づけと概要

「プロジェクト実施計画書」は，顧客要求をシステム要件として具体化し，見積可能な機能や作業のレベルに分解することを目的としている。システム開発プロジェクトでは，期限・コスト・開発スコープを顧客と合意し，開発を進めることになり，受注者であるSierには，その合意内容に対する「完成責任」や「説明責任」を果たすことが求められる。また，合意した品質，コスト，納期を達成するための計画の精度を高め，その遂行フィジビリティを担保することや，見積内容に関する説明責任を果たすことも求められる。

計画段階において「プロジェクト実施計画書」を作成する上で必須となる，要求の分析・整理方法や見積り方法に関するガイドラインとして，AP・インフラ・アーキテクチャ・テスト・移行・保守の各領域（計6分冊）からなる「プロジェクト実施計画書作成ガイド」を整備した。各分冊の章立てを図5に示し，詳細を1)～6)で説明する。

ガイド名	章立て	ガイド名	章立て	ガイド名	章立て	
AP	AP設計要素定義	アーキ テクチャ	共通処理	移行	移行共通	
	業務分析		処理方式		移行方式・スケジュール	
	機能分析		オンライン処理方式		移行体制・役割・運営	
	データモデル分析		バッチ処理方式		コンテンツエンシープラン、特別体制	
	見積り		帳票出力方式		業務移行	
インフラ	非機能		外部接続処理方式		標準化	対象業務
	システム構成の概略		標準化		BIPROGYにおける標準化	業務移行方式・移行構築・検証
	非機能要件の定義		全体設計		全体設計	業務研修
	環境		設計基準		システム移行	移行対象システム
	環境空間		開発基準		移行対象システム	システム移行方式・移行構築・検証
	各環境定義	見積り	システム本番移行	システム本番移行		
	環境構築	テスト戦略定義	データ移行	移行対象データ		
	運用	テストアプローチ	データ移行方式・移行構築・検証	データ本番移行		
	システム運用の定義とは	テストの自動化	見積り	見積り		
	システム運用の範囲	見積り	保守	保守準備作業の定義プロセス		
	システム運用項目の定義			保守方針		
	プロセスの概要			保守範囲・体制		
	運用項目の整理手順			見積り		
見積り						

図5 プロジェクト実施計画書ガイドの章立て

#### 1) AP

プロジェクト計画の策定段階では，顧客要求をもとにシステムが具備すべき機能を整理した上で，システム規模を見積る。その際，単なる機能名の羅列だけでは，各機能名称の裏にある「深さ」までは判断できない。要件定義に近いレベルで業務・機能・データの各モデルを分析することで，機能名の裏にある「深さ」を評価し，これに基づいて見積りを実施すれば，精度のある見積りになる。見積りの精度が低いと，作業スケジュールや要員投入計画の

誤差も大きなものとなり、プロジェクトの遂行フィジビリティが担保できない。

AP編では、次の各項につき、定義すべき内容とその目的を整理した。

- ・ 業務分析：現行業務や課題の調査・分析，システム化する新業務の範囲と流れ
- ・ 機能分析：システム化の範囲とする業務と機能，およびインタフェースの種類の明確化を目的とした，システムの観点での業務分析と機能抽出
- ・ データモデル分析：業務で扱うデータを顧客が理解できるレベルでモデリングした，概念ER図とエンティティ定義

## 2) インフラ

可用性，性能・拡張性，運用・保守性などの非機能要件を明確にすれば，採用すべきアーキテクチャの妥当性が判断でき，工数を見積ることができる。更に，システム規模や全体工期・スケジュール等を加味して開発・単体テスト・結合テスト・ステージング等に用いる環境を多重度も考慮して定義し，環境の構築作業を計画や見積りに折り込む。

インフラ編では，次の各項につき，定義すべき内容とその目的を整理した。

- ・ 非機能：システム構成の概略
- ・ 環境：プロジェクトの遂行過程で用いる各種環境の定義と，環境ごとのハードウェア，ソフトウェア，ネットワークやサイジング情報
- ・ 運用：システム運用の範囲や運用項目，ならびに，安定稼働を実現させるために定める，監視対象項目・障害対策やリリース運用，定期点検等

## 3) アーキテクチャ

開発対象システムに求められる非機能面の品質要求（性能，保守性，セキュリティ）を，一定の開発生産性を担保しつつ設計・実装するために，当該システムのアーキテクチャを事前に定義し，関係者間の認識を合わせておく。

アーキテクチャ編では，次の各項につき，定義すべき内容とその目的を整理した。

- ・ 共通処理：システム特性に応じたフレームワーク，システム共通処理
- ・ 処理方式：オンライン・帳票・バッチ処理・外部接続の各処理方式に関する，処理パターン，処理イメージ
- ・ 標準化：開発品質と生産性向上を目的とした，全体設計，設計基準，開発基準

## 4) テスト

プロジェクト計画の策定段階では，プロジェクト立ち上げ直後の工程である要件定義や論理・物理設計の進め方に主眼を置きがちで，テスト計画に関する検討がおざなりになる傾向がある。しかし，システムの最終的な本番稼働品質を担保するためにテストが極めて重要なのは言うまでもない。そこで，計画立案段階にて一定レベルの方針・戦略や定量的情報を整理し，見積りや工期・体制等にこれを適切に折り込んでおく。

テスト編では，次の各項につき，定義すべき内容とその目的を整理した。

- ・ テスト戦略定義：テストの全体方針
- ・ テストアプローチ：テストレベル，テストタイプ，テストレベルとテストタイプの関係

- ・ テスト設計アプローチ：テストレベルとテスト設計対象のドキュメントの関係
- ・ テストの自動化：テスト自動化の検討ポイント

#### 5) 移行

プロジェクト計画の策定段階では、移行に関する顧客要求が整理されていない、データ移行対象となる既存システムとその対象データも洗い出されていない、等の理由により、移行についての検討が不足しがちである。しかし、計画立案段階で、移行方式や移行対象等について一定レベルで整理し、見積りや工期・体制等にこれを適切に折り込んでおけば、本稼働までを見据えたプロジェクトとして全体感ある計画を立案し、全工程を通した遂行フィジビリティを担保することができる。

移行編では、次の各項につき、定義すべき内容とその目的を整理した。

- ・ 移行共通：移行要件の整理、全体方針、移行方式、移行スケジュール、体制・役割・運営、コンティンジェンシープラン、業務上の重要イベント
- ・ 業務移行：新システムを利用した業務運営の習熟を目的とした、研修教材や研修カリキュラムの作成支援、研修環境の準備等に関わるタスク
- ・ システム移行：移行方式、システム切替/切り戻し用スクリプト群の構築・検証、移行手順
- ・ データ移行：移行方式、データ移行ツール群の構築・検証、移行手順

#### 6) 保守

システム開発プロジェクトでは、開発フェーズだけではなく、保守フェーズや稼働後のインフラ更改など、システムライフサイクル全体の費用感や保守工程における関係ステークホルダ間の役割分担・前提条件等を予め整理・合意しておくことが重要である。適切な投資判断を行うためには、システムライフサイクル全体にわたる費用対効果を考慮することが求められる。

保守編では、次の各項につき、定義すべき内容とその目的を整理した。

- ・ 保守方針：定期保守、緊急保守、予防保守の分類ごとの保守の方針
- ・ 保守範囲：アプリケーション、ハードウェア・ソフトウェアおよび環境に関する保守範囲
- ・ 保守体制：保守の体制・役割

### 4.4 プロジェクト管理計画書の概要

「プロジェクト管理計画書」では、管理計画書で定義すべき内容に加えて、プロジェクト特性や開発対象システムの特性、開発戦略等を反映することが重要である。「プロジェクト管理計画書ガイド」は、「記載すべき内容と記述方法」と併せて、「各管理領域が必要な理由」「管理上、着目すべきポイント」などに関する説明を重視した。

### 4.5 標準テンプレート類の提供

「開発実行計画」では、マスタスケジュール粒度のレベルで主要タスクとドキュメントを関係者間で明確化し、体制・役割分担を整理して同じ認識に立つことが重要である。「開発実行

計画」の標準化と検討すべき事項の抜け漏れを防止し、実践的な開発モデルを定義するため、4種類の標準テンプレートを次のとおり整備した。詳細を1)～4)にて説明する。

- ・ 主要タスクを整理する「工程定義」
- ・ ドキュメントを整理する「ドキュメント定義」
- ・ 体制・役割分担を整理する「体制図」
- ・ スケジュールを整理する「マスタスケジュール」

#### 1) 工程定義

工程定義は、「大日程ベースの工程定義」と「役割別の工程定義」より構成される。「大日程ベースの工程定義」では、マスタスケジュールに記載する粒度の各工程の主要タスクを「全体マップ」第二階層の領域別に整理する。また、「役割別の工程定義」では、「大日程ベースの工程定義」の定義内容を、担当部門や協力企業などの役割別に分類して、それぞれの役割を明確にする。

「大日程ベースの工程定義」でプロジェクト特性、ドキュメント体系、生産性などを考慮して主要タスクを整理し、「役割別の工程定義」にこれらを反映することで、実行計画を完成させ、計画の妥当性を俯瞰的に確認することができる。

#### 2) ドキュメント定義

ドキュメント定義は、開発戦略をもとに、「全体マップ」の領域ごと、および工程ごとに主要なドキュメントを整理し、それらの関連をフローで定義する。

#### 3) 体制図

体制図は、担当部門や協力企業の要員を組み込んだ体制を定義する。

#### 4) マスタスケジュール

マスタスケジュールは、プロジェクトの開始から終了までの全体を俯瞰し、重要なマイルストーンや他プロジェクトとの関係性などのスケジュール的な要素を整理して、プロジェクト関係者で共有・合意する。

### 5. 残された課題

本取り組みでは、計画段階において、実質上、要件定義プロセスを前倒して実施し、実施計画書を完成することで、抜け漏れのない計画立案と見積り精度の向上を図っている。しかし、次の課題が残されている。

#### 1) ステークホルダの協力取得

計画段階において要件定義プロセスを前倒して行うためには、関係ステークホルダの協力が欠かせない。

#### 2) 高完成度の実施計画書作成の難しさ

さまざまな制約により、実施計画書を高い完成度で作成することが難しい場面が存在する。要件を明確に定義できない場合、多くの前提事項が伴われ、それが実行段階でのリスク増大を招く。

また、システム開発プロジェクトでは、開発標準として定義すべき事項が多数存在するため、そのすべてを短期間で整備することは難しい。新規技術への柔軟な対応も含め、開発標準は継

続的に整備・改訂していくことが求められる。

## 6. おわりに

本稿では、計画立案段階において、開発計画として整備したナレッジを紹介するとともに、計画立案の精度向上には、開発計画の本質的な意味を正しく理解し、開発全体を網羅のかつ俯瞰的に把握することが重要であることを述べた。

IT 業界では、経験豊富な先輩達の知識やスキル、暗黙知を形式知化して、後進に継承するための方法が課題となっている。本取り組みでは、有識者の属人化した経験・ノウハウを集約して形式知化しており、これを普及することで技術継承の一翼を担っている。技術を継承するためには、ナレッジの整備だけでなく、集合教育やメンタリング制度、実案件ベースでの適用支援を並行して推進しなければならない。

ユーザ企業において、DX（デジタルトランスフォーメーション）推進やITを駆使したシステムによる競争力強化などを進めるためには、要件定義/外部仕様レベルだけでなく、プログラム設計/実装に関する計画の精度向上や遂行フィジビリティの担保が欠かせない。本取り組みが、開発計画策定の一助として役立てば幸いである。

- 
- \* 1 開発形態には、スクラッチ開発、ストレートコンバージョン、リライト、パッケージ適用などがある。
  - \* 2 PMBOK は、Project Management Body of Knowledge の略。プロジェクト管理に関する業界標準として世界的に広く使用されているガイドブックであり、Project Management Institute (PMI) が出版する。
  - \* 3 ITIL は、Information Technology Infrastructure Library の略。IT サービスマネジメントにおけるベストプラクティス（成功事例）をまとめた書籍群であり、イギリスの中央政府機関である CCTA (Central Computer and Telecommunications Agency) によって1980年代に開発された。現在は、AXELOS 社がライセンスを所有する。

- 参考文献**
- [1] 「DX レポート～IT システム「2025 年の崖」の克服と DX の本格的な展開～」, 経済産業省, 2018 年 9 月 7 日  
[https://www.meti.go.jp/shingikai/mono\\_info\\_service/digital\\_transformation/20180907\\_report.html](https://www.meti.go.jp/shingikai/mono_info_service/digital_transformation/20180907_report.html)
  - [2] 「IT 人材需給に関する調査 調査報告書」, みずほ情報総研株式会社, 2019 年 3 月  
[https://www.meti.go.jp/policy/it\\_policy/jinzai/houkokusyo.pdf](https://www.meti.go.jp/policy/it_policy/jinzai/houkokusyo.pdf)
  - [3] 「日経コンピュータ 2018 年 3 月 1 日号 [IT プロジェクト実態調査 2018]」, 株式会社日経 BP, 2018 年 3 月 1 日
  - [4] 「企業 IT 動向調査報告書 2023」, 一般社団法人日本情報システム・ユーザー協会 (JUAS) 2023 年 3 月 31 日  
[https://juas.or.jp/cms/media/2023/04/JUAS\\_IT2023.pdf](https://juas.or.jp/cms/media/2023/04/JUAS_IT2023.pdf)
  - [5] システム構築の上流工程強化（非機能要求グレード）紹介ページ, アーカイブ, 独立行政法人情報処理推進機構 (IPA), 2018 年,  
<https://www.ipa.go.jp/archive/digital/iot-en-ci/jyouryuu/hikinou/ent03-b.html>

※ 上記参考文献に含まれる URL のリンク先は、2023 年 10 月 23 日時点での存在を確認。

**執筆者紹介** 中 村 誠 (Akira Nakamura)

1989年日本ユニシス(株)入社。保険会社、証券会社の業務アプリケーション開発に従事。2021年よりSIビジネスの安全性・効率性を目的としたナレッジを作成・展開する組織に所属。

