

社会福祉法人生活クラブ風の村様

# コンサルティング仕様書

見積仕様

2026 年 01 月 30 日

PreSoft

eMu 上田 昌夫

## 改訂履歴

版	改訂日	改訂箇所	改訂概要	作成者
見積	2026/01/30		見積仕様として作成	上田

# 目次：コンサルティング仕様

## 「業務構造改革 要求設計」

<a href="#">1. はじめに</a>	5
<a href="#">1.1.本書の位置付け</a>	5
1. はじめに	5
1.1.本書の位置付け	5
1.2.弊社が理解する貴法人の取り組み	5
1.3.全体像の把握と最初にすべきこと	5
1.3.1. WBS (Work Breakdown Structure : 作業分解構造) と「要求設計」	5
1.3.2. 基本方針：進めるための「2つの軸」	6
2. 支援内容	8
2.1.進め方	8
2.2.WBS 要素と作成物	9
2.2.「都市計画」について	11
2.2.1. 支援内容	11
2.2.2. 作成物	11
2.3.マスターデータ構造について	11
2.4.コンサルティング仕様の説明について	12
3. 支援における基本的な考え方	14
3.1.システム思考	14
3.2.エンタープライズ・アーキテクチャ	14
(1) エンタープライズアーキテクチャとは	14
(2) エンタープライズアーキテクチャのフレームワーク	15
(3) アーキテクチャの必要性	15
3.3.全体構造（静的構造）の把握	16
3.3.1.管理階層	16
3.3.2.業務相	18

3.3.3.一般標準モデル.....	18
3.4.業務プロセスの設計.....	19
(1) 業務プロセスの原理・原則.....	19
(2) Customer-Performer の原則：業務委託関係の対話モデル.....	19
(3) 業務プロセス.....	20
(4) 業務プロセスの標準モデル.....	21
3.5.ユースケース.....	21
3.5.ドメインモデル；データアーキテクチャ.....	22
(1) ドメインモデルと“基幹システム”の定義.....	22
(2) ドメインモデルのデータ構造；勘定パターン.....	22
4. 支援作業について.....	24
4.1.支援作業.....	24
4.2.ヒアリング.....	24
4.3.プロジェクトミーティング.....	24
4.4.期間.....	24
添付資料：都市計画：基本計画と実施計画の目次例 .....	25
添付資料：作成物例.....	27

## 1. はじめに

### 1.1. 本書の位置付け

本書は、社会福祉法人生活クラブ風の村様（以下、「貴法人」と表します）が、株式会社百易ソフトウェア東京とeMu（以下、「弊社」と表します）へ委託するコンサルティング業務「**業務構造改革 要求設計**」に関し、業務の形態、内容、及び、業務遂行の基本的な考え方、方法論、について記載するものです。

本版は「見積仕様書」として作成しており、見積もりの前提条件を表しています。委託決定後は「コンサルティング仕様書」として支援活動の基本方針とします。

尚、実際の活動にあたっては、貴法人ご担当と協議の上、都度確認しながら進めることとし、その結果なども本書に追記していくこととします。

本書は、以下の文書をインプットとしています。

- 弊社「20251223\_風の村報告書\_v201」
- 弊社「20251222\_風の村報告\_別添）ヒアリング意見集」
- 弊社「20260126\_説明会」資料

見積仕様での注）本書中で用いるエンタープライズアーキテクチャなどの言葉についての詳細は、本件をスムーズに進めるため、また今後将来にわたる有益な知識としてご理解いただくため、本件の支援業務の中で「座学」を想定しています。

### 1.2. 弊社が理解する貴法人の取り組み

本プロジェクトは「AI 導入のための現状の把握」から始まりましたが、現在表面化している個々の課題解決を AI による単なる機能導入にとどまらず、ヒアリングでも明らかになった貴法人のみなさんが認識している本質的な構造的課題を解決することと理解しています。

即ち、目指すのは、次のように業務として成立する状態です。

- 源流入力による「一気通貫（リアルタイム・一個流し）」
- ナレッジ・セントラル ～「知識」と「作業」の分離～
- 「計画と実績」による業務の可視化 ～マネジメント不全の解消と按分の自動化～

また、AI などによる“コンピュータ・システム”はこの状態を実現するための手段であり、本件では「ソフトウェアで“機能”を作る」より先に「“業務”として成立させる」ことが要点です。

### 1.3. 全体像の把握と最初にすべきこと

#### 1.3.1. WBS（Work Breakdown Structure：作業分解構造）と「要求設計」

最初に WBS で「やるべきことの 100%」を押さえます。これにより、体制・調達・移行・運用などの抜け漏れを防ぎ、全体的な見通しを立てます。

##### （1）WBS：Level1

図 1-1 は、本件の Level1 の WBS です。

本件の最終目標を「業務構造改革」と設定し、最終的には、設計した業務プロセスと、それを実現する仕組みである“コンピュータ・システム”と両立して構築するとしています。従って、開発／展開／定常運用は、システム開発だけでなく、業務の設計と開発・実装も含んでいます。

WBS 要素として、Level2 には、他に「要求設計」「プログラムマネジメント」「エンタープライズアーキテクチャ」が、必要です。

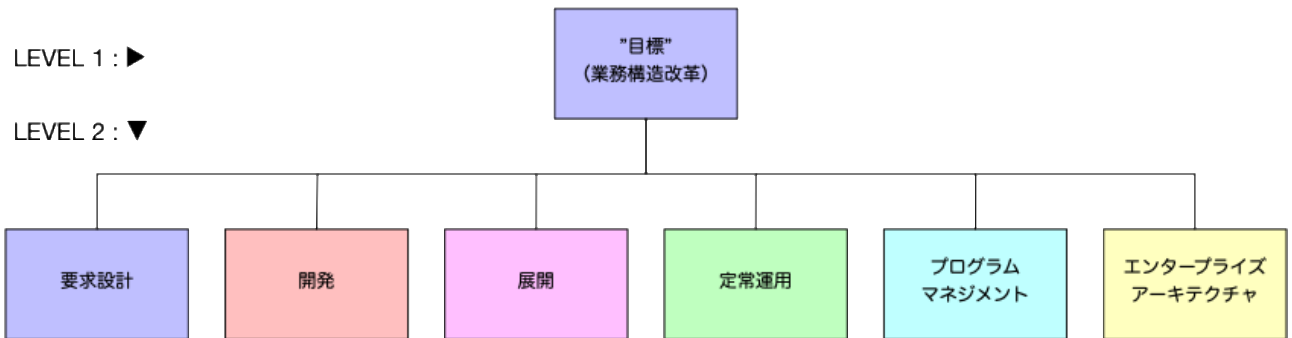


図 1-1. 基本 WBS (Level1,Level2)

## (2) WBS : 要求設計

施主（プロジェクトオーナー、プロジェクトメンバ）が描く新しい業務の構想を「原要求」と呼び、この「原要求」に対して設計者が解決策として提示するものを「要求」として、両者を区別します。

「要求設計」を実施するフェーズでは、設計者が施主とのヒアリングや議論から「原要求」を抽出し解決策となる「要求」を設計する、という意味で「要求設計」と称しています。

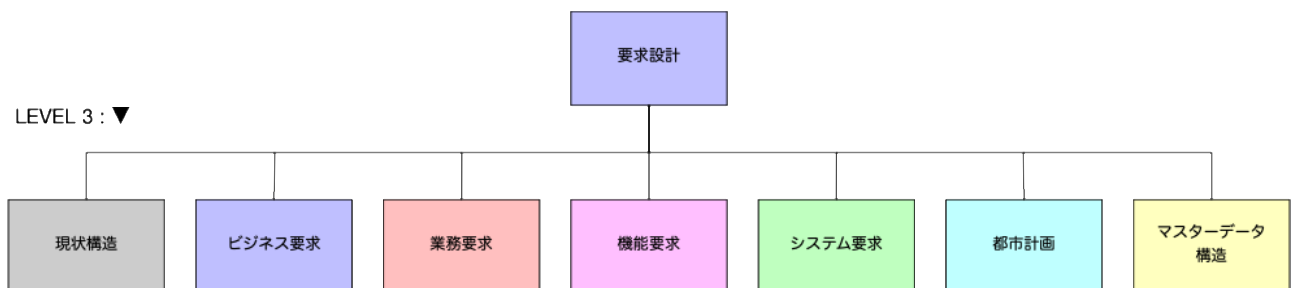


図 1-2. 要求設計 WBS (Level2, Level3)

### 1.3.2. 基本方針：進めるための「2つの軸」

本プロジェクトを成功させるためには、WBS にある以下の「2つの要素」を両輪として回す必要があります。

#### ・ プログラムマネジメント (PM) : 進める力

最終目標の達成に向けて、計画を策定・維持、プログラムを推進する機能です。環境変化に適応しながら、プロジェクトをゴールへと推進します。

注) 複数プロジェクトが立ち上がり、それを全体管理するという意味でプログラムと呼びます  
計画通りにはいかないことを前提としたマネジメントで実績の確認と修正を繰り返します。  
(PMBOK で定義される「適応型プロジェクト」です)

・“計画”の階層化

・基本計画

全体を見通す計画（都市計画）

・実施計画

基本計画で定める“フェーズ”毎の実行の具体化

・作業計画

実施計画で定める“インテレーション”または管理毎の作業内容の具体化

・PMO による日々の管理

毎日、および管理単位、インテレーション、フェーズ毎に計画と実績をチェックし、結果に応じて計画を修正する

・ エンタープライズ・アーキテクチャ（Enterprise Architecture）：整える力

法人全体を、“活動する一つのシステム”として考え、以下の4つの分類体系のそれぞれの階層での“アーキテクチャ（設計方針の原理・原則）”を定めます。

「要求設計」は、図 1-2 の WBS 要素を定義すると同時に、それぞれのアーキテクチャ（設計方針の原理・原則）を定めます。

BA: Business Architecture

DA: DATA Architecture

AA: Application Architecture

TA: Technology Architecture

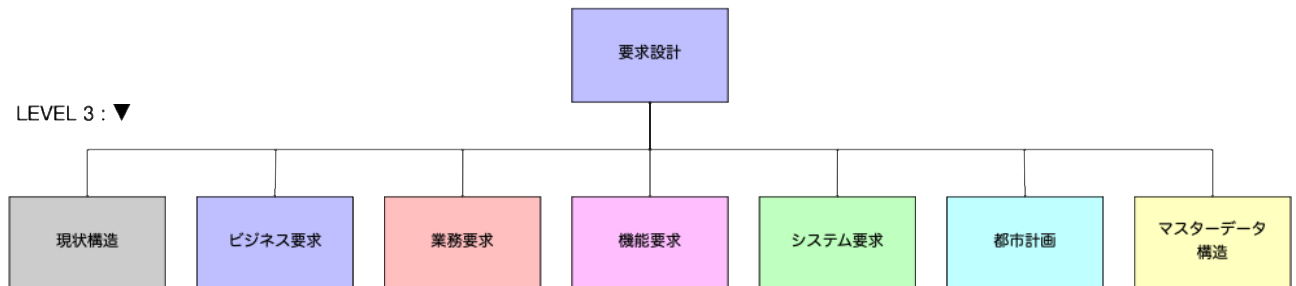
昨今、重要な“セキュリティ”については、上記4階層に加え下記の階層を検討することも有効です。

SA: Security Architecture

## 2. 支援内容

本件は、前項で述べた「**要求設計**」フェーズの実行支援です。

このことにより、全体像（WBS Level1）の中での位置付けが明確になり、現実的なフェーズ単位でのアプローチを環境変化に対応しながら進め最終目標への到達確度をあげることができます。



再掲した上図の「要求設計」の各要素について内容を提示します。

要求設計の中核となる事柄です。「ビジネス要求」「業務要求」「機能要求」「システム要求」を抽出し、業者への RFP（提案依頼書）の一部となる「要求仕様」を策定します。

### 2.1. 進め方

#### （1）ヒアリング

本件の焦点である現ご担当者を中心にヒアリングを実施します。また、業務構造改革として将来を見据えていらっしゃることから、理事長、幹部、関連する全部署からのヒアリングも実施します（対象者の選定、スケジュール調整をお願いします）。

その際、最終成果物だけでなく、中間生成物、他からの入手物も含め全てのドキュメントをヒアリングの内容と並行して確認します（ご提示お願いします）。また、現在利用されている「業務システムその仕様（特に ER 図、データベース構成、など一式）も確認します（ご提示お願いします）。

尚、ヒアリングの内容の整理は、質的研究である M-GTA、SCQM（構造構成的質的研究法）、KJ 法、生成 AI などを適用し作業しますが、作業途中の中間生成物のご提示できませんのでご了承ください。

#### （2）ヒアリングの実施

ヒアリングは一回 2 時間程度で実施します。

対象者は原則お一人ずつですが、現場にてチームで同じ業務をされている場合にはチーム単位で実施することも可能です。対象者の選定はお任せしますが、一週間に二日、1 日あたり 2 セッションが標準的な頻度です。

「要求設計」フェーズでは、全体 3 ヶ月で想定しますので、ヒアリング期間は 2 ヶ月程度、約 25～30 セッションでスケジュールします。

基本一度は対面で、あとは当方の整理状況に応じて、必要な場合は追加ヒアリング、または、メールなどでの質疑応答を実施します。



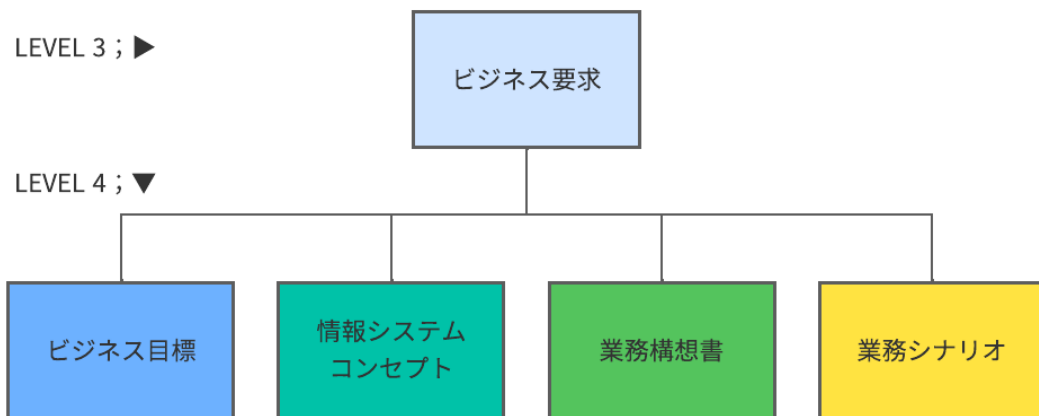
## 2.2. WBS 要素と作成物

### (1) 「ビジネス要求」 Level 3~4

ビジネス要求は、施主及びプロジェクトメンバとの議論、及び、ヒアリングに基づいて策定します。

通常は、“あるべき姿の構想づくり”のセッション（ワークショップに近い）を別に持ち「業務構想書」「業務シナリオ」などを作成していただきますが、本件では省略しヒアリングで「ビジネス目標」「情報システムコンセプト」などを抽出し、都市計画の基本計画書に記述します。

“あるべき姿の構想づくり”は、今を見つめ将来を見据えるには非常に価値のある作業です。もし、ワークショップをご希望であれば、追加で開催も可能ですのでお申し付けください。



### (2) 「業務要求」 Level3~4

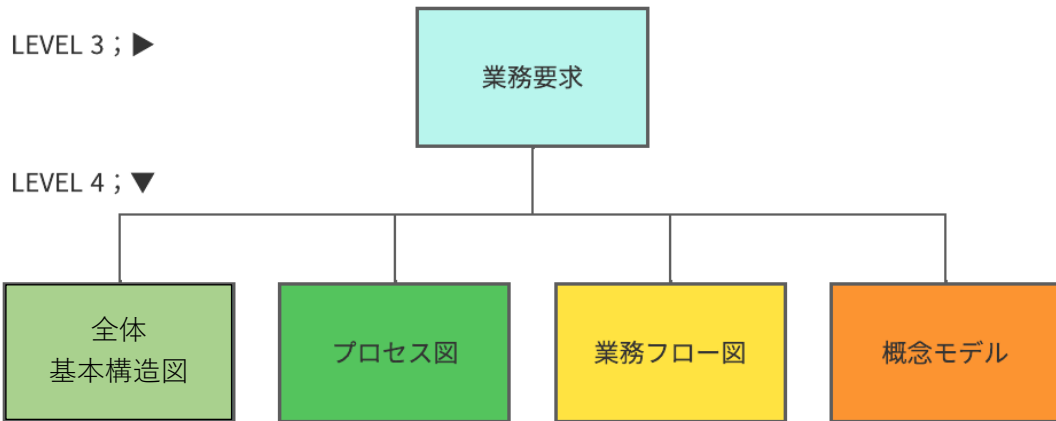
業務要求は、施主（プロジェクトオーナーなど）及びプロジェクトメンバとの議論、及び対象者とのヒアリング（各種資料分析）に基づいて策定します。

日常の業務活動で生成される各種情報を一気通貫した業務プロセスで管理できるように、データ構造の設計方針を定めます。

作成物：

- ・ 全体基本構造図
- ・ 業務プロセス図
- ・ 概念モデル（設計方針）

これらについては、3項でその内容を説明し、添付資料に例を提示します。

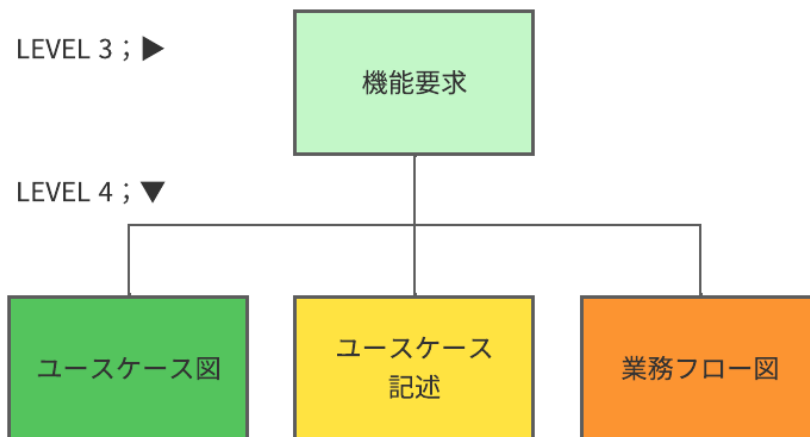


### (3) 「機能要求」 Level3~4

機能要求は、施主（プロジェクトオーナーなど）及びプロジェクトメンバとの議論、及び対象者とのヒアリング（各種資料分析）に基づいて策定します。

本件では、業務フローに現れる各アクティビティ（業務）に対し、その業務での仕事を遂行するために有効、または、あれば良いと考えられる“機能”を発明していただき、“ユースケース”として業務プロセスに追記します。

“ユースケース”については、3項で業務プロセスへの表記を説明します。

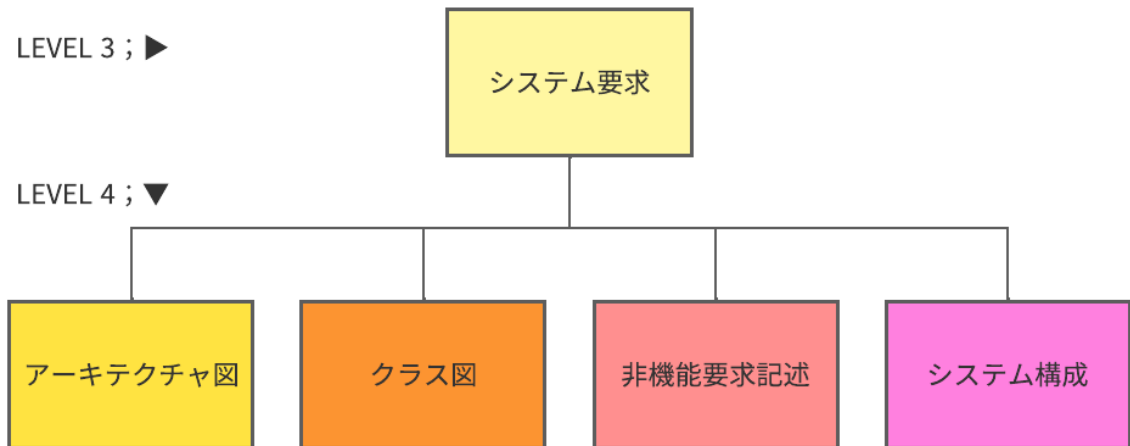


### (4) 「システム要求」 Level3~4

業務要求は、施主（プロジェクトオーナーなど）及びプロジェクトメンバとの議論、及び対象者とのヒアリング（各種資料分析）に基づいて策定します。

本件では、アーキテクチャ図（業務要求での全体基本構造図と相似です）と非機能要求記述をまとめます。セキュリティ要求はここで検討します。

本件では、“アーキテクチャ図”は業務要求の全体基本構造と同等です。他については、ヒアリングや既存システム資料から条件を抽出し、必要に応じ基本計画書に記載します。詳細は、次フェーズ以降で、システム開発方針決定後の「要求仕様書」の策定で検討します。



## 2.2. 「都市計画」について

### 2.2.1. 支援内容

施主及びプロジェクトメンバとの議論、及び対象者とのヒアリング、提供いただく各種資料分析に基づいて、必要な事柄を抽出し、WBS Level3 の要素である「都市計画」を策定します。

「都市計画」は、都市建設で業務構造改革をメタファーする業務構造改革の全体像を包括的に捉えようとする試みです。本件でも採用する根本的アプローチです。

- 既存のインフラストラクチャを活かし、必要なインフラストラクチャを整備し、住民との合意を形成しながら、街区の再開発を漸進的に行う
- 企業情報システム構築も、 既存システム基盤を活かしつつ、基盤・バックボーンを整備し、ステークホルダーとの合意を形成しながら、既存サブシステムとの連携（業務の連携）も確保しつつ、サブシステム単位で再構築を進めていく必要がある
- 時間をかけて少しずつ進めざるを得ない長期的な基本計画であり、常に見直しが必要
- 「経営の仕組み」としての「企業情報システム」全体を、個別のソフトウェアの集合体としてではなく、一つの有機体として捉え、その全体像を包括的に考えます。（システム思考）

「都市計画」は WBS Level2 の要素である“プログラムマネジメント”の作成物でもあります。

### 2.2.2. 作成物

- 基本計画書
- 実施計画書

以上については、添付資料に目次例を提示します。

## 2.3. マスターデータ構造について

一般的にはシステム開発の中で実施されるケースが多いですが、弊社ではエンタープライズ・アーキテクチャでのデータアーキテクチャに相当し。ビジネスとアプリケーション以下の仕組みをつなぐものとして重要視しており、WBS 要素として切り出しています。

本件では、ヒアリングおよびご提示いただいた資料から、貴法人にある「概念」を抽出し「概念モデル」の作成を試みます。

## 2.4. コンサルティング仕様の説明について

弊社の支援では、それぞれの作業で適用する手法を定めています。

世の中にさまざまな手法があることは事実ですが、プロジェクト全体で同じ考え方を使うことで、参加者の理解の深化、作業そのものの迅速化、効率化を図ります。

その内容については、3項に挙げますが、その内容を本フェーズ内でご説明しご理解いただきたいと考えています。

説明については、プロジェクトミーティングで定期的に時間を確保することを想定します。コンテンツやマスターのデータ構造の検討、実装する機能の発明（ユースケース）などは、その中で実践していただきます。

内容は、「ビジネスアーキテクト養成講座」と同等ですので、以下にそのカリキュラムを掲載していますのでご参照ください。

---

### ビジネスアーキテクト養成講座 カリキュラム（全 15 回）

#### 【フェーズ1：基本思想とシステム思考】

従来のシステム開発の限界を知り、企業を「システム」として捉えるための思考基盤を作ります。

- ・ 第1回：次世代システムの基本思想と3つの転換
  - 現状の課題（変化に弱い、経営が見えない）と、構造・アプローチ・設計思想の転換（業務システムから経営システムへ）。
- ・ 第2回：ビジネスアーキテクトの役割と「業務設計」
  - なぜ業務設計が必要なのか、ITとビジネスのギャップ、施主（要求元）としてのアーキテクトの立ち位置。
- ・ 第3回：システム思考の基礎（全体を見る力）
  - システムの定義、還元主義との違い、ハードシステムとソフトシステム、創発性と階層性。
- ・ 第4回：因果ループ図とシステム原型
  - システムの動的変化、フィードバックループ（自己強化・バランス）、問題の転嫁や成功の限界などのシステム原型。

#### 【フェーズ2：ビジネスアーキテクチャの構造と分析】

企業活動を記述するための静的・動的な標準モデルを学びます。

- ・ 第5回：仕事の定義と「もの・こと分析」
  - 仕事とは何か（始めのもの・終わりのもの・基本変換）、手段と目的の分離、機能の抽出。
- ・ 第6回：企業の基本構造（静的側面：ドメイン分割）
  - 管理階層（商流・資源・実行）、業務相（計画実行・報告・知識）、標準ドメイン分割。
- ・ 第7回：業務プロセスの原理原則（動的側面）

- Customer-Performer の原則（依頼・約束・実行・報告・検収）、PD マネジメント、リアルタイム一個流しー。
- ・ 第 8 回：標準業務プロセスモデルの構築
  - 標準的な仕事の進め方（プロセスフロー）、計画の自動生成、例外処理（拒絶・変更・キャンセル）の組み込みー。

### 【フェーズ 3：ドメインモデリングと TEA】

本講座の核となる、会計の仕組みを応用したデータモデリング手法を習得します。

- ・ 第 9 回：概念モデリングの基礎
  - モデルの定義、型とインスタンス、関連と多重度、ロール（役割）、集約と汎化ー。
- ・ 第 10 回：拡張勘定パターン（TEA モデル）の基礎
  - Transaction（取引）、Entry（仕訳）、Account（勘定）の構造。複式簿記の論理を在庫や活動記録に応用するー。
- ・ 第 11 回：資源管理への拡張と「予定・実績」
  - 「在庫」から「資源保有」への拡張、名目と個品、Dual-Date（取引日と登録日）、未来在庫の導出ー。
- ・ 第 12 回：知識と操作の分離（Knowledge Level）
  - 「知識（ルール）」と「操作（仕組み）」の分離、観測パターン、仕様の柔軟な記述、マスタデータの知識化ー。

### 【フェーズ 4：ケーススタディとシステム実装】

具体的な事例（酒問屋問題）を通じた演習と、最終的なシステム実装像を描きます。

- ・ 第 13 回：ケーススタディ「酒問屋問題」その 1（現状分析と課題）
  - 典型的な在庫管理・受発注業務のモデリング、従来型アプローチ（帳票中心・モノリシック）の限界と課題抽出ー。
- ・ 第 14 回：ケーススタディ「酒問屋問題」その 2（モデルによる解決）
  - TEA モデルの適用、移動（Entry）による在庫管理、分割納入やセット品への対応、トレーサビリティの確保ー。
- ・ 第 15 回：次世代システムの実装と「人づくり」
  - アプリケーションプラットフォーム構想、API 設計、業務設計によるサステナビリティと人材育成への接続

### 3. 支援における基本的な考え方

#### 3.1. システム思考

弊社支援での基本となる考え方です。

業務構造改革は、ビジネスの仕組みを、戦略目標を軸に全体で見直すことですが、そのためには、狭い意味での IT やソフトウェアだけでなく、組織、業務、人、商品、そして顧客まで含めた要素一つ一つを連携させて、「社会福祉法人生活クラブ風の村」という大きな“システム”を構築することだと考えます。

この基本的な考え方は「システム思考」として JIS でも定義されています。

システム思考：

- システムとは
  - 多数の構成要素が有機的な秩序を保ち、同一目的に向かって行動するもの (JIS Z8121)
- システム特性
  - 創発（Σ 要素の振る舞い＜全体の振る舞い）
  - 通信，コミュニケーション
  - 階層化と機能分担（自己組織化）
- 対象を「もの」ではなく、目的を持った全体の過程として解釈する
  - 要素に還元できない全体を確認する
  - 全体の中で要素の位置づけを知る
  - 全体が要素を規定すると同時に、要素たちが全体を規定する
  - 原因と思っているものが結果であることもある
- 分析還元主義に対するアンチテーゼ：全体論的（Holism）

#### 3.2. エンタープライズ・アーキテクチャ

##### （１）エンタープライズアーキテクチャとは

弊社の支援の目標は、言い換えると、「エンタープライズ・アーキテクチャ」を創り、法人や組織が継続的に良くなり続けるための仕組みを構築する、ことです。

エンタープライズアーキテクチャ（Enterprise Architecture、EA）とは、法人や組織の情報システム、ビジネスプロセス、データ、技術などを体系的に整理し、その全体像を明確化するためのフレームワークや手法を指します。EA は、組織の戦略目標に対して IT リソースを整合させ、効率的な運営を実現することを目的としています。

エンタープライズ・アーキテクチャは、組織や事業によらない全体の設計方針です。また、大企業だけのものではありません。業種・業態・規模を問わず、やるべきことは同じ、ビジネスや活動と IT /DX をつなげる考え方です。

## (2) エンタープライズアーキテクチャのフレームワーク

四分分類体系のフレームワーク（図）で、階層毎に独立して計画・実施できます。尚、最近では、セキュリティアーキテクチャ（SA）を明示するケースもあります。

### ・ ビジネスアーキテクチャ（BA）；ビジネス（仕事）のやり方・進め方

- 全体基本構造
- ビジネスモデル；  
意味の定義をちゃんと別に書くことが必要
- プロセス図（ビジネスレベルの大きな業務フロー）
- 組織，人材育成の原則（人に関する体系はないため）

### ・ データアーキテクチャ（DA）；データの構造

- 概念モデル
  - BA-AA のデータ間の関係を述べる
  - データの取り扱いの原則

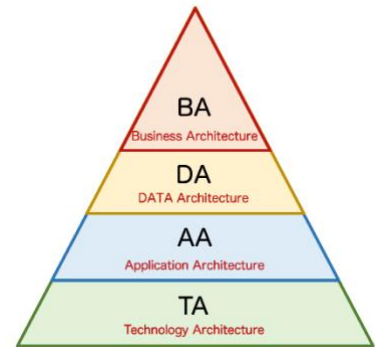
### ・ アプリケーションアーキテクチャ（AA）；ビジネスを支援するアプリケーション

- 全体構造図（論理アーキテクチャ図）
  - 情報システム体系、サブシステムの関係
- 開発プロセスの原則

### ・ テクニカルアーキテクチャ（TA）；技術、インフラ

- 技術体系、IT 基盤
- クラウド、通信、セキュリティ、安全、システムのバックアップセンター、パソコンの仕様・・・原則を決めて具体化する

\*ここから、または、ここだけ書かれる場合がありますが、それは百害あって一利なしです



## (3) アーキテクチャの必要性

アーキテクチャをシステム開発都度ではなく、個別のシステム開発と切り離し、法人全体の設計方針として定めることで、以下のような事柄を目指します。

### ・ ビジネスの質の確保

- ・ 全体と部分の整合・一貫性
- ・ ビジネスシステムと情報システムの整合
- ・ 中長期的展望と更新
- ・ 優先順位

### ・ ソフトウェア作成における生産性の確保

- 独立開発，段階的開発，不整合の回避
- 開発手法のコントロール
- テストの爆発回避
- 運用開始後の変化への追従 → 追従できない分はリスク
  - データの成長と変化
  - 業務の成長と変化
  - 技術の成長と変化
  - ビジネスの成長と変化

### 3.3. 全体構造（静的構造）の把握

“風の村を活動する一つのシステムとして見る” ため全体構造を把握します。

法人や組織は、最上位の経営レベルから末端の現場まで、いくつかの管理階層に自己組織化された構造を持っており、その基本構造とドメイン分割の考え方について述べます。

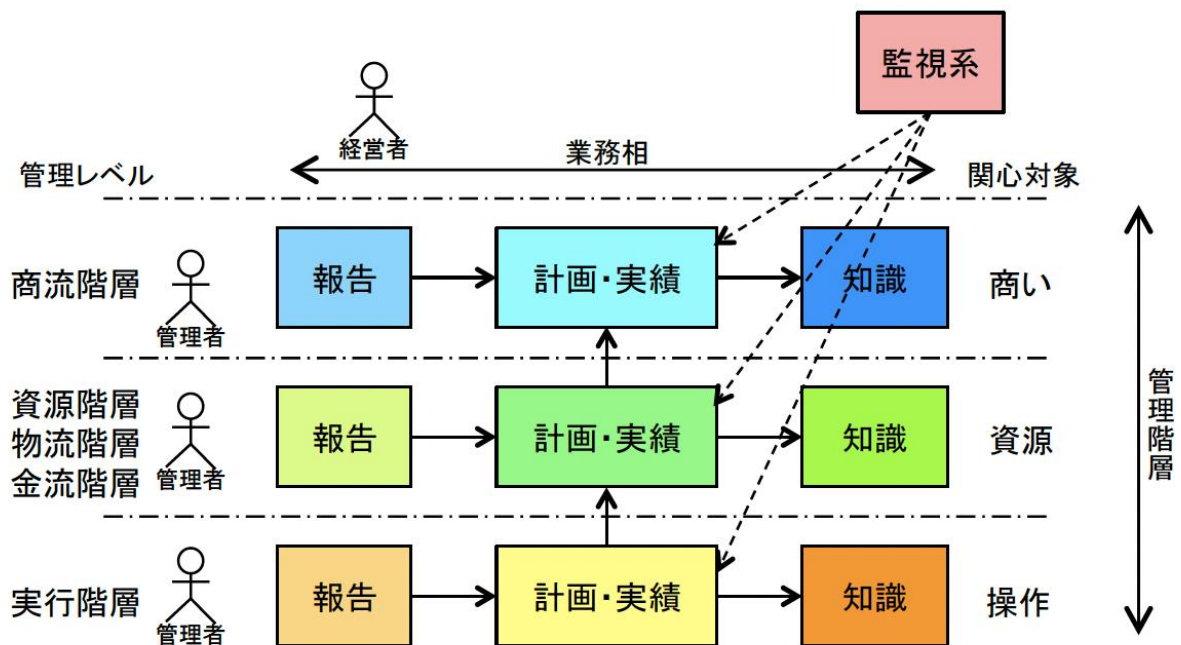


図 3.3-1； 静的基本構造

#### 3.3.1. 管理階層

- 情報システム全体を、その関係対象 (concern) に基づいて階層化します
  - 階層構造は、概括的な関係対象から詳細な関係対象に向かって細分化して配置されます
  - 処理の依頼は作業予定に展開されて実行されます
  - 作業が終了するたびに作業実績が作業予定に対応づけて記録されます
  - 作業を自層で実行せずに下位層に委譲することもあります
  - 下位層で作業が終了する度に作業実績が委譲元に報告され、同様に記録されます
  - 但し、処理の依頼は必ずしも最上位層から入って最上位層から出るとは限りません、処理の依頼が、途中の階層で発生し、その結果も同じ階層で記録されることもあります (Subsumption アーキテクチャ)



- 階層別に、ビジネス上の関心対象のライフサイクルを管理します
  - 階層の上位-下位関係は、後述する「Customer-Performer」の構造に基づく処理の委譲の順序によって決定されます
- 管理階層は、管理の主体と管理対象のライフサイクルに基づきます

### (1) 商流層

- 商流層は、顧客から受け取ったビジネス要求（「依頼」）を満足する事象（「ライフサイクル事象」という）を記録する機能群です
  - ライフサイクル事象が資源の利用を必要とする場合は、資源管理層にその依頼を委譲します
    - 具体的には、受注→出荷→検収→請求→回収、仕入委譲→入荷、生産委譲→入荷、などです
  - 商流層で在庫を持つ（営業在庫）がありますが、これは資源管理層における「ものの在庫」を営業の観点でまとめたものと見ることができます
- 商流層は、施主が得るであろう価値の最大化を図ることを目的とします
  - 何が施主の価値かは施主によって異なります

### (2) 資源管理階層

- 資源管理層は、商流層から委譲された資源に関する依頼（資源管理）を満足するライフサイクル事象を記録する機能群です
- 資源管理層は資源依頼に基づいて、品目、設備、権利（著作権、使用権、など）、お金、時間、空間、情報などの資源を利用（消費または占有）し、あらたな資源を生成するための一連の変換機能を自階層内で行うか、実行依頼を生成して実行層に処理を委譲します
  - 資源の利用にあたっては、必要な資源量を算出（計画）し、利用可能な資源と時刻を特定（引当）し、変換の実施を指示（dispatch）し、その結果を実績として記録します
    - この層の目的は、時々刻々変化する資源の availability を最大限活用すること、すなわち原価の最小化ということができます
- 物流管理ドメイン、生産管理ドメイン、請求回収ドメインなどがこの階層に配置されます
- 商流層から依頼を委譲されることなく、自階層で独自に資源を変換する機能もあります
  - たとえば、部材を仕入れる（工場在庫）、設備を購入・廃棄するなどです

### (3) 実行層

- 実行層は、資源管理層から実行依頼を委譲されて、実行依頼を満足するライフサイクル事象を記録する機能群です
- 実行層は実行依頼に基づいて、一連の手順に展開し、資源の現物を操作することを作業員や制御系（制御装置、ファームバンキングなど）と対話し、実行結果を得ます
  - この層は、実世界と接続しているため、さまざまな現実に対処しなければなりません

- ライフサイクル事象は、資源管理層の実行依頼の忠実な履行であるが、そこには常に操作ミスや中断、失敗、不測の事態が付きまといます
  - そのため、ミスや中断が発生したときは、実行層であらかじめ決められた例外対応を行う（Control）か、資源管理層に報告して新たな実行依頼をもらい（Supervising）、それを履行するという動的な管理機能を持ちます
- 物流実行ドメイン、製造実行ドメイン、請求実行ドメインなどがこの階層に配置されます
  - とくに、製造実行ドメインは、MES（Manufacturing Execution System）あるいは MO&S（Manufacturing Operations and Control）と呼ばれます

### 3.3.2. 業務相

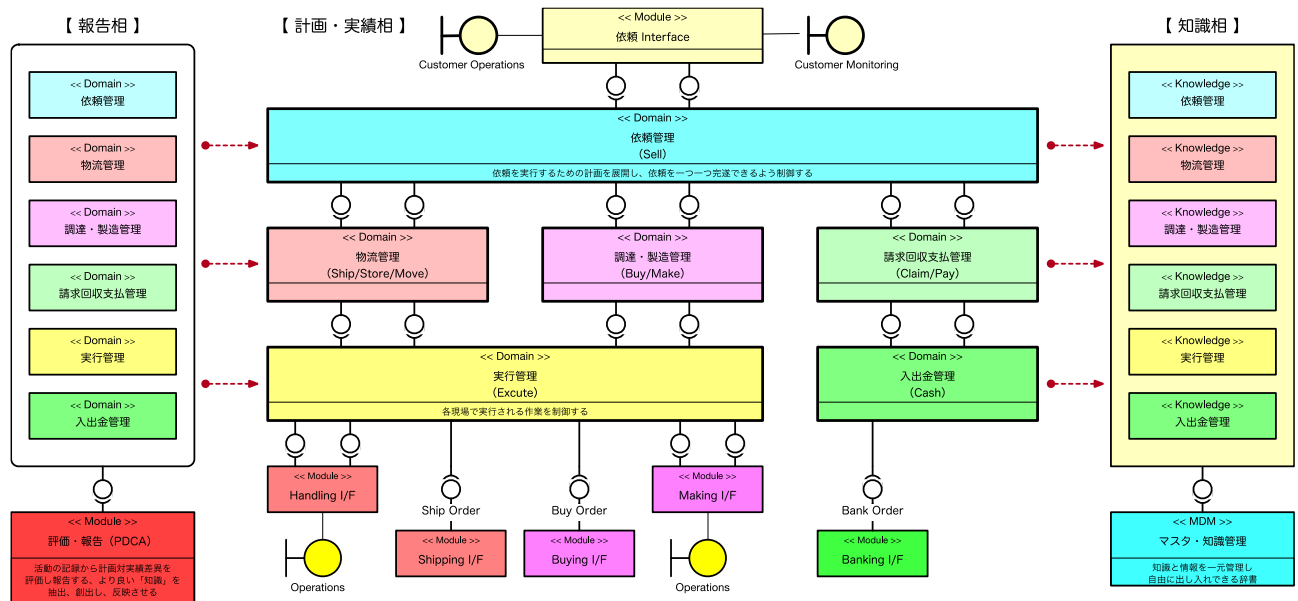
オペレーショナルな業務は、計画（予定）を立てて、それに基づいて実行し、結果を実績として記録し、予実格差を分析して、製品を修正する、と同時に知識を更新する、という「局面」から成ります（PDCA）。

前項の“管理階層”を上記の局面に沿って業務の相に分けます。

- 知識相
  - 依頼の遂行のための実行計画を立案するための基礎情報です
- 計画・実績相
  - 企業活動の現在と未来の生きている情報です
  - 指示直下の管理階層に指示を与え、その実績を受け取ります
  - フィードフォワードのための情報です
  - 今現在の状況の変化を受けて、変化する少しだけ未来の状況を予測します
- 報告相
  - 計画・実績相で発生した情報を蓄積し、活動を評価するための過去の情報です

### 3.3.3. 一般標準モデル

一般的な企業活動を以下の通り、標準モデルとして表現できます。



### 3. 4. 業務プロセスの設計

業務プロセスは、3.3 項の全体基本構造のドメイン分割された各ドメイン上で稼働します。ドメイン間は疎結合され、ドメイン間は **Customer-Performer の原則** のプロトコルでやりとりします。

#### (1) 業務プロセスの原理・原則

業務プロセスの設計では、以下の原理・原則を考慮します。

##### ① PD マネジメント、計画・実行

- ・ 仕事は、最初に計画を作り、実績を記録しながら進める
- ・ 現実には計画通りには進まないことを前提とし、リアルタイムに予実差異を評価し計画を都度見直す

##### ② Customer-Performer の原則

- ・ 仕事は、顧客からの「依頼」に基づき「実行」し、最終的に「実績」を顧客に戻し、「検収」をうけることで完遂する（別項で詳細説明）

##### ③ リアルタイム・一個流し

- ・ 仕事は、一つ一つ滞りなく、責任もって確実に遂行する
  - ・ 顧客からの依頼一つ一つに対して、やるべきことの始まりから終わりまで、責任を持って確実に実行することを目的とする。
  - ・ ビジネスからの要請に迅速、且つ柔軟に対応する
  - ・ 常に変化する環境要因に対し、直ちに変化に対応した「計画」に変え、それに従って仕事を完遂できるようになることを目的とする。

#### (2) Customer-Performer の原則：業務委託関係の対話モデル

依頼者 (Customer) と受託者 (Performer) の対話は 4 つのフェーズにモデル化できます。

1. Customer は Performer に業務を「依頼」する
2. Performer は受託条件を確認して「約束」する
3. Performer は依頼内容に基づいて「実行」し、その結果を「報告」する
4. Customer はその結果に満足すれば「検収」する

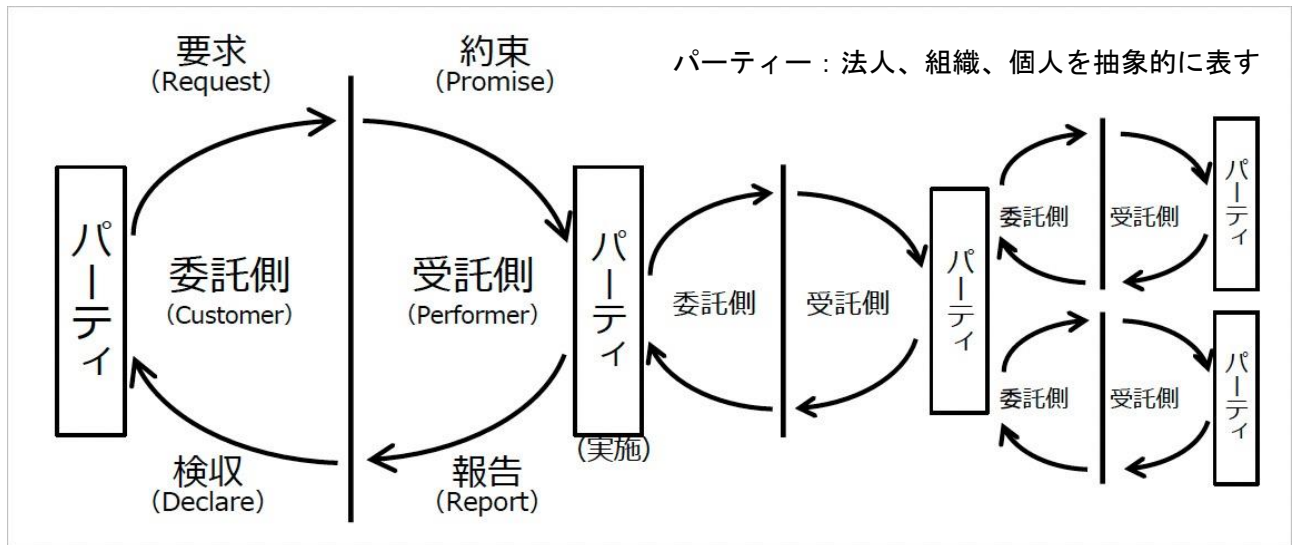


図 3.4-1 ; Customer-Performer の原則 ; 出典 JIS B 3915

### (3) 業務プロセス

業務活動全体の流れを対極的に把握し検討するために、プロセス図を用います。

- プロセス図では、一連の小さなプロセスをまとめて一つの「プロセス」として表します。その具体的な中身である小さなプロセスの順序は隠蔽します。
- 「プロセス」は「処理」なので、入力と出力を持ちます。この入出力される「情報」と「リソース」および「ゴール」だけを表すようにします。
- どのプロセスに問題があるのか、どう改善したり順番を組み替えたりしたらよいかといった検討がしやすくなります。

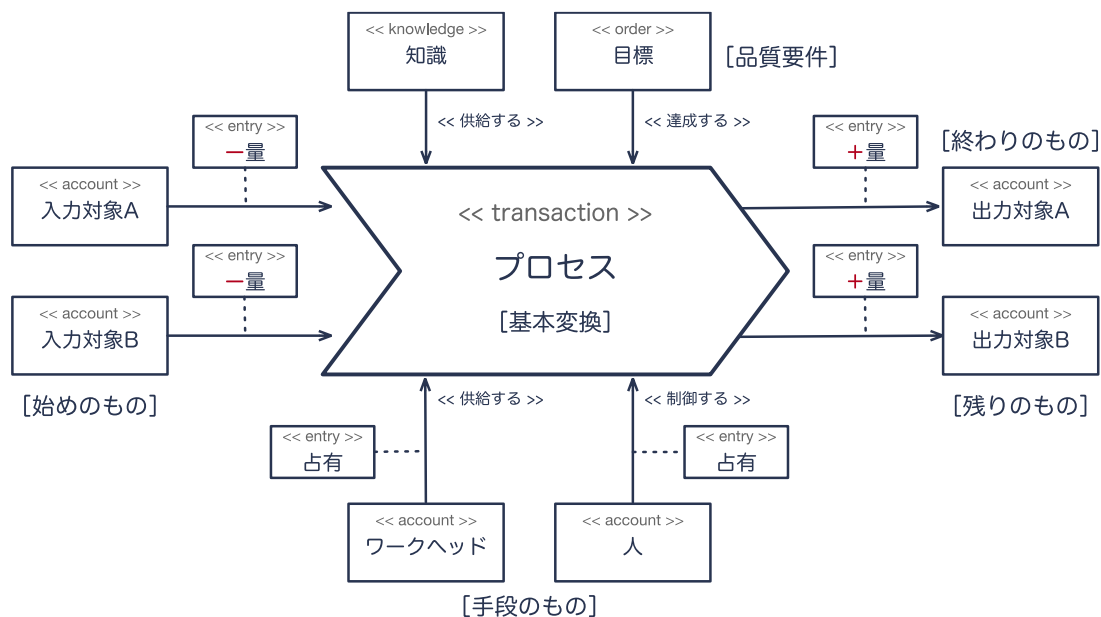


図 3.4-2：仕事の構造：業務活動を表すプロセス図

#### (4) 業務プロセスの標準モデル

業務プロセスの三つの原理・原則を使うことで、業務プロセス・業務フローを標準化します。

標準化の意味は、従来、業務プロセス・業務フローは、業者の「要件定義」に代表されるように、表記法こそある程度決まっていますが、実際には作成者への依存が大きく、なんでもありの状況だったからです。そのため、他人には理解し難く、且つ、再利用もほとんど不可能な状態でした（再度作る方が早い）。

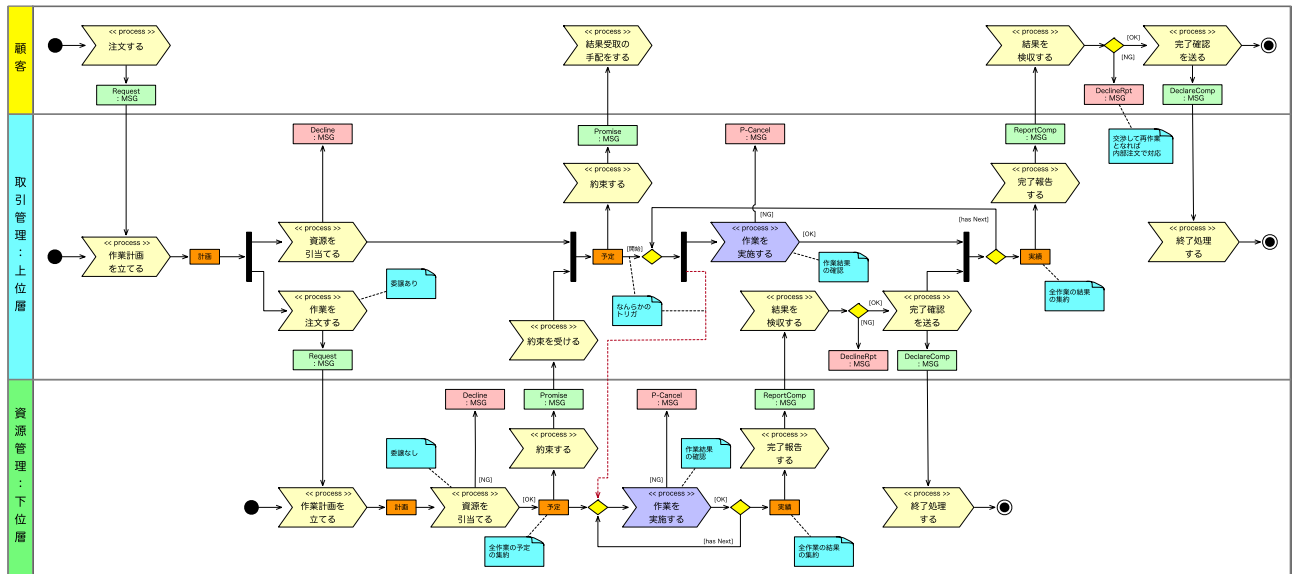


図 3.4-2 ; 業務プロセスの標準モデル

#### 3.5. ユースケース

ユースケースは“機能”を表示します。

ここでは、企業よくみられる営業事務処理を集約したセンター機能の業務フローの例を下図に示します。業務フローを作る「活動」に、その「活動」で使えると便利または効率的だと「発明」された【usecase】が紐づけられます。

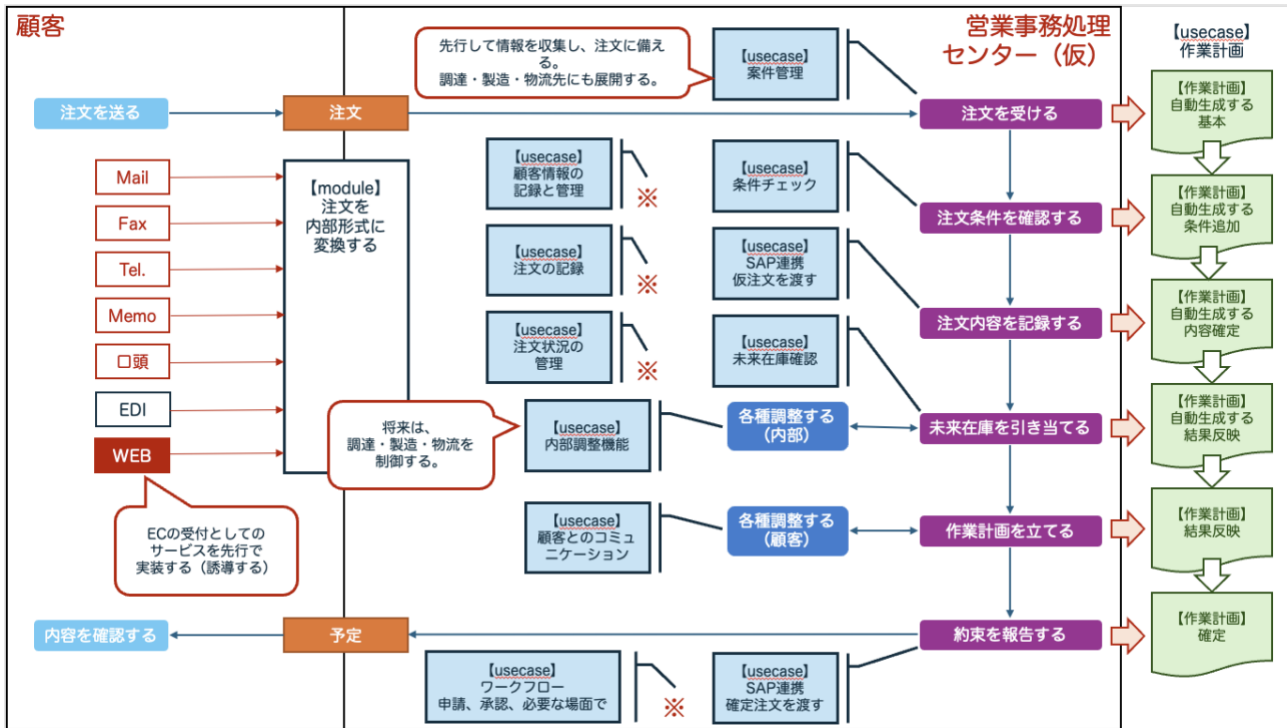


図. 3.5-1；営業事務処理センター（仮）業務フローと USECASE

### 3.5. ドメインモデル；データアーキテクチャ

#### （１）ドメインモデルと“基幹システム”の定義

ドメインモデルは、データアーキテクチャであることから、そのドメインモデルを実装した“情報システム”の性質を決めます。弊社は、このドメインモデルを実装した情報システムを“基幹システム”として以下の通り定義しています。

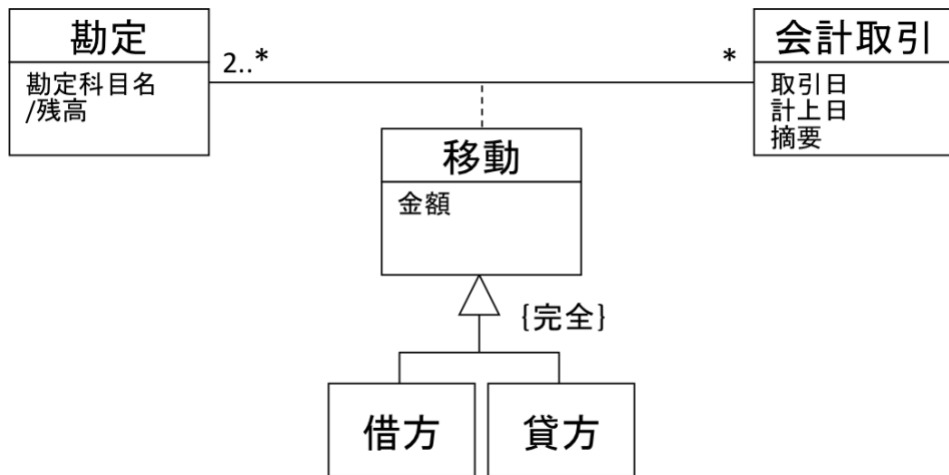
##### 定義；業務活動の事実を記録する

- ・ 組織活動の目的をより円滑に（早く、安全に、誤りなく、容易に）達成できるよう支援するために、組織活動そのもの（素活動と事実）をモデル化する。
  - ・ 組織マネージャ（あるいはその代理人としての分析者）の関心に基づく実世界の解釈を反映する。
- ➡ 実世界の「モノ」ではなく、事物についての理解や情報を扱う。

#### （２）ドメインモデルのデータ構造；勘定パターン

この、業務活動の事実を記録するドメインモデルとして「拡張勘定パターン」を使います。勘定パターンは、複式簿記のデータモデルであり一般的に会計システムで使われます。

複式簿記は、お金の取引（活動）を表現できますが、これを拡張することで、全ての取引（活動）を表現できるようにします。下図に、勘定パターンの概念モデルを示します。

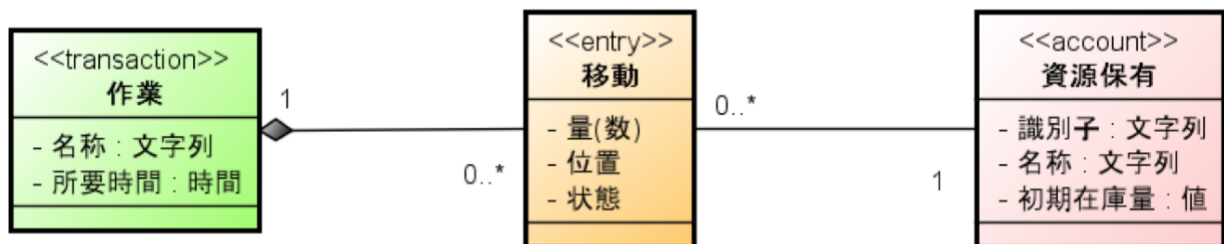


#### 複式簿記仕訳と同じデータ構造

- 企業の活動を「取引」として定義
- 業務プロセス、業務フローで表す「仕事」

#### 活動に必要な資源を「勘定科目」として定義

- 資源：時間、金、人、製品、商品、サービス、設備、原材料・・・



## 4. 支援作業について

### 4.1. 支援作業

「要求設計」の実施と「都市計画」の策定

※) 本書、1～2項を参照ください。

尚、本件は、準委任契約であり、納品物そのものを保証するものではありません

### 4.2. ヒアリング

- 1日2セッション（2h 目処）／週2回／2ヶ月＝25～30人・部署（対象はご相談）

- 原則上田1名で実施（録音します）

※) IT に関するご相談もメールその他でご対応します

### 4.3. プロジェクトミーティング

- 隔週で進捗報告、およびコンサルティング仕様の説明を実施します

※) プロジェクトメンバの選任をお願いします

- プロジェクトオーナー；理事長

- プロジェクトリーダー；選任をお願いします

### 4.4. 期間

- 3ヶ月とします

※) 3ヶ月で何かをコミットするわけではありませんが、経験上特に要求設計では3ヶ月程度が適度だと考えています。四半期という区切りに対応できることもありますが、要求設計の性格上、短いと浅すぎます、また、逆に、長く過度に詳細にする意味はあまりありません。

以上



## 添付資料：都市計画：基本計画と実施計画の目次例

### プロジェクト基本計画書 Version 01.00

#### 目次：

1. はじめに	4
1.1. 本書の位置付け	4
1.2. 前提となるインプット	4
1.3. WBS とフェーズ	5
2. 目的・目標及び達成基準	6
2.1. 目的	6
2.2. ○○の目標	6
2.2.1. ○○の経営課題と対策	6
2.2.2. スコープを定義するための優先軸	7
2.3. マイルストーン；稼働時期	8
3. 想定されるリスクと対処	9
3.1. 「期間が短い」	9
3.2. 「システム開発の経験が無い」	9
4. 基本方針	10
4.1. 基本方針	10
4.2. ミニマムスタートの意味と今後のフェーズへの影響	11
4.2.1. フェーズへの影響	11
4.2.2. アーキテクチャ；大きく育つ拡張性	11
4.2.3. IT ガバナンス；体制による継続性	11
4.3. 開発アプローチ；適応型アプローチ*注	11
5. 要約スケジュール	12
5.1. プロジェクト期間	12
5.2. フェーズ毎のスケジュール想定	12
6. ステークホルダーとプロジェクト体制	13
7. 要約予算	13

### 基幹システム構築 要求設計 Phase2；要求仕様の策定

#### 実施計画書 Version 01.00

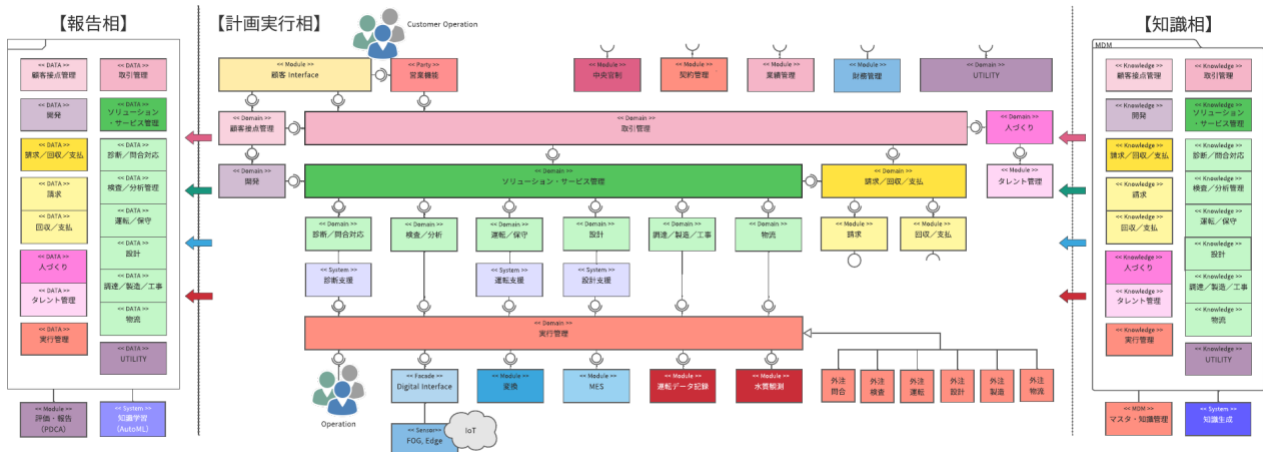
#### 目次：

1. はじめに	5
1.1. 本書の位置付け	5
1.2. 前提となるインプット	5
1.3. フェーズと WBS	6
2. 目的・目標の確認	7
2.1. 目的の確認	7
2.2. 目標と達成基準	7
2.2.1. 目標；要求仕様の策定	7
2.2.2. 達成基準	7
3. 基本方針	9

3.1. ミニマムスタート	9
3.2. 適応型アプローチ	9
3.3. 対象領域を定める優先軸	9
4. 要求設計 Phase2；要求仕様の策定	10
4.1. 進め方	10
(1) 工程	10
(2) 体制	10
4.2. 要求設計の作業手順	12
4.3. 要求設計 Phase2 の WBS と要素	13
4.3.1. 「要求設計 Phase2；要求仕様策定」の WBS；再掲	13
4.3.2. WBS 要素	13
4.4. ミニマムスタートでの対象領域の考え方	16
4.4.1. 全体構造のシンプル化；サブシステム分割	16
4.4.2. ドメイン（サブシステム）集約の考え方	17
4.4.3. 「生産管理」での考え方	18
4.4.4. 「依頼」について	19
4.5. 全社展開への拡張について	21
(1) ドメインについて	21
(2) 「生産管理」の個別性について	21
4.6. 結果データの記録に関する一案	22
5. 要約スケジュール	23
付録. 開発へ向けて	25
1. 開発に向けて準備する	25
1.1. “開発実施計画”を策定する	25
1.2. 開発業者を選定する	25
<b>2. 体制の立上と開発着手</b>	<b>27</b>
2.1. 社内体制の立上げ	27
2.2. PMO とアーキテクチャチームの設置	27
(1) PMO	27
(2) アーキテクチャチーム	27
2.3. 開発体制の立ち上げ	27
3. 開発	28
3.1. 開発を進める	28
3.2. プロジェクトマネジメント	28
4. 次フェーズの展開	28

添付資料：作成物例

### ① 全体構造図：アーキテクチャ図例



## ② 業務プロセス；契約

