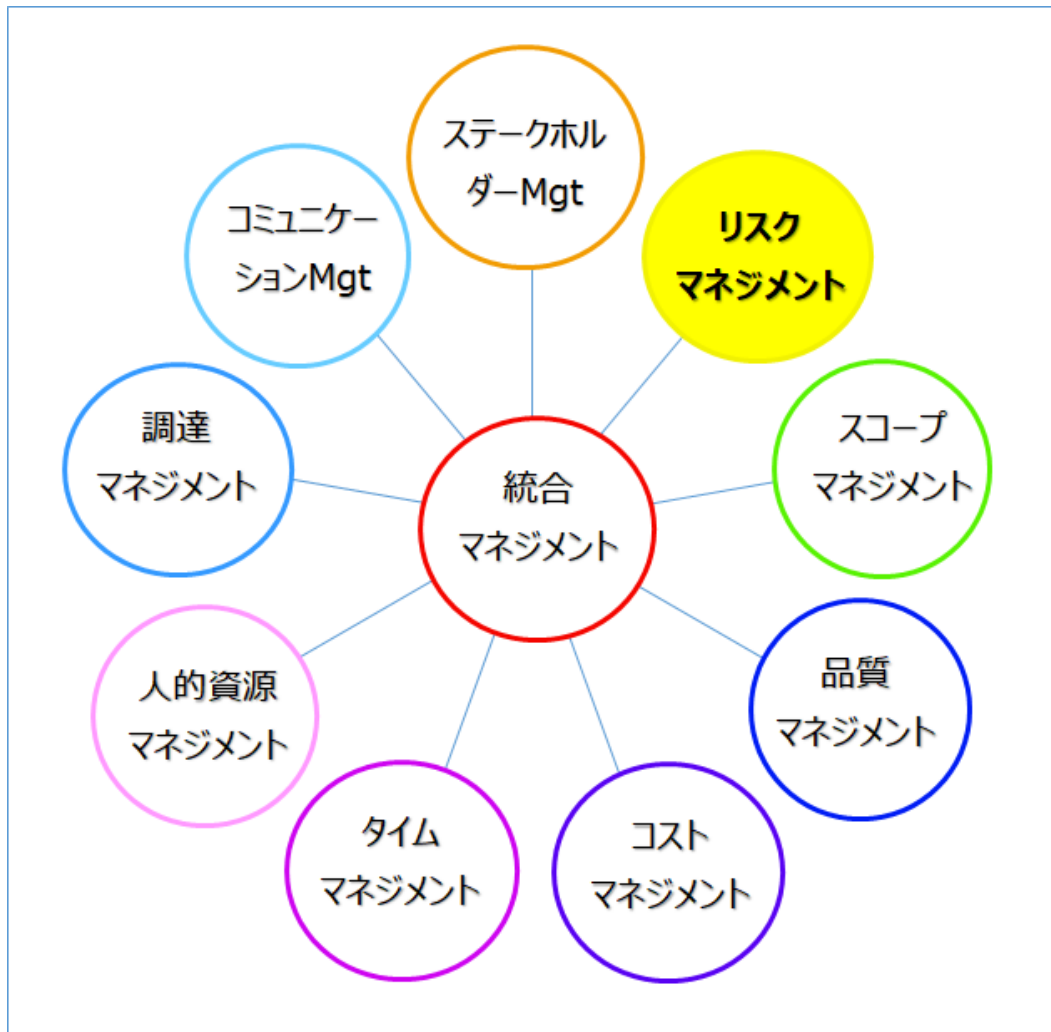


実用プロマネ わかる現場の

# リスクマネジメント





## 目次

### 第1章

## プロジェクト・マネジメントにおける問題とリスク p 1

- 1-1. リスクマネジメントの役割 p 1
- 1-2. プロジェクトマネジメントにおける主な問題 p 2
- 1-3. プロジェクトマネジメントにおけるリスク p 3
  - ◆【プロジェクトマネジメントリスク一覧】 p 3
- 1-4. プロセス管理 p 4
  - ▷プロセス管理表について p 5

### 第2章

## 事前準備工程のリスク p 6

- 2-1. 顧客情報の収集 p 6
  - ◎顧客が開発側に何を期待しているのかの情報 p 7
  - ◎開発側が知りたい情報の入手 p 8
  - ◎社外協力会社の情報 p 9
- 2-2. 未経験エリアへの対応準備 p 10
- 2-3. 事前準備工程のリスク p 11
  - 【事前準備工程のリスク管理表】 p 11

- 3-1. 早期の仕様凍結 p 1 2
  - ◎あいまいな要求仕様と無理な短納期・低コスト p 1 2
  - ◎早期の仕様凍結 p 1 3
    - ◆【早期仕様凍結のチェックポイント】 p 1 3
    - ▷先行開発着手の得失について p 1 4
  - ◎仕様の理解 p 1 5
    - ◆【仕様凍結のチェックポイント】 p 1 5
  - ◎仕様調査 p 1 6
    - ◆【仕様調査 のチェックポイント】 p 1 6
    - ▷なぜそんなに大人しいの？ p 1 6
  - ◎要求仕様書の記述項目 p 1 7
    - 【要求仕様書の記述項目例】 p 1 7
  - ◎要求仕様書精度の簡易チェック法 p 1 9
  - ◎要求仕様書の詳細チェックリスト p 1 9
  - ◎要求仕様書の精度検証 p 2 2
    - 【要求仕様書 採点表】 p 2 2
- 3-2. 要求仕様に関するリスク p 2 3
  - 【要件定義工程のリスク管理表】 p 2 3
- 3-3. 顧客要求の重要度順位の設定 p 2 4
  - 【開発作業における優先順位の決め方】 p 2 5
  - 【優先順位決定に必要な3つの視点】 p 2 5
  - 【開発作業における優先順位を誤らせる原因】 p 2 5
- 3-4. 見積りにおけるリスク p 2 6
  - ◎ソフトウェアの価格 p 2 6
  - ◎見積り回答書に対する認識 p 2 7
  - ◎受注者にとっての見積り回答書の重要性 p 2 7
  - ◎見積り回答書の品質 p 2 8
    - ▷見積り回答書の形式的なチェック p 2 8
    - ◆【見積り回答書の形式的なチェックリスト】 p 2 8
    - ▷見積り精度の向上 p 2 9
    - ◆【見積り精度向上のチェックリスト】 p 2 9
  - ◎見積り方式 p 3 0
  - ◎見積りにおけるリスク p 3 1

- 【見積りリスク】 p 3 1
- 【見積りリスクの回避対策】 p 3 1
- ▷見積りリスク回避対策の実行に当たって p 3 2
- ◎概算見積り p 3 3
  - 【概算見積りの方法】 p 3 3
- ◎見積り回答書 p 3 4
- 3 – 5. プロジェクト体制の構築 p 3 5
  - ◎プロジェクト体制構築の要件 p 3 5
    - ◆【プロジェクト体制構築の要件】 p 3 5
    - ◆【優れた開発チームの能力特徴】 p 3 6
  - ◎開発体制不備による失敗リスク p 3 6
  - ◎外注開発体制の構築 p 3 7
    - 【外注会社評価項目】 p 3 7
  - ◎統合的なプロジェクト体制の構築 p 3 8
    - ▷統合管理および統合コミュニケーション の欠落 p 3 8
- 3 – 6. 開発環境の手配 p 3 9

## 第4章

### 実行工程のリスク p 4 0

- 4 – 1. リスクの解消 p 4 0
  - ◎リスクと課題の違い p 4 0
  - ◎プロジェクトにおけるリスクの探し方 p 4 1
  - ◎三つの時点で押さえるリスク p 4 2
  - ◎開発現場にこそがるリスクの事例 p 4 3
    - ◆【現場におけるリスクの事例】 p 4 3
    - 【設計工程のリスク管理表】 p 4 5
  - ◎見えやすいリスクと見えにくいリスク p 4 6
  - ◎失敗に備えるコンティンジェンシープラン p 4 7
- 4 – 2. ドキュメントベース開発の実行 p 4 9
  - ◎ソフトウェア開発に必要なドキュメント p 4 9
  - ◎開発ドキュメントの現状 p 5 0
  - ◎ドキュメントの不良が招いた失敗事例 p 5 1
  - ◎ドキュメント精度の簡易チェック法 p 5 2
  - ◎リアルタイムなドキュメントのメンテナンスが必要な訳 p 5 3
  - ◎開発管理表の概要 p 5 4

#### 4-3. 優先順位ベース開発の実行 p 56

##### ◎開発業務における実行の優先順位 p 56

▷要求仕様を顧客価値の重要度順に整理すること p 56

▷顧客価値の重要度順に仕様を凍結していくこと p 57

▷集中仕様凍結会議の実施 p 57

▷顧客価値の重要度順に凍結した仕様から開発着手を行うこと p 57

##### ◎QCDには優先順位はつけられない p 58

## 第5章

### 管理工程のリスク p 59

#### 5-1. 要求仕様の変更管理 p 59

##### ◎ベースラインの意味 p 59

##### ◎変更管理が行われにくい理由 p 60

##### ◎要求仕様の変更管理不在がもたらす災い p 61

##### ◎仕様変更管理表がもたらす効果 p 62

付録図表 p 63

チェックリスト一覧表 p 70

著者プロフィール等 巻末

# 第1章

## プロジェクトマネジメントにおける問題とリスク

### 1-1. リスクマネジメントの役割

#### ◎リスクマネジメントの役割

プロジェクトへのリスクの影響が最小限になるようにマネジメントすること、すなわち開発プロセスに内在するリスクを早期に発見し、解消することでプロジェクトのQCD目標を達成させること。

【リスクマネジメント】（図1-1）



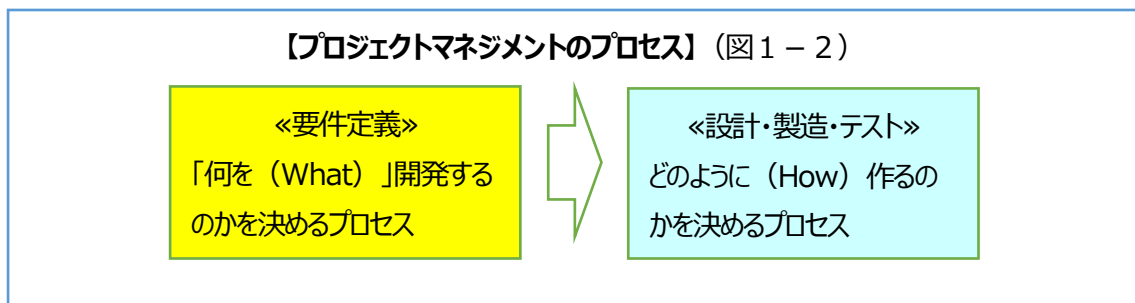
## 1 - 2. プロジェクトマネジメントにおける主な問題

プロマネにおけるプロジェクトマネジメントの問題は大きく分けて次の二点に集約されます。

- ① プロジェクトマネジメントの全体像を理解していないこと。
- ② プロジェクトマネジメントの実行の方法を理解していないこと。

要するに何を (What) どのような方法 (How) で実行して良いのか理解できていないところに問題の真因があります。この問題は開発実務における、確定された要求仕様 (What) に基づいて設計・製造・評価 (How) を行う場面に置いても同様の問題を引き起こしています。

何を実行すべきかが分かっているならば、実行段階には進めないことは当たり前で、何を (What) すべきかを最初に理解しておくことが最も重要なことと言えます。



何をすべきか、さらにそれをどのように処理すべきかを考え実行する基本的な力は、結局、目の前の現実や事実を自分の目で直視し、それが意味するところを自分自身で解釈・判断し、それをどのような方法で解決するのかを自分で決定し、自分で行動を起こすという、自律性の力の発揮によるものと言えます。この自律性の発揮こそが、全ての問題を解消する原点となっています。

### ◎プロジェクトマネジメントにおける主要な問題点

- ◎プロマネが、プロジェクトマネジメントの全体像を理解していないこと。
- ◎プロマネが、プロジェクトマネジメントの実行の方法を理解していないこと。

PM

<sup>1</sup> **自律性** 直面する問題に対して自分自身の目で確かめ、状況を把握・判断し、実行に移す能力のこと。反対語は他人依存性。



### 1 - 3. プロジェクトマネジメントにおけるリスク

プロマネがプロジェクトをマネジメントするということは、下記のリスクを解消するということに他ならず、これらのリスク回避活動がプロマネの役割であり職責そのものだと言えます。

#### ◆【プロジェクトマネジメントリスク一覧】

- 外注まで含めた統合的プロジェクトマネジメント（品質・進捗・コスト・人材管理）の不在
- マルチベンダー下における責任分担の不明確さ
- 開発体制の不備（能力不足の人員配置）
- 上位マネジメントとの意思疎通不良（孤立無援によるリスクの増大）
- 無理な開発費減額要求の無条件な受入れ
- 無理な納期短縮要求の無条件な受入れ
- 見積りのルール破り（承認ルーチンの無視）
- あいまいな開発スコープおよび要求仕様の放置
- 仕様変更管理ルールの不在
- 条件なしの規模の大きな機能追加要求の受け入れ
- 利益確保を優先し、品質の確保を怠る
- 不適切な事前着手
- 知識のない分野の開発請負
- 新言語採用にあたっての準備不足
- 新技術採用における準備不足
- 既存資産流用の可否判断ミス
- 新旧システムの同時並行開発
- 他社ブラックボックス・ソフトウェアの事前検証不足
- システムの劣化あるいはスパゲッティ化の放置
- 納期厳守で不良品を出荷する
- 納期遅延による必要工程の手抜きないしはスキップの黙認ないしは指示

上記以外にもマネジメントに関するリスクは多く存在しますが、いずれにしても大部分のリスクは開発の初期工程に集中しており、これらのリスクを解消することがプロマネを含めたマネジメントの仕事だとも言えます。

◎プロジェクトをマネジメントするとは、  
プロジェクトのリスクを解消すること。

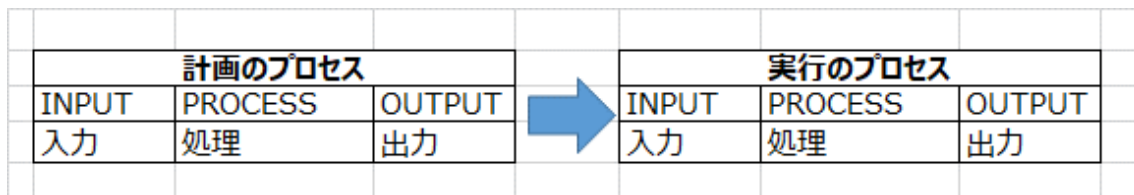
PM

## 1-4. プロセス管理

プロジェクトをマネジメントするということは、プロセスをマネジメントすることであり、それは必要なプロセスをタイムリーに確実に実行することによって達成されます。

開発プロジェクトは複数の工程すなわち複数のプロセスで構成されています。各プロセスは INPUT—PROCESS—OUTPUT で構成されており、前のプロセスの OUTPUT は次のプロセスの INPUT となり全てのプロセスは互いに接続されています。プロジェクトマネジメントにおける計画のプロセスと実行のプロセスの連鎖を示すと次のようになります。

### 【連鎖するプロセス】（図1-4-1）



あるプロセスの INPUT は前のプロセスの OUTPUT に相当します。プロジェクトを成功に導くためには、各プロセスにおける OUTPUT が約束通りのもので、約束された時期に提供される必要があります。

◎ **Input – Process – Output が適切な時期に正しく実行されれば、プロジェクトは必ず成功する。**

PM

これらの複数のプロセスにおける主要なイベント<sup>2</sup>および成果物を時系列的に並べたものがプロセス管理表と呼ばれるものです。

多くの品質問題の本質は、「段取り 7 分の後 3 分」と言われる通り、仕事の段取りの悪さにあります。段取りとは、すなわちプロセス管理のことです。

プロセス管理における改善すべき主な問題点として次のようなものがあります。

### 【プロセス管理の問題点】

- ・ 各工程における実行手順忘れ
- ・ 各工程における手抜き行為
- ・ 人依存・経験依存のバラバラな開発手順
- ・ 視覚化されていない開発手順

これらの問題は、通常のガントチャートなどのスケジュール進捗表やレビューなどでは発見することが難しく、以下の対策が必要です。

<sup>2</sup> イベント 出来事、行事のこと。開発工程の節目ごとにおいて行われる、成果物などの受け渡しのことを指す。

## 【プロセス管理の改善】

- ・ 標準プロセス管理表の作成（開発業務手順のビジュアル化）
- ・ 標準プロセス管理表に基づく開発手順の実行（開発業務品質の確保）
- ・ 標準プロセス管理表実行のレビュー
- ・ プロセス管理表に ISO<sup>3</sup>・CMMI 項目を絡めたチェック機能の盛り込み強化



## プロセス管理表について

プロセス管理には適切なプロセス管理表が必要になります。プロセス管理表の適切な運用によって、開発に起こりがちな実行忘れ、手抜きなどのリスクを排除することが可能になります。

プロセス管理表<sup>4</sup>は E X C E L 表の横軸に、各プロセスの主だった成果物ないしはイベントについて手順項目・作業内容・担当者名・成果物名・チェック欄・実施予定日・実施日などの記述欄を設け、縦軸は各プロセス区分の時系列の早いもの順の並びとなった表で表されています。

下記はその一部分を示したものです。

## 【プロセス管理表（部分）】（表 1 - 4 - 1）

【プロセス管理表】（サンプル）				* P L ( o r P M ) , S P L ( o r S P M )						
NO	手 順	進 捗	作 業	担 当 者	成 果 物	担 当 チェック	P L チェック	実 施 予 定 日	実 施 日	レ ビ ュ ー 結 果
事前準備	1		顧客の要求内容・予算・納期等の事前把握	開発部	・調査報告書					
	2		新ソフトウェア・新ハードウェア等に対する技術知識のマスターおよびプロトタイプによる事前調査	開発部	・調査報告書					
計画プロセス	1		仕様、スケジュール、要員検討、見積依頼	P L , S P L	・見積回答書 ・リスク管理表					
	2		要求仕様の明確化	P L	・要求事項の明確化チェックリスト作成					
	3		SEとの仕様打ち合わせ	S E , P L , 開発メンバー	・議事録 / 要求仕様書の承認					
	4		開発に対する有効な歯止め策を大、小日程の作成	P L , S P L	・歯止めチェックリスト					
	5		スケジュール	P L , S P L	・大日程表、小日程表					
	6		プロジェクト品質目標の作成	P L , S P L	・プロジェクト品質目標					
	7		プロジェクト計画書の作成 リスク管理表更新	P L , S P L	・プロジェクト計画書					
			プロジェクト計画レビュー 開発開始会議	S Q A グループメンバー、P M , P L	・レビュー記録					
	8		外注見積委託作成	P L , S P L	・見積書					
	9		開発管理	P L , S P L	・プロセス管理表 ・スケジュール表 ・課題管理表 ・規模進捗					
10		デザインレビュー④	要求事項の明確化チェックリスト	S E → 開発部	・要求仕様の採点確定					

全プロセスを表現したプロセス管理表のサンプルは「付録図表 1. プロセス管理表」を参照のこと。

◎ プロセス管理表なしではプロジェクトマネジメントはできない。

PM

<sup>3</sup> ISO 国際標準化機構。ISO 9001 は、品質マネジメントの国際標準規格。ISO21500 は、プロジェクトマネジメントの国際標準規格。

<sup>4</sup> プロセス管理表 開発における各工程の流れに沿って、開発の主要なイベントおよび成果物を時系列順に並べた管理表で、各工程における実行手順忘れ・手抜き行為などを防止するためのもの。

## 第2章

### 事前準備工程のリスク

#### 2-1. 顧客情報の収集

IT開発を情報戦だと認識している人は意外に少ないと思われます。未知のものをできるだけ正確に把握するためには、事前調査に時間をかけ、そこで得られた情報を小まめに記録し、蓄積しておくことが必要となります。

開発にとって重要である顧客情報、とりわけ顧客特有の**思考・行動様式**や納期・要求仕様・発注予算に関する情報は、プロジェクトの生命線ともいえる情報ですが、プロマネをはじめとした開発関係者は相も変わらず待ちの姿勢が多く、これらの情報をいち早くキャッチするための、顧客との**直接コミュニケーション**<sup>6</sup>がおろそかになっている場合が多いのではないのでしょうか。これらのキー情報をどれだけ早い時点でキャッチアップできるかが、プロジェクトの勝敗を決める重要な**ファクター**<sup>7</sup>になります。プロマネやプロジェクトリーダーにおいては開発開始前の段階で、自分自身が直接行動することにより、これらの情報を収集しておく必要があります。

準備

#### ◎事前入手が必須な情報は次の四つ、

1. 顧客特有の思考・行動様式	①コストカット重視 or 品質重視 ②情緒的 or 合理的 ③硬直的 or 柔軟的 ④競合他社追従的 or 独自性重視、など
2. 納期情報	決定済 or 未決定、短納期 or 妥当な納期
3. 要求仕様の骨子	骨子が明確 or 不明確、仕様決定力が強い or 弱い
4. 発注予算情報	ある程度判明 or 不明

☆上記情報はプロジェクトの成否を分ける重要なリスク情報であり、不都合な情報であった場合は早めの対策が必要となります。

<sup>5</sup> **思考・行動様式** 人や組織における、その成長過程において習得された、特有なものの考え方および行動。

<sup>6</sup> **直接コミュニケーション** 実際に顔と顔を突き合わせた対面コミュニケーションのこと。ダイレクトコミュニケーション、フェイス トウ フェイス コミュニケーションとも言われる。

<sup>7</sup> **ファクター** ある結果を生み出す要因。



## 顧客が開発側に何を期待しているのかの情報

第一に入手が必要な情報は、顧客が開発側に何を期待しているのかを示す顧客特有の思考・行動様式に関する情報です。開発会社が接する顧客側の部署は、システム部とか情報企画部とか呼ばれる部署が一般的ですが、それらの部署はその会社特有の思考・行動様式を強く反映しています。プロジェクトを成功裏に終了するためには、彼らがどのような思考様式や行動様式をもっているのかを、開発が始まる前の受注活動の中で素早く理解しておく必要があります。

顧客側の思考・行動の代表的なパターンを示します。

### ◎身勝手な顧客の例

できるだけ安い金額と短納期でたくさんの仕様を盛り込み、良い品質のものを希望する顧客は、開発側にとっては困ったものですが、自分が車や家を購入する場合を振り返ってみると、一方的に非難することもできません。これが日本の消費者の典型的な、商品購入における思考・行動のパターンなのです。経験上このような顧客は全顧客のうちの約9割を占めていると思われます。このような顧客に対して何の対策もなしに無防備に受注・開発を行うと以下のような問題が発生してくるでしょう。

#### ①仕様の範囲（スコープ）がいつまでも確定されない

一旦決まった仕様ですら二転三転し、開発工程の後半になっても仕様変更や追加が止まらない。

#### ②プロジェクトが危機に陥る

約束した開発期間や開発費は、開発半ばで消費し尽くしてしまう状況に陥る。

このような状況を回避するためには、事前準備のプロセス期間中に顧客の思考・行動に関する上記のリスクを排除しておく必要があります。下記の[リスクヘッジ](#)<sup>8</sup>策が有効です。

#### 【身勝手な顧客に対する防衛策】

1. 営業部署との連携による、顧客との密接なコミュニケーションを実行すること。
2. “新システム一式、仕様は打合せによる”などという基幹仕様未定な状態で受注はしないこと。
3. 仕様凍結の最終期限を相互で約束しておくこと。
4. 仕様凍結後の仕様変更・追加は別途見積り（開発費および開発期間）とすること。
5. 短納期・低開発費を強く迫る顧客に対しては、開発済みの標準的な仕様の提案を行うこと。

身勝手な顧客に対する防衛策によって、顧客を妥当な要求に導き、開発側は妥当な開発期間・開発費を提示することで、両者ともに納得できる形で開発を進めることが可能になります。

<sup>8</sup> リスクヘッジ (Risk Hedge) 損失・失敗の防衛策のこと。

さらにこれらのリスクヘッジは開発プロジェクトにおける最大のリスクである、要求仕様問題・見積り問題および開発行為のやり直しという三つのリスクの排除に大きく貢献します。

### 【プロジェクトにおける三大リスク】

1. いつまでも決まらない要求仕様

2. 精度が低い見積り

3. 開発行為のやり直し



### 開発側が知りたい情報の入手

開発側が最も知りたい情報は次の三つになります。

1. 納期情報

2. 要求仕様の骨子

3. 発注予算情報

これらの情報は、事前準備の段階での入手が必要です。

あらかじめ顧客のこれらの情報を大まかにでも入手していれば、顧客との仕様検討や見積り等を開発側主導で進めることが可能になります。これらの情報の入手は以下の行動によって可能になるでしょう。

1. 営業部署と連携した、顧客との密接なコミュニケーションの実行による信頼関係の醸成。

2. 失注競合他社の提案書などの情報入手。

準備

◎顧客の情報を知れば失敗リスクの半分は消える、  
残りの半分はプロジェクト自身の開発力にある。

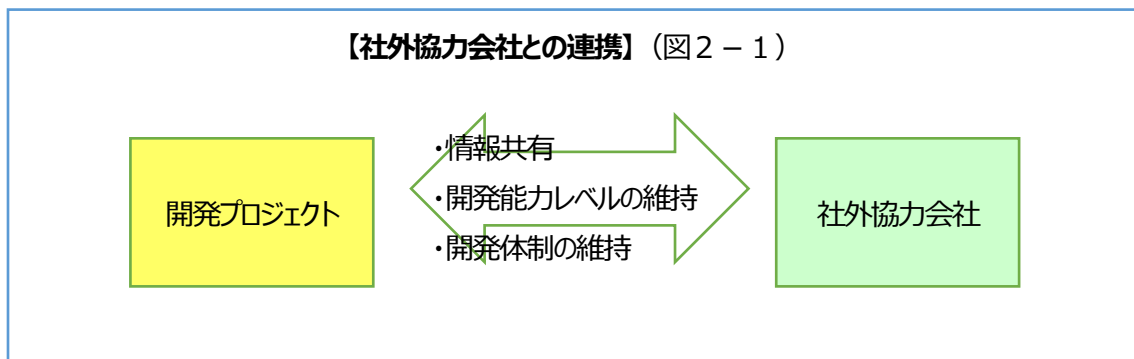


## 社外協力会社の情報

ある程度以上の規模の開発においては、社外の協力会社が開発工程のある部分を委託することも多くのプロジェクトで実行されています。プロジェクトにおける開発能力は、社内と外注でのレベルを一定以上の状態に保つ必要があります。どちらが低くてもプロジェクトを成功させることはできません。

外注化を行っているプロジェクトにおいては、常時協力会社の要員のスキルレベル情報・コスト情報・品質情報・生産性情報を入手し、それに対する評価を行い、結果をフィードバックしておく必要があります。このような活動を通して、プロジェクト開始前に必要な能力レベルの外注の人材を確保することが可能になります。

またプロジェクトの実行期間中においては、協力会社も含めた統合管理の実行は不可欠で、QCD・リスク情報や協力会社からの要望事項等のすべての情報について、日次の情報共有会議や定期的なプロジェクト統合会議などにおいての情報共有が必要です。



<sup>9</sup> プロジェクト統合会議 ステークホルダーとの定期的な情報共有会議。

## 2-2. 未経験エリアへの対応準備

顧客を初めとしたステークホルダーから事前に得られた様々な要求情報の中でも特に重要なものとしては、当該プロジェクトにとって未経験のエリアに関する情報があります。これらの未経験エリアに関する情報の入手は早ければ早いほど良く、素早くこれらの問題に対処しておく必要があります。

例えば、未経験のOS・開発言語・業種アプリケーション・ネットワーク・ハードウェアプラットフォーム・端末機器などに関する顧客要求が、プロジェクトにとって未経験ないしは経験が浅いものであった場合、プロジェクトが開始されてからこれらの知識の習得や人材・機材の用意をしていては、限られた開発期間の中でプロジェクトを無事に完遂することは不可能です。

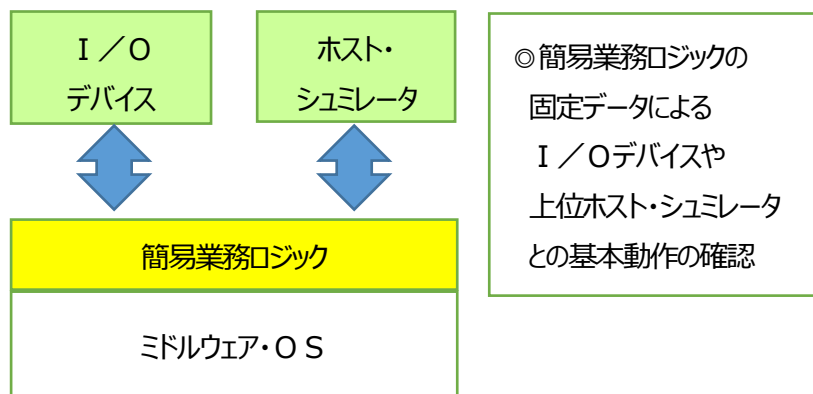
これらの未経験エリアの問題への最も効果的な対応方法は、**プロトタイプ<sup>10</sup>**の作成による事前の検証です。プロトタイプによる検証は未知のものにおける弱点・相性・性能・品質などを事前を知るためには必ず必要となるものです。予算も時間もないから、とりあえずプロジェクトを始めてから考えようというような姿勢ではプロジェクトは必ずといって良いほど失敗するでしょう。

またプロトタイプの作成および検証期間中に、未経験エリアに関する知識の習得および必要な人材・機材の手配を行っておけば、計画プロセス以降のプロセスはスムーズに進行できるでしょう。

準備

◎ 未知・未経験エリアに対する事前準備は、プロトタイプによる事前検証を。

【プロトタイプ・アプリケーションの例】(図2-2)



<sup>10</sup> **プロトタイプ** (prototype) ソフトウェア開発において、本番プログラムの作成に先立って新技術・新機能の検証および問題点の洗い出しのために仮に作成されたプログラムのこと。



### 2-3. 事前準備工程のリスク

事前準備工程におけるリスクは次の表に示した通り、非常に多岐に渡っており、後工程に被害を及ぼす前に解消しておく必要が有ります。

【事前準備工程のリスク管理表（部分）】（表2-3） 拡大表は付録図表2. を参照のこと。

プロジェクト名：		初版発行日：		更新日：		文書番号：	
リスク要因分類	QCD影響			事前準備工程リスク	✓	チェック	
	Q	C	D			検出	日
ヒトに関するリスク	① 他者依存的姿勢（自律性の放棄）	○	○	○	・リーダーシップの欠如 ・マルチベンダー下における分担責任のあいまいさ ・他社パッケージの未検証採用 ・基幹技術の選定（なりゆきまかせ） ・開発組織の自律性不足		
	② 上位マネジメントの関与不足		◎		・顧客交渉戦略・能力不足 ・不適切なプロマネの選任		
	③ ユーザーの参加・協力度不足	◎	○	○	・顧客の参加・協力度が低い		
	④ 組織能力不足（未熟な組織文化、戦略の欠如）				・不適切な事前着手（短納期） ・新旧システムの同時並行開発（開発量・時期の重複）		
	⑤ 見積り能力	◎	◎	◎	・顧客との名目だけの共同研究開発 ・能力不足のプロマネの選任 ・頻繁な更新によるシステムの劣化・スバゲッティ化		
	⑥ 要件定義能力	◎	○	○	・要件定義能力不足		
	⑦ リーダーのプロジェクトマネジメント能力（外部交渉、タイムマネジメント、現場主義、見える化能力など）	◎	◎	◎	・顧客交渉戦略・能力不足 ・不適切な事前着手 ・新旧システムの同時並行開発 ・企画能力不足 ・適時の支援依頼発信能力不足 ・問題の放置 ・未経験言語採用の準備不足 ・開発目的の誤り（受注優先） ・技術者のトレーニング不足 ・開発体制の不備（経験者・能力・知識・人数） ・未経験分野／業界参入の準備不足 ・新技術採用の準備不足 ・パッケージベース開発におけるフィット&ギャップ調査不足 ・タイムリミット管理能力不足 ・技術方式選定対応における柔軟性		
	⑧ メンバーの技術能力、ヒューマンエラー（うっかりミス）	◎	○	○	・オープンシステム知識の不足 ・開発言語能力不足 ・新技術知識不足 ・顧客業務知識不足		
	⑨ プロセス管理の有無	◎	○	○	・プロセス管理の不足		
	⑩ コミュニケーション能力（阻害・ギャップ）	◎	○	○	・顧客のパートナー化失敗 ・顧客とのコミュニケーション能力		
	⑪ 関連部署との連携不足	◎	○	○	・組織間の協調性・コミュニケーション		
モノ	⑫ ドキュメントの不備（要件定義書・設計書・チェックリスト・手順書など）	◎	○	○	・開発ガイドラインの不備（開発プロセス、設計手順書、コーディング規約、単体・総合評価テスト手順書等）		
	⑬ 開発のベースの有無	◎	◎	◎	・開発のベースの有無		
	⑭ 開発環境の不備	◎	○	○	・開発環境の不備		
カネ	⑮ 開発費不足	◎	◎	◎	・赤字受注 ・開発費不足（見積りの失敗、開発の失敗）		
	⑯ 資源投入戦略の誤り（開発費投入時期ミス）	◎	○	◎	・資源投入戦略の誤り		
情報	⑰ あいまいな開発範囲（スコープ）	◎	◎	◎	・あいまいなスコープ		
	⑱ あいまいな要求仕様	◎	◎	◎	・あいまいな要件		
	⑲ 情報の不備・不足（マネジメント情報、技術情報、過去の失敗情報）	◎	○	○	・情報の不備・不足 ・未経験分野／業界の業務知識・技術情報不足 ・新技術の技術情報不足		

## 第3章

### 計画工程のリスク

#### 3-1. 早期の仕様凍結

##### あいまいな要求仕様と無理な短納期・低コスト

プロジェクトの起点は、顧客あるいはベンダー<sup>11</sup>から提示される要求仕様および見積り依頼にあります。これはまた同時に、プロジェクトにおける問題の起点もここにあるということになります。開発の第一段階における要求仕様の品質および見積り回答の妥当性が、プロジェクトの成否を決定づけることになります。

顧客要求に関する主なリスクは次の通りです。

- あいまいでいつまでも決まらない要求仕様

回避策 ⇒ 参照：p 1 3「早期の仕様凍結」、p 1 5 仕様の理解、p 1 6 仕様調査

- 無理な短納期・低コスト

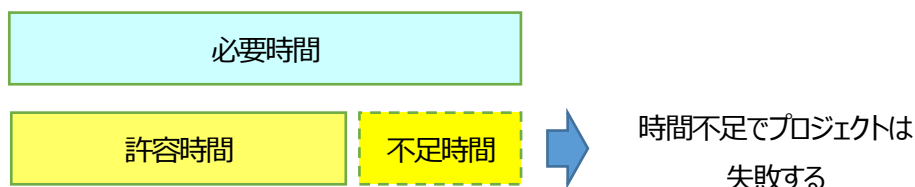
回避策 ⇒ 参照：p 7【身勝手な顧客に対する防衛策】、  
p 3 1【見積りリスクの回避策】

これらのリスクの解消法は、「仕様の早期凍結」および「妥当な納期・開発費の獲得」ですが、これを実現するためには、それ相応の仕様知識、技術力および交渉力が必要になり、その開発組織の能力レベルが上がるに従ってその成果も拡大していきます。

計画

◎プロジェクト問題の起点である要求仕様の品質は、ソフトウェアの品質と量を規定する。

##### 【必要時間と許容時間】 (図3-1-1)



<sup>11</sup> ベンダー (vendor) ソフトウェアないしはハードウェア製品のメーカーまたは販売会社のこと。ユーザー企業から発注されるソフトウェア開発の元請け企業となる場合が多い。

## 早期の仕様凍結

早期の仕様凍結のチェックポイントは次のようになります。

### ◆【早期仕様凍結のチェックポイント】

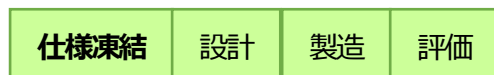
- 早期仕様凍結のために、顧客および関連各社の参加・協力の要請を行うこと。
- 仕様凍結の期限を切り、発注顧客側と受注開発側で合意しておくこと。
- 顧客・開発会社間の直接コミュニケーションによる、要求仕様の期限内凍結を行うこと。
- 集中検討会ないしは合宿等にて、短期集中的に仕様決定を行うこと。
- 受注側においても提案型仕様凍結を行うこと。
- 顧客価値<sup>12</sup>の優先度順に、仕様凍結を行うこと。
- 顧客価値の高い仕様順に、開発着手すること。
- 開発仕様の目的・背景・範囲・内容を文書にて明確化すること。
- 基幹仕様未凍結状態ないしは疑問点・不明点を残したままで、先行開発着手は行わないこと。

計画

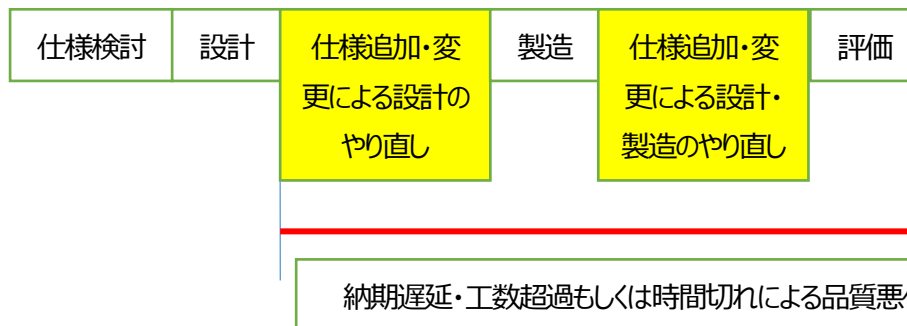
◎ 基本的な要求仕様は、設計着手前に凍結されなければならない。

### 【決まらない要求仕様をもたらす弊害】（図3-1-2）

<決まる仕様>



<決まらない仕様>



<sup>12</sup> 顧客価値 顧客がビジネス上重要視している機能やシステムのこと。



### 先行開発着手の得失について

要求仕様の決定が遅れ始めると、開発スケジュールを遅延させないために早期の仕様凍結をあきらめて**先行着手**<sup>13</sup>を始めるプロジェクトが後をたちません。それも開発の基幹仕様が決まっていなくてもかかわらず、先行着手に走ってしまっているのです。枝葉に相当するような仕様の先行開発ならば、後でやり直すとしても被害は微少で済みますが、基幹仕様の場合は、高い確率で大きなやり直しが発生します。先行着手を行う開発者たちは、何をやるのかが決まってもいない仕様を、一体どうやって開発するのでしょうか。この場合、先行着手を別の言葉で言えば**想定開発**<sup>14</sup>ということになります。

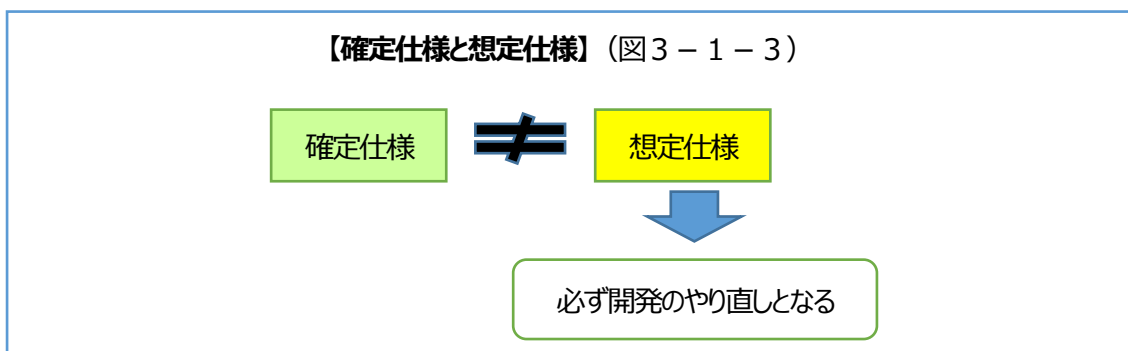
早期の仕様凍結をあきらめた開発組織は、いつしか事前着手が当たり前の行為になってしまうだけではなく、二転三転し開発工程の後半になっても決まらない顧客の要求仕様に振り回されることになってしまいます。

事前開発という想定開発によるやり直しに加えて、いつまでも決まらない要求仕様によるやり直しまでも背負うことになったプロジェクトは、必ずQ C Dに大きな傷を負うことになります。先行着手で稼いだと思っただけの時間など何の効果もなかったこととなります。

基幹仕様未凍結での開発着手を禁止するという事は、非合理的な想定開発を禁止すると同時に早期の仕様凍結に全力で取り組まなければならないということの意味しているのです。

この問題は、仕様決定の当事者になりにくい下請けのプロマネや開発リーダーをより深刻な状況に追い込むこととなります。しかしこのような状態に追い込まれないためには、下請けのマネジメントは元請けのマネジメントに対して、顧客との仕様検討の場の下請けのプロマネや開発リーダーの参加を要請する必要があります。

早期仕様凍結を本当に実現したいと思うならば、元請け側がこの提案を断る理由はどこにもないでしょう。



<sup>13</sup> **先行着手** 仕様未凍結状態で、開発側の想定仕様に基づいて開発を始めること。

<sup>14</sup> **想定開発** 確定仕様に基づかず、自分の想定仕様による開発。多くの場合、やり直しが発生する。

## 仕様の理解

優れた開発者における開発工程ごとの時間の使い方は、仕様の調査・検討である初期工程に重く、後になるに従って軽くなっています。さらに開発すべき対象をどれだけ正確に把握しているか、どれだけ早く理解しているか、がプロジェクトの成否の分かれ目になります。この勝負は見積り時および要件定義工程においてほぼ決定しています。プロジェクトを成功させる要因の重さは、“何を作るのか”が分かることで六割、“どのようを作るのか”が分かることで三割、その他要因で一割程度だと思われます。

これらの事実を直感的に理解している開発者は、見積り時において、仕様の全体像の把握および主要な仕様の把握に全力を上げています。これができれば妥当な見積りが可能となり、基本設計の概要を描くことも可能になります。要するに、進むべき地図を先に明確に描いておくのか、それとも迷いながら進むべき道を探すのかの違いです。どちらが早く、しかも正確に目標に到達できるかは一目瞭然です。

計画

◎ 仕様の全体像および主要仕様の把握は、設計工程の前に済ませておくこと。

仕様凍結にあたってのチェックポイントは次のようになります。

### ◆【仕様凍結のチェックポイント】

- まずは仕様の全体像の把握から始めること。
- 要求仕様の背景や意味を必ず理解しておくこと。
- 仕様検討の段階で要求者と徹底的な仕様検討を行うこと。
- 疑問・不明点の発掘を行い、その解消に向けて、要求者に対し積極的な行動を取ること。
- 仕様決定の Q & A<sup>15</sup>は直接対話による確認を行うこと。
- 不明なことは直ちに分かっている人・部署に聞くこと。
- 仕様検討にて新たに知り得た仕様や技術情報を、ドキュメント<sup>16</sup>によって他のメンバーに伝えること。
- 早期の仕様凍結を行うこと。
- 基幹仕様未決定で開発に着手しないこと。

### 【仕様の理解度】（図 3 - 1 - 4）

仕様の調査・検討の度合い



経験による知識の度合い

<sup>15</sup> Q & A Question & Answer 質疑応答の略。ここでは仕様の不明点および疑問点に関する質疑応答のこと。

<sup>16</sup> ドキュメント 仕様書や設計書などに限らず管理用文書、ビジネス書類、メモなども含む全ての文書のこと。

## 仕様調査

要求仕様には必ず疑問点や不明点が多く含まれています。これらの問題をすべてクリアにしなければ間違いのない仕様を決定することはできません。仕様調査にあたってのチェックポイントは次のようになります。

### ◆【仕様調査<sup>17</sup>のチェックポイント】

- 仕様骨子の事前の把握と整理を済ませておくこと。
- 仕様調査は、顧客価値の重要度順および基幹的仕様から始めること。
- 重要な仕様・基幹仕様に関するQ & Aは、直接対面コミュニケーションで確認すること。
- 仕様追加・変更の影響度<sup>18</sup>の事前調査を済ませておくこと。
- 全ての問題点・疑問点を掘り起こすこと。
- 仕様内容および変更内容は、チーム内（含む評価チーム）で即時的な情報共有を行うこと。
- 仕様調査のQ & A情報や変更内容は、要求仕様書や設計書も同時に更新すること。
- 仕様決定事項や変更事項は、集中的かつ情報共有可能なシステムで管理すること。



### なぜそんなに大人しいの？

仕様決定の遅延で何度もその後の工程が崩れた経験をしているのに、被害を受けている開発者たちは、なぜか大人しいのです。愚痴を言いつつ、ただ頑張るだけで良いのでしょうか。仕様提示者にクレームをつける勇気も気力もないのでしょうか。論理的な仕事を遂行しているはずなのに、一向に合理的な行動を起こせないような状態に何故なってしまったのか、それをどうしたら良いのか、もう一度自分自身で真剣に考える必要があるでしょう。

計画

◎仕様に関する疑問点・不明点は、設計着手前に完全に解消しなければならない。

### 【仕様の残存疑問点・不明点】（図3-1-5）

残存疑問点・不明点の放置



後工程で不具合となって現れる

<sup>17</sup> **仕様調査** 要求仕様の疑問点や不明点を調査すること。この作業の質が、見積りの精度に大きく影響する。

<sup>18</sup> **影響度** 仕様変更の影響が間接的に及ぶ他のソフトウェア部分を特定すること。仕様変更影響度表により管理される。

## 要求仕様書の記述項目

要求仕様書の精度レベルを一定以上に保つためには、標準的な統一フォーマット<sup>19</sup>を使用し、記述必須項目および記述の粒度を指定した、要求仕様書の作成ガイドライン<sup>20</sup>が必要です。

要求仕様書の精度評価にあたっては、必須項目の記述の有無および記述内容のレベルをチェックし、見積りおよび設計を可能にするための条件をどの程度満たしているのかを評価する必要があります。

以下に要求仕様書の記述項目例を示します。

### 【要求仕様書の記述項目例】

1. システム企画 : 要求仕様の意図・目的・背景が記述されていること。
2. システム概要 : システム構成図、ソフトウェア構成、ハードウェア構成、コード体系<sup>21</sup>等の明示。
3. 要求仕様概要 : 全ての要求機能を漏らさず、機能ごとに端的に記述されていること。
4. システム運用フロー<sup>22</sup> : 顧客側におけるシステムの業務運用フローが記述されていること。
5. 要求仕様の詳細説明
  - ① 要求仕様詳細 : 開発項目ごとに要求仕様の詳細説明が論理的に記述されていること。
  - ② 入出力情報の定義 : 入力画面／出力画面・出力リスト等が論理的に示されていること。
  - ③ 内部データ処理の定義 : ビジネスロジック<sup>23</sup>のインプット・プロセスロジック・アウトプット条件等の明示。
  - ④ 外部通信の定義 : 送・受信データの定義、通信手順・通信フォーマットの定義等が明確であること。
6. 非機能<sup>24</sup>説明
  - ① データ要件の定義（件数の最大・最小値、保有期間等）が明確であること。
  - ② 性能要件の定義（レスポンス<sup>25</sup>・パフォーマンス<sup>26</sup>の規定）が明確であること。
7. 条件・制限事項 : 運用条件、制限、信頼性事項、拡張性事項、セキュリティ事項の明示。
8. 要求スケジュール : 開発開始・完了日、成果物出荷日、市場展開日程等の明示。
9. その他 : 既存システム構成、既存システムからのデータ移行<sup>27</sup>等の情報および条件の明示。

要求仕様書記述項目表のサンプルの一部は次の通りです。

<sup>19</sup> フォーマット 書式および構成があらかじめ設定されている文書のこと。

<sup>20</sup> ガイドライン あるものごとにおいて実行すべき指針を示したドキュメント。

<sup>21</sup> コード体系 JAN・EAN・UPC などの商品コード。

<sup>22</sup> 運用フロー 客先におけるシステムの業務運用の流れを示したドキュメント。

<sup>23</sup> ビジネスロジック アプリケーション機能における処理方法の論理のこと。

<sup>24</sup> 非機能 非機能要件とは、性能や信頼性・拡張性・セキュリティなど、機能要件以外のもの全般を指す。

<sup>25</sup> レスポンス ソフトウェアの応答速度。たとえば、あるキーを押してから表示が行われるまでにかかる時間など。

<sup>26</sup> パフォーマンス あるソフトウェアに起動をかけてから全ての処理が完了するまでにかかる時間。処理速度とも言う。

<sup>27</sup> データ移行 旧システムで使用していたデータを新システムに乗せ換えること。

【要求仕様書 記述項目】(表3-1-1)

			開発要件 必要項目	見積要件 必要項目	今回提出 の項目
表紙					
変更履歴					
1. システム企画・目的				*	
2. システム概要	2. 1 システム構成図	2. 1. 1 全体構成図		*	
		2. 1. 2 店舗機器構成図		*	
		2. 1. 3 ネットワーク図			
	2. 2 ハードウェア構成				
	2. 3 ソフトウェア構成				
	2. 4 コード体系				
3. 要求機能概要	3. 1			*	
(要求機能を漏らさず箇条書きにする)	3. 2			*	
	3. 3			*	
	3. 4			*	
	3. 5			*	
4. 運用フロー					
5. 要求機能説明	5. 1 要求機能詳細 (開発項目毎に要求仕様の詳しい説明をする。)	5. 1. 1			
		5. 1. 2			
		5. 1. 3			
		5. 1. 4			
		5. 1. 5			
	5. 2 入出力情報	5. 2. 1 入力画面			
		5. 2. 2 入力モニター			
		5. 2. 3 出力画面			
		5. 2. 4 出力リスト			
		5. 2. 5 レシート			
	5. 3 その他	5. 3. 1 集信データ			
		5. 3. 2 配信データ			
		5. 3. 3 オンラインフォーマット			
		5. 3. 4 メモリーバランス			
6. 非機能説明	6. 1 データ要件	6. 1. 1 データ件数		*	
		6. 1. 2 データ保有期間		*	
	6. 2 要求性能			*	
7. 条件・制限事項	7. 1 運用条件			*	
	7. 2 制限事項			*	
	7. 3 信頼性				
	7. 4 拡張性				
	7. 5 セキュリティ				
8. 概略スケジュール	8. 1 稼働スケジュール				
	8. 2 展開スケジュール				
9. その他	9. 1 従来システム構成				
	9. 2 従来システムからのデータ移行				

計画

◎ 過不足ない要求仕様書は、精度の高い見積りとスムーズな開発の基盤となる。



## 要求仕様書精度の簡易チェック法

要求仕様書の簡易的なチェック方法について説明します。

あいまいな要求仕様書であるか否かは、その要求仕様書を使って一部の重要な仕様部分について基本設計書が作成できるかどうかを検討してみることです。もし基本設計書の重要な部分が作成できないとしたら、それは要求仕様書に欠陥があるということになります。ほとんど基本設計書が書けないとすれば、その要求仕様書は全く使えないレベルのドキュメントだと言ってもかまわないでしょう。

また要求仕様書と対をなしている仕様書としては、総合テストにおいて使用される運用テスト仕様書があります。運用テスト仕様書は、開発されたソフトウェアが要求された通りに運用に沿って動作するかどうかをテストするためのドキュメントです。要求仕様書の一部分について、総合テストの担当者に運用テスト仕様書が書けるかどうかを試してみても要求仕様書の精度を確認できるでしょう。

## 要求仕様書の詳細チェックリスト

Check Timing : 要求仕様書発行・受領時

要求仕様書精度を詳細に渡ってチェックしたい場合には、下記のチェックリストを使用すると良いでしょう。

要求仕様書になじみの薄い初級プロマネの人においては、実際の要求仕様書を数点選び、下記のチェックを行って見ればイメージをつかむことができるかと思います。

### ◆① 妥当であること

- 顧客やユーザーのニーズと一致していること。
- 上位のシステム要求仕様書などの関連する他のドキュメントとの矛盾がないこと。
- 未確定項目がある場合は、どのように合意するか、依頼者と合意形成方法を決めておく。

### ◆② あいまいでないこと

- 要求仕様書に記述されている要求が、ただ一通りに解釈できること。
- 要求仕様書の“良し悪し”を判断する手段や基準をもつこと。
- 「範囲」を読み取れるように要求を表現すること。
- 仕様は「仕様である」ことを明示し、説明は「説明である」ことを明示して記述すること。
- 要求仕様書では、記述内容が“特定”できる表現になっているものを“仕様”とすること。
- 要求仕様書の構成や内容は、後工程の読者に分かるように書くこと。
- 「等」や「e t c」の文言は使用しない。使用する場合は、○月○日までに決めるとコメントをつける。

### ◆③ 完全であること

- 顧客やユーザーの情報システムに対するニーズ<sup>28</sup>が、漏れなく要求仕様書に記述されており、かつ図表の参照や用語の定義などの要求仕様書の形式が整っていること。
- 「境界」は早い段階で決めること。
- 「要求」のモレを防ぐために、カテゴリの分類や要求の分割・階層化に漏れがない、隙間がないことを確認できるようにすること。
- 要求仕様書には、「操作性」「保守性」「交換性」などの「品質要求」を記載すること。
- 階層化の基準として、以下を（状況によっては組み合わせて）使い、「隙間」なく分割すること。  
⇒ 時系列分割（時間軸分割）／構成分割／状態分割／共通分割など。
- モレなく書くこと。
- 要求仕様の番号をテストケースの番号とひもづけし、テストケースにモレがないことを確認すること。
- 仕様をグループに分け、さらに集合を小さくし、混じり気のない仕様のグループを作る。
- <グループ名>に要求の性質を持たせるためには、範囲を表したグループ名を選ぶこと。
- 「……は、……しない」という「否定表現」を避け、t h e nとe l s eの両方を明らかにすること。

### ◆④ 矛盾がないこと

- 要求仕様書内部で矛盾や衝突がないこと。
- ほかの機能の仕様と衝突していることに気づくためにも、仕様は早期に展開すること。
- 早い段階で全体の仕様化を行うこと。

### ◆⑤ 重要度と安定度のランク付けがされていること

- 各要求について、重要度と安定度<sup>29</sup>を示す指標を明確につけておくこと。
- 確認中の仕様をそのまま記述し、変わる可能性があることを明記すること。

### ◆⑥ 検証可能であること

- 開発されたソフトウェアが、要求仕様書に記述された要求を満たしているかどうか確認可能であること。
- 検査部門の人に、「検査可能」という側面から要求仕様書のレビューを実施してもらう。
- 品質要求（「操作性」「保守性」「交換性」など）はテストでも確認すること。

---

<sup>28</sup> ニーズ 要求ないしは要望のこと。顧客のニーズが漏れなく要求仕様書に記述されていることを確認する手段としては、顧客との仕様打ち合わせ議事録や顧客から提供される提案依頼書（R F P : Request For Proposal）および顧客への提案書などがある。

<sup>29</sup> 安定度 仕様変更の可能性の高さ低さのことを仕様の安定度と言う。

◆⑦ 変更が容易であること

- 要求仕様書に対する変更が、容易に、完全に、一貫して行えるようになっていること。
  - a) 目次や索引、明確な相互参照が整備され、使いやすい構造になっていること。
  - b) 冗長でない、つまり、同じ要求が要求仕様書内で複数個所に記述されていないこと。
  - c) 他の要求と混ざらず、各要求を独立・分離して表現して、互いに依存していないこと。
- 重複なく書くこと。
- 仕様書全体を「均一」に記述することにこだわらないこと。関係者間で共有できている認定仕様<sup>30</sup>まで、詳細に記載しなくてもよい。
- 仕様番号の確定作業は、仕様化の最初の段階では行わないこと。グループ分け確定後に行うこと。
- 似た記述が続く場合に、何が違うかをすぐに読み取れるようにすること。

◆⑧ 追跡可能であること

- 要求仕様書に記述された個々の要求に関し、その起源が明確であり、開発が進行するに伴って作成された文書等との対応付けがとれること。
  - a) 後方追跡可能性があること。
  - b) 前方追跡可能性があること。
- 設計や実装の工程で明らかになった「仕様」は、要求仕様書に書き戻すこと。
- 「要求」と「理由」をセットで表現すること。
- 要求仕様には固有の記番号を付けること。

(参考資料： I E E E 8 3 0 品質特性、U S D M)

計画

◎要求仕様書のチェックポイント

1. 妥当であること
2. あいまいでないこと
3. 完全であること
4. 矛盾がないこと
5. 重要度と安定度のランク付けがされていること
6. 検証可能であること
7. 変更が容易であること
8. 追跡可能であること

<sup>30</sup> 認定仕様 その顧客において過去に十分な稼働実績がある、いわゆる枯れた仕様のこと。

## 要求仕様書の精度検証

要求仕様書の精度が悪い場合、正確な見積りもできず、さらには設計すら不可能になってきます。このような事態を避けるために、要求仕様書を出す側および受け取る側の両方において要求仕様書の精度を常にチェックする必要があります。要求仕様書の精度低下の問題点を改善するために、要求仕様書を作成する側・受領する側双方において以下のような対策を実行する必要があります。要求仕様書を作成する側（顧客ないしは開発会社側の代行部門）においての実行が困難な場合は、開発部門においてだけでも実行する必要があります。

要求仕様書の精度確保のためには次の2項目の実行が有効です。

- ① 開発要件を過不足なく網羅した要求仕様書ひな形の作成および運用
- ② 要求仕様書の採点評価

要求仕様書の完成度評価のための採点表のサンプルを下記に示します。

【要求仕様書 採点表】(表3-1-2)

項番	分類	大項目	中項目	要求仕様内容	記述必須項目	記述の有無	記述内容得点	備考
				ユーザー名: XXXX	開発担当部署: XXXX			
				プロジェクト名: OOOO	採点者: XXXX			
				0: 記述なし				
				1: 仕様としては不十分				
				0: なし	* 記載が80点未満の場合、基本的には見積不可とする。			
				1: あり	* 記述内容得点が60点未満の場合、基本的には見積不可とする。			
				2: 多少不明点はあるが内容はわか				
				3: 要件として十分				
1	目的	1	1	表紙・変更履歴・目次	1	1	3	
2		2-1	システム化の理由	1	1	3		
3		2-2	システム化前後の運用	1	1	3		
4		2-3	システム化のメリット	1	1	3		
5	概要	3	3	システム構成図	1	1	3	
6		4	4	ハード構成				
7		5	5	ソフト構成	1	1	2	
8	運用	6	6	運用一覧				
9		7	7	運用スケジュール	1	1	3	
10		8	8	概要ビジネスフロー（運用フロー）				
11	概要	9	9	業務一覧	1	1	3	
12		10	10	ビジネスルール定義				
13		11	11	詳細ビジネスフロー				
14	機能	12	12	画面一覧				
15		13	13	画面設計		0		
16		14	14	画面項目説明				
17		15	15	帳票一覧				
18		16	16	帳票設計				
19		17	17	帳票項目説明				
20		18	18	機能定義書	1	X	X	
21		19	19	画面遷移図				
22		20	20	要求データ一覧				
23		21	21	要求データ仕様書	1	1	2	
24	22	22	オンライン一覧	1	1	2		
25	23	23	要求性能	1	1	2		
26	24	24	その他					
合計					12	11	29	

記載の有無	記述内容得点
満点ポイント	12
得点ポイント	11
得点 (100点満点換算)	92
記載率	29
記述内容満点	36
記述内容得点	81

記述内容満点 = 12×3 = 36点

記述内容得点 = (29/36)×100 = 81点

本表の拡大版は、「付録図表3. 要求仕様書 採点表」を参照のこと。

### 3-2. 要求仕様に関するリスク

要求仕様に関するリスク管理表の一例を次に示します。

【リスク管理表（要件定義工程）】（表3-2-1）

プロジェクト名：		初版発行日：		更新日：		文書番号：	
要因分類	QCD影響			要件定義工程リスク	✓	チェック	
	Q	C	D			検出日	
ヒトに関するリスク	①他者依存的姿勢（自律性の放棄）	○	○	○	・顧客の動き待ちの姿勢の結果、納期遅延		
	②上位マネジメントの関与不足		◎		・顧客交渉戦略・能力不足 ・大幅な仕様追加要求の丸呑み ・大幅な開発費削減要求の丸呑み ・大幅な納期短縮要求の丸呑み		
	③ユーザーの参加・協力度不足	◎	○	○	・顧客の参加・協力度が低い		
	④組織能力不足（未熟な組織文化、戦略の欠如）	◎	◎	◎	・見積りプロセスの手続きルール違反		
	⑤見積り能力	◎	◎	◎	・既存資産流用の可否判断 ・顧客要件の精査 ・フィット&ギャップ調査能力		
	⑥要件定義能力	◎	○	○	・要件定義能力不足		
	⑦リーダーのプロジェクトマネジメント能力（外部交渉、タイムマネジメント、現場主義、見える化能力など）	◎	◎	◎	・顧客交渉戦略・能力不足 ・大幅な仕様追加要求 ・大幅な開発費削減要求 ・大幅な納期短縮要求 ・適時の支援依頼発信能力不足 ・変更管理未実施による開発費の増加 ・顧客に対する不都合な事実の隠蔽（故意による事故）		
					・要件定義体制不備（業務経験・知識不足の要員で要件定義を実施）		
					・要件定義遅延のため不十分なテストで客先リリース ・顧客要求の優先順位コントロール失敗による開発失敗		
					・タイムリミット管理能力不足		
	モノ	⑧メンバーの技術能力、ヒューマンエラー（うっかりミス）	◎	○	○	・新技術知識不足 ・顧客業務知識不足	
⑨プロセス管理の有無		◎	○	○	・プロセス管理の不足 ・仕様凍結未確認のまま設計着手 ・基本設計の承認なしで詳細設計着手		
⑩コミュニケーション能力（阻害・ギャップ）		◎	○	○	・顧客のパートナー化失敗 ・顧客とのコミュニケーション能力 ・プロマネと上司間のコミュニケーション ・営業と開発部門間のコミュニケーション能力 ・営業部との交渉力		
カネ	⑪関連部署との連携不足	◎	○	○	・組織間の協調性・コミュニケーション		
	⑫ドキュメントの不備（要件定義書・設計書・チェックリスト・手順書など）	◎	○	○	・要求仕様書なし		
	⑬開発のベースの有無	◎	◎	◎			
情報	⑭開発環境の不備	◎	○	○			
	⑮開発費不足	◎	◎	◎			
情報	⑯資源投入戦略の誤り（開発費投入時期ミス）	◎	○	◎			
	⑰あいまいな開発範囲（スコープ）	◎	◎	◎	・あいまいなスコープ		
	⑱あいまいな要求仕様	◎	◎	◎	・あいまいな要件 ・あいまいな仕様		
	⑲情報の不備・不足（マネジメント情報、技術情報、過去の失敗情報）	◎	○	○	・情報の不備・不足		

本表の全体図は、「付録図表4. 要件定義工程のリスク管理表」を参照のこと。

### 3-3. 顧客要求の重要度順位の設定

基本的な仕様の確定およびWBSの作成によって開発すべき顧客要求事項の内容はすべて明確に定義されました。続いて重要な作業として、プロジェクトの限られた時間と開発費を最大限有効に使うために、顧客要求事項の開発の優先順位を決めておく必要があります。

要求事項はすべて顧客の要求に基づいていますが、すべてが同じ顧客価値をもっているわけではありません。どの仕様がより顧客にとって重要かということは、要求仕様を決定する過程における顧客とのコミュニケーションの中で判断することができます。重要度の設定作業は、まず複数の要求事項を顧客価値の重要度の順に、例えばA/B/Cの三つのグループに分けることから始めます。つまり開発の優先順位はA>B>Cのグループの順となります。続いて各グループ内における優先順位の決め方として、Aグループに属する仕様として例えばa、b、cがあった場合、aを先に開発しなければbもcも動作できない場合は当然のことにaから着手することになります。このaをロードブロック<sup>31</sup>仕様（機能）と呼ぶことにします。同一の顧客価値内での優先順位はロードブロック仕様が優先権をもつこととなります。ロードブロックの順に並べたものがプロセスの順になって来ます。このように優先順位は、価値の順とプロセスの順という二つの意味合いを同時に判断する必要があります。

#### ◎ 優先順位の判断基準は次の二つ

1. 顧客価値の重要度順
2. ロードブロック仕様の順

計画

開発プロジェクトに起こりがちな優先順位無視の行動として以下のものがあり、プロマネは開発の全工程においてこれらの問題が発生しないように対策を行う必要があります。

#### 【優先順位を無視した不正な開発行動】

- ・ 仕様未決定状態における事前開発着手
- ・ 要件定義書未完での基本設計の着手
- ・ 基本設計未完での詳細設計の着手
- ・ 詳細設計未完でのコーディング着手
- ・ コーディング未完での評価テスト着手
- ・ 総合評価未完でのソフトウェアのリリース<sup>32</sup>

<sup>31</sup> **ロードブロック** 元の意味は道を塞いでいる障害物のこと。先に進むためにはその障害物を片付ける必要があると言うような文脈で使用される言葉。

<sup>32</sup> **リリース** 完成したソフトウェアおよび関連成果物を顧客へ引き渡すこと。

これらの優先順位を無視した行動の主な原因は、開発時間および予算の不足にあります。時間が不足した結果、あせって事前着手に走り、その結果多くの手戻り作業を発生させ、また複数の仕事を同時に処理できず不十分な作業内容による全体的な品質低下を招くことになります。

妥当な優先順位を決定するためには、その前に見積り交渉などにおいて妥当な開発期間および開発費の獲得が必須条件となります。

次に開発作業におけるものごとの優先順位の決め方を下記に示します。

#### 【開発作業における優先順位の決め方】

- ・ 仕事の意味・意図・背景を仕事の着手前に把握すること。
- ・ 仕事内容をブレークダウンすること。
- ・ ブレークダウンした仕事内容に優先順位を設定すること。
- ・ タイムリミットが直近に迫っているものから処理をすること（緊急度優先）。
- ・ 時間的余裕があるものは重要機能の順に処理すること。
- ・ 同等の優先順位のもの、効果が高く実行が容易なもの順で、効果が低く難易なものは最後に。
- \* 優先順位が判断できない場合は早めに顧客や上長に相談すること。
- \* この仕事に許される自分の残り時間を毎日意識すること。

作業の優先順位の決定にあたっては、次の三つの視点での検討が必要になります。

#### 【優先順位決定に必要な3つの視点】

1. 一つの作業内における処理の順番の妥当性の視点
2. 複数の作業の緊急度・重要度の優先順位判断の視点
3. 手持ち時間と必要時間の比較判断の視点

この三つの視点における判断ミスおよび必要時間の確保の失敗は、Q C Dを大きく毀損する結果を招くことになります。

また、優先順位の誤った認識としては次のようなものがあります。

#### 【開発作業における優先順位を誤らせる原因】

- ・ 割り込み作業等の優先度判断の誤り
- ・ 必須業務に優先順位はないという認識の欠如
- ・ 納期第一優先の誤り（Q C Dはみな等しい優先価値をもっている）

計画

#### 【優先順位の原則】

顧客の要求は機能単位に分割され、顧客価値の優先順位の高い順に開発される。

### 3-4. 見積りにおけるリスク

見積りは、要件定義およびチームの技術能力とやらんでプロジェクトを成功に導く三大要因の一つです。逆の表現をすると、見積り・要件定義・チームの技術能力はプロジェクトを失敗させる三大リスクとなっています。

見積り金額と開発期間（着手可能時期、開発完了時期）は、先に述べたWBSの最小レベルの部材に要する金額と期間の総和として算出されることとなりますが、見積り回答書を作成するにあたってはさまざまなリスクを考慮する必要があります。

#### ソフトウェアの価格

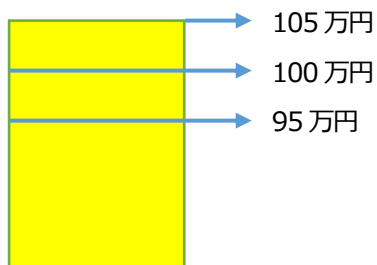
ソフトウェアの価格、特にカスタムソフト<sup>33</sup>の派生開発においては定価がないのが実状です。なぜなら顧客の要望に従って作成され、一本毎に内容が異なるため、いわゆる一品料理となり、全く同じものがこの世の中には存在しません。そのために価格についていくらが正当な価格なのか判断し難いのが実状です。

例えば100万円程度の開発物件の場合を考えて見ると、この見積りにおいて105万円の場合、100万円の場合、と95万円の場合の違いを明白に区別できるでしょうか。105万円と95万円では実に10ポイントもの差がでてしまいます。優秀なプロマネならば発注者側の状況と自社側の状況を深く勘案しつつ、適正な価格の落とし所を見極めなければいけません。元値の100万円という見積り内容にちゃんとした根拠があり、相手に納得させる能力さえあれば利益率の5%向上はそれほど難しいことはありません。

計画

◎有能なプロマネは+5%の見積り額を獲得する。

【見積り価格】（図3-4-1）



100万円程度の物件で、105/100/95万円の違いを明確に説明することは難しい。1千万円・1億円でも同様。

<sup>33</sup> カスタムソフト 顧客からの特別な要求仕様に基づいて開発されるソフトウェア。対語は市販品であるパッケージソフトウェア。



## 見積り回答書に対する認識

最初に見積り回答という行為に対する認識をしっかりと持つ必要があります。見積り回答書とは、法的な拘束力をもつ契約書です。受注者は発注者と約束したものを完成させる義務があり、発注者は受注者と約束した対価を支払う義務があります。この相互義務の履行を法的に約束した書類が見積り回答書です。見積り回答書は、単なるドキュメントとは訳が違うという認識が必要です。

計画

◎見積り回答書は、契約書である。

## 受注者にとっての見積り回答書の重要性

受注者の大きな目的の一つは、その仕事によって利益を上げることです。見積り回答書においては、達成すべき仕事内容、その対価および実行期間について約束してしまいますので、見積りミスや仕事の失敗は直ちに赤字を招き、企業の存続を危うくする場合があります。

さらに見積り回答書は開発チームの仕事の原点であり、その後の開発における人・モノ・カネ・時間を規定するものであり、すべての書類やドキュメントの中でも最重要な書類です。見積り回答書は、その会社や組織のすべての実力を映し出す鏡といっても過言ではありません。

見積り回答書の基本的な要件は以下の二つだけです。

- ① 分かっている内容についてのみ見積ること
- ② 分かっていない内容については、見積りに含んでいないことを明記すること

すなわち見積り回答は、開発の対象となる機能について事前に過不足なく定義された要求書に基づいて行い、想像や想定に関することがらを一切含めてはいけません。

計画

◎見積りは分かっているものだけを見積ること。

## 見積り回答書の品質



### 見積り回答書の形式的なチェック

妥当な見積り回答を行うためには、最初に見積りにおける記述モレ・考慮モレなどの単純なミスを防ぐために、見積り回答書の形式的なチェックが必要になります。モレなどのミスにつながりやすい項目としては、次のようなものがあります。

#### ◆【見積り回答書の形式的なチェックリスト】

- 見積り回答期限の事前確認は行ったか。
- 見積り対象の仕様は明確になっているか。
- 見積りにインプット条件は全て網羅したか。
- 見積りにアウトプット条件は全て網羅したか。
- 仕様の疑問点・不明点は全て顧客に確認したか。
- 仕様変更が及ぼす影響範囲は特定したか。
- 特別な見積り条件の要求があった場合、その考慮は行ったか。
- 見積り範囲やリスクに関する条件を見積り回答書に記述したか。
- 見積り回答書に記述した開発範囲以外のものは別途見積りとするという文章を記述したか。
- 開発着手時期および完了時期を記入したか。
- 見積り有効期限を記入したか。
- \* その他、過去の自他における見積り失敗項目を記述すること。

上記のチェックリストに、みなさんのプロジェクトで起こりがちなミスを加えれば、更に良いチェックリストになるでしょう。またこれらのミスを出さないような、見積り回答書の統一的なフォーマットやガイドラインの整備も必要となります。

計画

◎見積りガイドライン・見積りチェックリストによる単純ミスの防止を。



## 見積り精度の向上

次に必要なことは、見積りの内容自体の精度向上です。見積りの精度が低いために、妥当な開発費に対して異常に高い回答をすれば、顧客側の怒りを買い信用を失います。反対に誤って低い金額で回答してしまった場合、プロジェクトを成功裏に完了させることは不可能になってしまいます。開発期間についても同様のことが言えます。

一定の見積り精度を確保するための要点を次のチェックリストにまとめました。

### ◆【見積り精度向上のチェックリスト】

- 見積り対象の仕様知識に習熟していること。
- 見積り対象システムのプログラム構造を理解していること。
- 見積り前に要求仕様の事前調査を済ませていること。
- 既存の設計書等が不備な場合、影響範囲に絞ったソースコードの調査を行うこと。
- 事前調査の結果をドキュメントに残しておくこと。
- 見積り対象仕様における、過去の開発の失敗事例を把握しておくこと。
- 見積り対象仕様の開発に必要な技術を保有していること。
- 過去の類似開発の見積り／実績データを参考にした見積りを行うこと。
- 見積り範囲（開発システムのスコープ）を明確にすること。
- 見積条件を明示すること。
- ソフト性能<sup>34</sup>**設計の根拠となる、適用H／W等の性能値条件を明示すること。
- 要求仕様の精度レベルによって見積りリスク係数を設定すること（非公開とする）。
- 流用部分はブロック図で表現すること（顧客に説明しやすい）。
- 添付資料の充実を図ること（運用フロー・データフロー等）。
- 納期によるステップ分けは別資料とすること。

大方の見積り方法は、経験に基づく手法で行われているようですが、単なる経験的な直感での見積りでは、失敗の危険性が非常に大きくなります。経験に基づくと言っても、何らかの経験知、すなわちデータに基づく必要があります。見積りに関する経験知データとしては、過去のプロジェクトにおける見積り値と実績値のデータがあります。毎回、見積りと実績の差異およびその原因・理由についての記録を取り、その改善対策を行うことで、チームの能力は強化され、次なる見積りの精度ははるかに高いものになります。全てのプロジェクトにおいて、見積りおよび実績値の記録および差異比較の振り返りは必須です。

---

<sup>34</sup> **ソフト性能** ソフトウェアの動作速度、特に応答速度（レスポンス）および業務処理速度（パフォーマンス）は重要なソフトウェアの性能要件とされる。

正確な見積りをするためには、まず仕様理解力と設計能力が必要です。見積りは基本的に、何をどう作るかが分かっているだけで済みます。つまり明確な要件定義とそれを実現する設計が正しくできれば、見積りは可能です。要求仕様が明確なのに、正確な見積りができないということは、自分の仕様理解力と設計能力が未熟だということになります。見積り手法はあくまでも道具にすぎず、その手法があれば見積りができるというものではありません。ソロバンがあっても使えなければ意味がないのと同じです。まず自分の設計技術スキル向上と経験の積み重ねが必要です。

見積りの精度向上のためには上記の要件を満たす必要があります。当事者において能力不足の部分については他者の支援を仰ぐ必要があります。

計画

◎ 正確な見積りには、仕様理解力と設計能力が必要。

### 見積り方式

見積り手法の主なものには、①経験的手法、②LOC (Line of Code)<sup>35</sup>法、③FP (ファンクション・ポイント)<sup>36</sup>法などがあります。

①の経験的手法は広く行われている方法ですが、見積りの精度を上げるためには、過去のプロジェクトにおける、見積り対実績の差およびその原因についての分析データの蓄積が必要です。

②のLOC法は、 $\text{工数} = \text{ステップ数} \div \text{開発生産性} \times (1 + \text{間接費要員比率}) \times \text{余裕率}$ で求められます。

③のFP法は、 $\text{工数} = \text{基準値} \times (0.65 + \text{調整値} \div 100)$ で求められます。

②③の手法は、その開発組織としての今までの実績のデータの積み重ねがなければ、計算式のパラメータ値を決めることができません。見積りの精度は見積り方式に依存するというよりも、見積り者が見積り対象の仕様ないしは機能に精通している度合いに依存しています。すなわち対象仕様・機能に関する開発経験および知識に決定的に依存しています。

計画

◎ 見積りの精度は見積り方式に依存するよりも、仕様知識と開発能力に強く依存する。

<sup>35</sup> LOC (Line of Code) ソフトウェアの規模を表す指標のひとつで、ソースコードの行数を意味する。

<sup>36</sup> FP (ファンクション・ポイント) ソフトウェアの機能数および複雑度による重みづけをした点数の合計から開発工数を見積る方式。

## 見積りにおけるリスク

見積りに関する主なリスクは次の通りです。

### 【見積りリスク】

- ① 要件定義書内容の不備（必要事項の記述なし・記述ミス、不明点・疑問点が多いなど）
- ② 要件定義書なしの見積り依頼（口頭・電話のみによる依頼など）
- ③ いつまでも決まらず二転三転する要求仕様
- ④ 短時間・短期間での見積り回答要求
- ⑤ 口頭や概算ベースの回答が正式回答として扱われる
- ⑥ 最初から仕切り値で開発費や納期を要求してくる場合
- ⑦ 特定の依頼者による、頻繁な受注につながらない見積り依頼
- ⑧ 見積り者における仕様の理解不足や事前調査の不足
- ⑨ 見積り者における交渉力の弱さ

見積り回答におけるリスク回避には次のような対策が必要になります。

### 【見積りリスクの回避策】

- ① 見積り可能な要求仕様の提示がない見積り依頼は、出し直しを要求する。
- ② 妥当な見積り期間が設けられていない見積り要求には、必要期間を要求する。
- ③ 過剰な開発費削減・納期短縮要求に対しては、合理的根拠の提示を要求し安易な妥協はしない。
- ④ 口頭での見積り依頼に対しては口頭で返し、口頭レベルの依頼・回答は一切正式なものとしては扱わず、責任も持たない旨を通告しておく。
- ⑤ 公式のルートおよび承認のない見積り依頼は、出し直しを要求する。
- ⑥ 受注ヒット率が低い見積り依頼を頻繁に出す部署ないしは個人へ事情聴取を行う。
- ⑦ 短納期リスク物件においては、分割見積り・分割開発・分割リリース等の交渉を行う。
- ⑧ 複数社開発体制プロジェクトにおいては、自社責任範囲を見積り回答書に明記する。
- ⑨ 不慣れな仕様に関しては経験者の知見を借りる。

計画

◎ **リスクーな見積り依頼者が最大の見積りリスクである。**  
**誰からの見積り依頼なのかに気をつけること。**



### 見積りリスク回避対策の実行に当たって

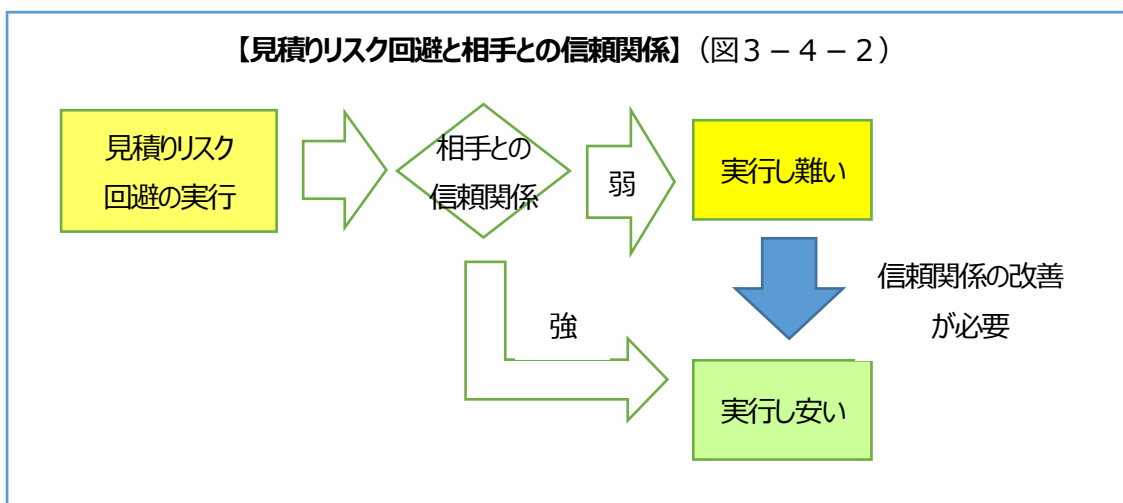
前記のリスク回避対策は、すぐに全てを実行することは無理なことでしょうが、これらのことは本来当たり前のことを当たり前に行うようとしているだけのことです。

これらのリスク回避を行わなければ、まともなQCDの達成はとてできません。一つひとつの回避策の実行を重ねていくことが期待されます。良い製品を提供し、QCDを成立させるために必要ならば、一見無理だとか強硬に見えることに取り組まなければ、現状を打開することはできません。これらの施策を通すためには、まず良い品質の成果物を先に提供し続ける実績が必要で、良い製品を妥当な納期で出荷できた実績を積みめば、見積り依頼部門も開発部門の要求に耳を傾けてくれるでしょう。

これらのリスク回避対策は、実際に開発組織において実行されたもので、架空の話ではないことを付け加えておきます。

そうは言っても、下請け開発会社が元請け開発会社に対して同様のリスク回避対策を行った場合、元請け担当者の怒りを買って、仕事を失うのではないのかという懸念の提起がありますが、元請けとの信頼関係が貧弱な場合はきっと怒りを買うことでしょう。そんな要求をする前に自分たちの開発品質をまともにしてください、と言われかねません。まずは自分たちの開発品質を合格レベルに向上させながら、これらの見積りリスク回避対策を順次実行していくことが順当なやり方でしょう。

【見積りリスク回避と相手との信頼関係】 (図3-4-2)



## 概算見積り

正式見積りの前段階として、詳細仕様が未決定の状態、概算見積りの要求を受けますが、概算見積りは一見簡単なようですが、実はそう簡単なものではありません。概算＝いいかげん、では困ります。自分が精通している領域でなければ、概算見積りも詳細見積りもできないと言ってもよいでしょう。

調査をどの段階で見切るかという判断は難しいものだと思います。自分における判断は、その見積り対象に対して、自分が保有する経験・知識および自分が利用できる他人の経験・知識の質と量によって決定されます。自分が利用できる経験・知識の質と量が大きい程、それによる判断は妥当性のあるものになります。

ある程度分かっているものならば、概算的見積りも可能でしょうが、利用できる経験・知識の総量（質）が少なければ、大雑把な見積りも困難になります。

細かく分析して工数や時間を見積もる詳細見積りは、いわゆる「定量的」見積りですが、大雑把に見当をつけるだけで良いのなら下記方法を試してみてください。

### 【概算見積りの方法】

- ① 要求仕様の全体像を、その目的・意味・背景を含めて把握すること。
- ② 自分で見積り可能なものとそうでないものを分ける。
- ③ 未経験項目および重要リスク項目をリストアップする。
- ④ それぞれに対して、必要工数を大雑把に3段階（大・中・小）または5段階に分けて記入する。  
各段階に要する工数は、例えば、大＝1ヶ月、中＝2週間、小＝1週間等に決めておく。  
重要リスクについても、もしそれが起きた場合についても同様に工数の大・小を記入しておく。  
自分で決められないものについては経験者や上長の意見を聞く必要があります。
- ⑤ これらの仕事を何人で実行するかを想定しておく。
- ⑥ 非常に大雑把ですが、上記④の開発期間の合計を⑤の人数で割ったものが想定されるスケジュール期間になります。金額についても同様の方法で算出します。上記のやり方を、仕様検討工程・設計工程・製造工程・評価工程に分けて算出すれば、ある程度妥当な数字が出てくるでしょう。

そうは言っても、要求仕様の骨子も決まっていないような物件については、概算見積りすら不可能であり、このような場合には、要求者に対して早く要求仕様の骨子を決めるように促す必要があります。

計画

◎ 概算見積りはあくまでも参考値。  
正式稟議承認には正式見積りを、の注意書きを忘れずに。

## 見積り回答書

一般的な見積り回答書に記載が必要な項目は次に示した通りです。

1. 発行年月日
2. 宛先、宛先（写し）
3. 見積り回答書：概算／正式 \* 概算または正式を明示する。
4. 承認印欄：最終承認印欄・担当部長印欄・担当課長印欄
5. 見積り依頼書番号欄、見積り回答書番号欄
6. 開発名記述欄、ベース開発名記述欄
7. 開発内容記述欄
  - \* 開発内容および範囲について漏れなく明示すること。
  - \* 見積りの根拠となる説明資料を添付すること。
8. 開発条件記述欄
  - \* ソフトウェア性能設計の根拠となる、適用ハードウェア等の性能条件を明示すること。
  - \* 要求内容不明等で見積りから除外した項目を明記しておくこと。
  - \* 開発内容記述欄に記述のないものは、別途見積りにする旨を明記すること。
9. 開発費記述欄
  - \* 回答書表紙には開発費合計を記述し、主要項目ごとの明細開発費は添付資料とする。
10. 開発スケジュール記述欄
  - リードタイム<sup>37</sup> アクチュアル \* リードタイムまたはアクチュアルスケジュールを明示する。
  - \* アクチュアルスケジュールの回答は、着手可能年月日および開発完了年月日を明記する。
  - \* リードタイムスケジュールの回答は、〇〇ヶ月の表記を行うこと。
11. 見積り担当者記述欄：所属名・担当者氏名
12. 見積り有効期限年月日
13. 添付資料 有：x x 枚 無

<sup>37</sup> **リードタイム** 開発に要する期間を示したもので、開発開始・終了等の時期は明示されていない。



### 3-5. プロジェクト体制の構築

プロジェクトに必要なリソースの中で最も重要なものは人間そのものです。物・資金・情報などさまざまなリソースを使うのも結局人間であり、人間の出来不出来によっては、これらのリソースが十分にあったとしてもそれを生かし切ることはできません。結局プロジェクトの成否の鍵はどのような人材でプロジェクトの体制を構築するのかにかかっています。

#### プロジェクト体制構築の要件

プロマネは社内全体の他のプロジェクトの状況を常に把握しておくことで、新規プロジェクトに投入すべき人材の確保を確かなものしておく必要があります。未経験者ばかりしかないプロジェクト体制などは、最初から失敗が約束されているようなものです。プロマネは、開発の規模や難易度を把握し、それに見合ったスキルの人材を用意しておく必要があります。多くの場合、自分のプロジェクトに必要な人材を適時に集めることは難しく、そのためには社内の組織を初めとして外注組織においてどのような人材がいて、適材者がいつプロジェクトに参加可能なのか、という情報を普段から把握しておく必要があります。

プロジェクト体制構築にあたって必要な要件は次のようになります。

#### ◆【プロジェクト体制構築の要件】

- 名ばかり管理職をプロジェクトの責任者につけないこと。
- 上級者から初級者に至るまで、適切なスキルの人材で、指揮系統が明確なピラミッド構成を築くこと。
- 社内で人材の確保が難しい場合は、協力会社なども含めて、広く人材の確保に努めること。
- メンバーの役割を明確にし、それに適合する人材を割り当てること。
- メンバーは、開発要件に必要な技術的能力およびチームプレーに必要なコミュニケーション能力（報告・連絡・相談）などを備えていること。
- メンバーのスキルに関しては、組織全体で個人スキル管理カードなどにより常時管理を行うこと。

上記の要件で結成された開発チームは、数次にわたるプロジェクト活動および改善活動を通して、下記のような優れた能力を持つ開発チームに成長していくことが期待されます。

#### ◆【優れた開発チームの能力特徴】

- 自律的思考<sup>38</sup>・行動に基づいた自己決定力
- 意欲ある人材でプロジェクトを構成するプロジェクト構築力
- 顧客との日々の直接コミュニケーションによる協調力
- チームにおける日常的で密接な直接コミュニケーション力
- 顧客からの変更要求に対する俊敏な対応力
- 顧客価値優先度の把握力
- ドキュメントの常備はもとより、ちゃんと動作するソフトウェアの素早い開発力
- 一定のペースを持続できる開発力
- 無駄・不要な仕事の削減力および業務効率化能力
- 定期的・効果的振り返りによる、学習能力および行動の変更・調節力

(参考資料：アジャイルソフトウェア開発宣言<sup>39</sup>)

計画

◎ 確実なチーム体制構築は、プロマネにおける人材能力の見極め力と人材獲得力にある。

#### 開発体制不備による失敗リスク

開発体制の不備はプロジェクトの失敗に直結してしまいます。開発体制不備に関するリスクには以下のようなものがあります。

#### ◆【開発体制不備のリスク】

- 開発要件を満たすプロマネ・開発者の不足
- 開発者のトレーニング不足
- 未経験分野・新規業界参入の場合の準備不足
- 新技術採用場合の準備不足

これらのリスクは開発実行の前の事前準備段階において解消される必要があります。要件定義工程や設計工程に入ってから、必要な人材が足りないと騒いでも手遅れなのかも知れません。

計画

◎ 必要な人材の獲得は事前準備段階で済ませておくこと。

<sup>38</sup> 自律的思考 自分自身でものごとを直接観察し、その意味するところを判断すること。

<sup>39</sup> アジャイルソフトウェア開発宣言 アジャイルソフトウェア開発宣言は、アジャイルソフトウェア開発とその諸原則を公式に定義した文書である。この諸原則はウォーターフォール開発にも有意義なヒントを与えている。

## 外注開発体制の構築

分業体制が一般的になっている状況下において、元請けベンダー側にとっては、どれだけ有能な外注会社を抱えているのかがプロジェクト成功のファクターの一つになっています。とは言っても外注会社が有能であるためには元請ベンダー側は、さらにそれよりも有能な組織である必要があります。元請がプロジェクトマネジメントもよく分からない、要件定義も設計もよく分からないようでは下請会社もついて来なくなるでしょう。

外注会社における開発力の成長・維持のためには、定期的な外注会社の評価やQ C D改善のための提案書の提出などが必要になります。

外注会社に対する評価項目は、技術力・品質・コスト・納期の実績等に基づいた下記の項目などが必要でしょう。

### 【外注会社評価項目】

- ① 保有技術およびそのレベルの実績推移
- ② 発注品の難易度評価
- ③ 品質の実績推移（不具合発生件数・率・深刻度）
- ④ 品質管理体制の評価
- ⑤ 技術者のランク別の単価推移
- ⑥ 取引金額の実績推移
- ⑦ 納期・生産性の実績推移
- ⑧ 協力度評価

計画

◎有能な外注は有能な元請プロマネの下で育つ。

## 統合的なプロジェクト体制の構築

従来の開発体制は、ベンダー側にてプロジェクトの統合管理、要件定義およびシステムの基幹部の開発および評価を担い、不足要員を国内下請け会社で補う形が一般的でしたが、現在は外形的にはベンダー側における開発や評価業務は下請け会社にシフトされており、特に製造工程はコストメリット追求のために海外**オフショア**<sup>40</sup>会社に発注されていることが多いようです。

これらの分業は、開発の工程単位で行われることが一般的で、この方式を**工程別分業方式**<sup>41</sup>と呼びます。この工程別分業方式が持っている致命的なリスクとしてプロジェクト統合管理の欠落があります。



### 統合管理および**統合コミュニケーション**<sup>42</sup>の欠落

2・3人のチームで要件定義からリリースまでの全工程を担う小さなプロジェクトでは、コミュニケーションや情報の断絶などは起こりませんでしたが、開発規模の増大により各工程が別々の担当者で行われはじめると断絶が起こり始めます。さらに一つの開発プロジェクトの各工程を異なった複数の会社で担当した場合は、断絶は深刻な状態にまで進行し、失敗のリスクが非常に高くなってしまいます。このことは皆さんすでに経験済みのことでしょう。

それにもかかわらず各社各担当で分断されたコミュニケーションや情報の溝を埋めるための統合的なプロジェクトマネジメントはなかなか実行されません。

ベンダー側の言い分は、あの工程はA社に発注したのだから、責任をもって仕上げるのが当然などと主張していますが、その一方で要求仕様の早期凍結という自社の義務は果たされていません。これでは下請け会社は、自分の責任を全うすることは不可能になってしまいます。複数社で構成された開発プロジェクトが成功するためには、より上位にある会社が全体を取り仕切る統合マネジメントを実行する必要があります。この重要な義務を果たさない限りプロジェクトが成功することはないでしょう。

月に一度程度の管理職どうしで行われる形式的なパートナー会議では、本当の意味でのプロジェクト内の情報共有を図ることはできません。少なくとも各社のプロマネや開発リーダーを集めた情報共有および問題解消のための会議を週一回程度は実行する必要があります。元請け会社は各社間の調整および全体的なQCDコントロールの指揮をとる必要があります。

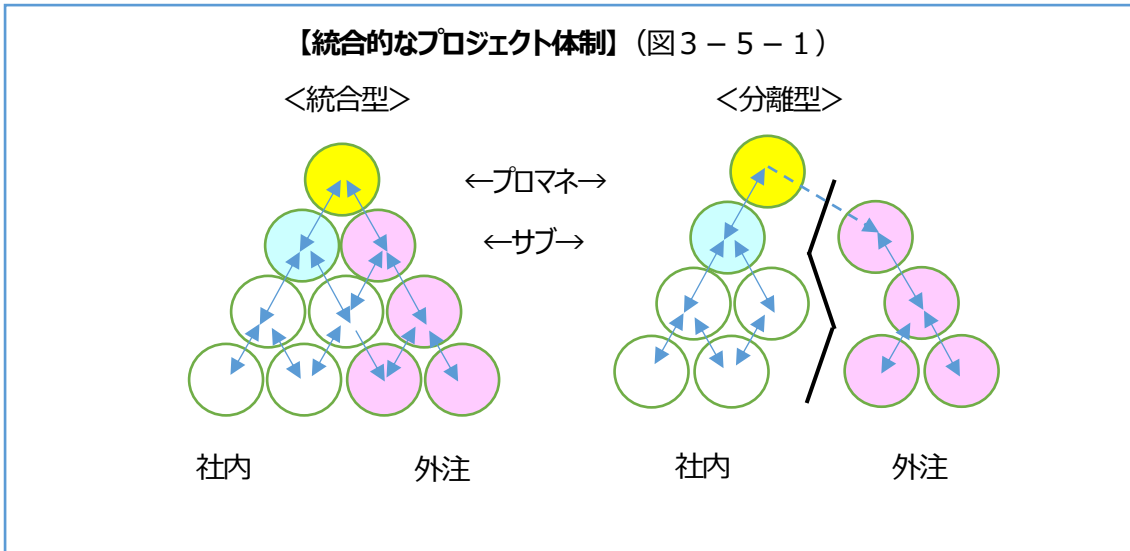
計画

◎プロジェクトの統合マネジメントは、  
コミュニケーションの分断を防ぎ、プロジェクトを成功に導く。

<sup>40</sup> **オフショア** (Offshore) ソフトウェア開発に関する業務を海外の会社に委託する形態のこと。

<sup>41</sup> **工程別分業方式** ソフトウェア開発の各工程を別々の会社で分業して担当している方式。

<sup>42</sup> **統合コミュニケーション** 開発工程ごとに担当会社や担当者が異なる場合に発生するコミュニケーションの溝を埋めるための計画的な情報共有会議の実施等。



### 3-6. 開発環境の手配

リソースの用意として忘れられがちなものが開発機器や評価機器の準備です。特に**ファームウェア**<sup>43</sup>開発における**ターゲットマシン**<sup>44</sup>と呼ばれる開発ソフトウェアが搭載される機器の手配には注意を払っておく必要があります。ターゲットマシンはハードウェア開発部署ないしは製造部門から入手する必要があり、通常そのリードタイムには長期間を要する場合も珍しくありません。特に依頼先が他社の場合は小まめなフォローが必須です。

手配の約束や依頼は行ったが、その後のフォローを忘れていたために、マシンの入手時期が数週間も遅れてしまう場合もあり、スケジュールに大きなインパクトを与えることになります。

また近年においては厳しいセキュリティが求められる社会情勢下であり、開発中においては顧客の重要なデータを取り扱うことも多く、これらの取り扱いに関しては社内ルールのもとに厳重な取り扱いを行う必要があります。特にクレジット・電子マネー・ポイントなどのデータの取り扱いに関しては、顧客の承認を得た専用のセキュリティルームの用意が必要であり、本番環境での開発およびテストで使用されるこれらのデータは、顧客所有の現金と同じであるという認識を開発者全員で持つ必要があります。

これらのデータの流出問題は、賠償問題に直結し、会社の経営を揺るがした事例も少なからずあります。

<sup>43</sup> **ファームウェア** ファームウェア(firmware)とは、電子機器に組み込まれたコンピュータシステム（ハードウェア）を制御するためのソフトウェアで、ソフトウェアをROM等の集積回路にあらかじめ書き込まれた状態で、機器に組み込んだもの。

<sup>44</sup> **ターゲットマシン** ファームウェアを搭載するハードウェア機器。

## 第4章 実行工程のリスク

### 4-1. リスクの解消

#### リスクと課題の違い

リスクと課題<sup>45</sup>の違いについて質問を受けることがありますが、この二つについて改めて定義をしておきます。リスクとは、現時点では問題として現れてはいないが、今後問題として現れる可能性のあるものです。一方、課題とは、現時点ですでに問題化してしまっている問題を指しています。

リスク管理表<sup>46</sup>と課題管理表<sup>47</sup>の二種類の管理表が存在する理由はこの違いによるものです。問題になる可能性のあるリスク項目と既に問題化している課題項目を一つの管理表で管理することは不適切であり、まだ問題化していないからリスク対応はとりあえず良いなどという勝手な判断の原因になってしまうこともあります。

◎ リスクとは、現時点では問題として現れてないが、今後問題化の可能性のあるもの。  
課題とは、現時点ですでに問題化してしまっているもの。

実行

【リスクと課題】 (図4-1-1)

課題	すでに問題化しているものごと
リスク	まだ問題化していないものごと

<sup>45</sup> 課題 現時点ですでに問題化してしまっているものごと。

<sup>46</sup> リスク管理表 そのプロジェクト特有の想定リスクおよび開発組織で繰り返し起こしている障害やミスなどの過去の失敗を中心としたリスクを記載した管理表で、開発の工程ごとのリスク管理表が必要とされる。

<sup>47</sup> 課題管理表 すでに表面化している問題をその期限までに解決するための管理表。

## プロジェクトにおけるリスクの探し方

リスク管理表の作成にあたって、自分のプロジェクトにおいては全てリスクだらけで何が本当にリスクなのかよく分からないという声を聞くことがあります。その結果リスク管理表は作成されず、リスクが問題化して初めて対応をとるといった事態に陥る場合が少なくありません。

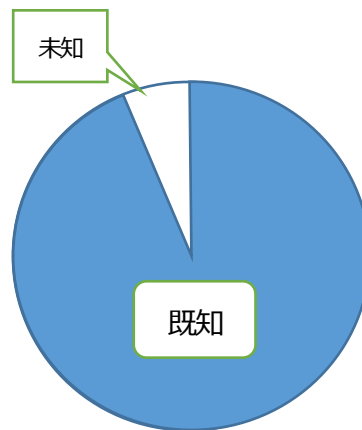
このような事態を避けるためには、リスクには既知のリスクと未知のリスクの二通りがあるということを知っておく必要があります。既知のリスクとは過去の失敗の履歴そのもののことです。未知のリスクとは今まで経験したことがないような新技術や複雑な仕様に関するリスクのことです。本書の対象である中小規模の派生開発プロジェクトにおけるリスクの99%は既知のリスクだと言っても良いでしょう。

まずは自分の開発組織で繰り返し起こしている障害やミスなどの失敗について、顧客および開発双方においてダメージの大きなものから順に記録をしていけばリスク管理表は容易に作成することができます。

実行

◎リスクには既知のリスクと未知のリスクの二通りがある。

【既知のリスクと未知のリスク】 (図4-1-2)



## 三つの時点で押さえるリスク

プロジェクトにおけるリスクは全工程にわたって存在していますが、各工程に均等な割合で散在しているわけではなく、ほとんどの重大なリスクは要件定義および見積り工程に集中していると考えて間違いはないでしょう。とりえず開発工程を前・中・後の三つに分けてどのようなリスクが散在しているのかを見てみることにします。解消すべき三つの時点のリスクは以下の通りです。



### ◆開発の入り口で押さえるべきリスク

- 適正な見積りにて妥当な開発期間と開発費を獲得すること。
- あいまいな要求仕様の明確化および早期の仕様凍結に全力を注ぐこと。
- 要求仕様書を常時メンテナンスし、“使えるドキュメント”として設計に引き継ぐこと。
- Q C Dの数値目標の設定を行うこと。



### ◆開発中に押さえるべきリスク

- あらゆる無駄の排除とリスク項目の解消を実行すること。
- 設計書をリアルタイムでメンテナンスし、常にプログラム内容との同一性を保持しておくこと。
- 失敗の真因・再発防止策をドキュメントとして残すこと。
- 創意工夫を設計書やガイドラインなどのドキュメントに残すこと。



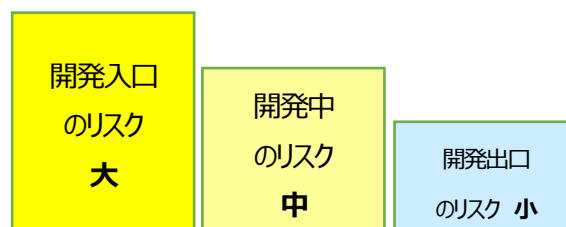
### ◆開発の出口で押さえるべきリスク（ラップアップミーティング<sup>48</sup>の実行）

- Q C Dの目標値と実績値を比較・分析し、問題の原因を数値データと共に明確にし、対策した結果をまとめておき、プロジェクト完了報告書にて他チームとも情報共有を行うこと。
- 失敗あるいは創意工夫の記録を振り返り、次の開発および他の開発者に申し送ること。

実行

## ◎リスクの三つの押さえ所、開発の入り口・中間地点・出口。

【三つの時点のリスク】（図4-1-3）



<sup>48</sup> ラップアップミーティング 開発完了時に行われる、開発行為の全体を振り返る会議。



## 開発現場にこそあるリスクの事例

リスクの察知は、プロジェクト自体の情報のみならず顧客の情報および全ての関係する組織の情報収集によって行われますが、特に有効なリスク情報は関係者たちとの直接コミュニケーションから得られる場合が多いものです。

突然遅延する進捗報告、肝心の要件が決まっていないこと、残っているはずの予算がないこと、突然の仕様の追加や変更、メンバーの突然の長期欠勤など、開発プロジェクトにおけるリスクにはさまざまなものがありますが、多くのリスク問題は結局コミュニケーションに起因する問題が多いのです。これらのリスク回避には毎日の積極的なコミュニケーションが必要不可欠です。

下記は現場におけるリスクの事例を17の категорияに整理分類したものです。

非常にたくさんのリスクが存在していることにあらためて驚き、新人のプロマネにおいては、とてもこのようにたくさんのリスクを一人では回避できないと思われることでしょうか、これらのリスク回避は決して一人のプロマネが背負いきれるものではなく、サブプロマネやプロジェクト全員で取り組むべき問題です。またあなたが現在担当しているプロジェクトのリスクは、この内のいくつかのリスクに限定されるのかも知れません。

### ◆【現場におけるリスクの事例】

17のリスク項目とリスク内容は以下の通りです。

- ①**不条理な顧客要求**
  - 決まらない要求仕様
  - 無理な短納期・低コストの要求
- ②**見積り問題**
  - 不十分な仕様調査・仕様理解
  - 見積り交渉力の弱さ
- ③**相互義務の不履行（元請・下請間、組織間、上司・部下間）**
  - プロジェクト統合管理の不在
  - 説明責任の欠如（丸投げ）
  - 情報共有・コミュニケーションの不足
- ④**仕様・納期問題**
  - 仕様決定遅れ
  - 短納期
- ⑤**コミュニケーション不良**
  - 顧客・開発チーム間の意思疎通・情報共有の不良
  - チーム内の意思疎通・情報共有の不良
- ⑥**情緒的開発姿勢**
  - 感情的・情緒的な思考・行動
  - 数値目標の不在
- ⑦**モチベーション問題**
  - モチベーションの低下
- ⑧**本質の把握ミス**
  - 自分の頭で考えない自律性の未熟さ

- ⑨優先順位問題
  - 割り込み作業等の優先度判断ミス
  - 必須業務に優先順位はないという認識の欠如
  - 納期第一優先の誤り
- ⑩ドキュメント・ベース開発の欠如
  - 開発行為の科学的根拠（ベースライン）の喪失
  - 低品質なドキュメントが招くQ C Dの失敗
  - 文書化能力の低下
- ⑪実務能力問題
  - 知識不足（技術・仕様・システム構造知識、評価テスト知識）
  - 仕様決定能力の低さ
  - 設計・製造能力の低さ
- ⑫リーダーシップ問題
  - マネジメントされないプロジェクト管理
- ⑬チームプレー問題
  - 相互義務の不履行・相互扶助の不在
- ⑭手抜き問題
  - 必要工程の中断ないしはスキップ
- ⑮時間認識問題
  - 仕様凍結期限意識の欠如
  - 期限・タイミング意識の欠如
  - 許容時間・必要時間認識の欠如
- ⑯人材育成／ノウハウの継承問題
  - 個人のスキル向上の疎外
  - 組織能力向上の疎外
  - プロジェクトQ C Dの低下
- ⑰学習能力の欠如問題
  - 失敗に学ばない
  - 振り返り（ラップアップ）の未実行
  - 改善活動の未実行

以上述べてきたように、詳細な事前調査とリスクの事前解消を行ってれば、開発の見積りを間違えたとか、仕様を間違えたとか、開発期間や費用が足りないとか、トラブルの多発などという問題も発生しにくいでしょう。

設計工程におけるリスク管理表の一例を次に示します。詳細サンプルは「付録図表 5. 設計工程のリスク管理表」を参照のこと。

【設計工程のリスク管理表】(表 4 - 1 - 1)

プロジェクト名：		初版発行日：		更新日：		文書番号：	
要因分類	QCD影響			設計工程リスク		✓	チェック
	Q	C	D				検出日
ヒトに関するリスク	①他者依存姿勢（自律性の放棄）	○	○	○	・協力会社への丸投げ（開発統合責任・製品最終責任の放棄）		
	②上位マネジメントの関与不足		◎				
	③ユーザーの参加・協力度不足	◎	○	○			
	④組織能力不足（未熟な組織文化、戦略の欠如）	◎	◎	◎	・オフショア開発		
	⑤見積り能力	◎	◎	◎			
	⑥要件定義能力	◎	○	○			
	⑦リーダーのプロジェクトマネジメント能力（外部交渉、タイムマネジメント、現場主義、見える化能力など）	◎	◎	◎	・未経験技術者の大量投入 ・基本設計レビューの未実施 ・安易なフレームワークの改修		
	⑧メンバーの技術能力、ヒューマンエラー（うっかりミス）	◎	○	○	・フレームワーク構築能力 ・異常系技術力の不足 ・タイミング設計能力の不足 ・排他処理設計能力の不足 ・非同期制御設計能力の不足 ・リトライ設計能力の不足 ・ログ技術力・認識不足 ・オープンソフト対応技術力不足 ・メモリーリーク対応技術力不足 ・性能（パフォーマンス・レスポンス）技術力不足		
	⑨プロセス管理の有無	◎	○	○	・プロセス管理不足 ・基本設計レビューの未実施 ・性能要件（パフォーマンス・レスポンス）レビューの未実施		
	⑩コミュニケーション能力（阻害・ギャップ）	◎	○	○	・組織間の協調性・コミュニケーション		
	⑪関連部署との連携不足	◎	○	○	・協力会社・オフショア先などのコミュニケーション能力		
モノ	⑫ドキュメントの不備（要件定義書・設計書・チェックリスト・手順書など）	◎	○	○	・要求仕様書なしの設計 ・貧弱なドキュメントやガイドライン		
	⑬開発のベースの有無	◎	◎	◎			
	⑭開発環境の不備	◎	○	○			
カネ	⑮開発費不足	◎	◎	◎			
	⑯資源投入戦略の誤り（開発費投入時期ミス）	◎	○	◎			
情報	⑰あいまいな開発範囲（スコープ）	◎	◎	◎	・あいまいなスコープ		
	⑱あいまいな要求仕様	◎	◎	◎	・あいまいな要件		
	⑲情報の不備・不足（マネジメント情報、技術情報、過去の失敗情報）	◎	○	○	・情報の不備・不足		

## 見えやすいリスクと見えにくいリスク

第2章においては事前準備工程のリスク、第3章においては要求仕様に関するリスク、本章においては設計工程におけるリスクを取り上げてきましたが、工程別リスクの視点を見えやすいもの／見えにくいものという視点で見てみた場合を図で表すと次のようになります。



見えやすいリスクを**表層リスク**<sup>49</sup>、見えにくいリスクを**深層リスク**<sup>50</sup>として分類した場合、表層リスクは氷山の一角にしか過ぎず、実は多くのリスクが深層リスクとして存在していることに気づきます。また表層リスクの多くは深層リスクに起因しているものが非常に多いのではないかと思います。

見えにくい深層リスクは、本書における Human Activity (人間的活動) すなわちヒトや組織文化に由来する場合が多く、一旦これらのリスクを見逃した場合の被害は広範囲かつ重大な問題を引き起こす結果になるものと思われます。

<sup>49</sup> **表層リスク** 要件定義・設計書・開発費などの不備や不足などのモノやカネに関するリスクで表面化しやすいリスクのこと。

<sup>50</sup> **深層リスク** ヒト・組織文化に由来する行動や仕事のやり方に起因するリスクのこと。

## 失敗に備えるコンティンジェンシープラン



### コンティンジェンシープランとは何か

コンティンジェンシープラン<sup>51</sup>とは不測事態対応計画とか、緊急時対応計画などと訳されていますが、予期しない失敗に直面した場合に備えるための計画のことです。

どのように綿密に考えられた計画であったとしても、計画した時点では予測できなかったリスクが問題化し危機状態に陥るプロジェクトも珍しくはありません。

この不測事態対応案が初期の計画において立案されていない原因としては、この計画でうまく行くはずだという思い込みや、この計画で行くしかないという**戦略的思考**<sup>52</sup>の狭さや、このような代替計画を考えること自体が自分の計画に自信がないと他人に思われたくない、というような体面ばかりを重んじる情緒的な考え方や、単に面倒くさいという怠惰性にあります。

実行

◎コンティンジェンシープランとは、予期しない失敗に直面した場合に備えるための計画のこと。



### 具体的なコンティンジェンシープランの例

コンティンジェンシープランはプロジェクトの達成目標であるQ C Dを支えるヒト・モノ・カネ・時間・情報に対して計画しておく必要があります。下記にそれぞれについて列挙してみます。

#### 【ヒト】

1. 一日 24 時間をフルに使うため、2 交代制あるいは 3 交代制の可能性について考えておく。  
開発機器類の夜間休止時間をなくすことや開発要員の負荷の平準化を図る効果があります。
2. 開発の**ボトルネック**<sup>53</sup>になりそうな人材の予備投入に備えて、レスキュー能力のある他の組織や人材の目星をつけておく。特にプロジェクト全体の指揮をとるプロマネ人材の増強は、開発チーム全体が**デッドロック**<sup>54</sup>状態になった場合に大きな効果を発揮します。単に設計者やプログラマーの増員では乗り切れない場面も多々あることに注意する必要があります。
3. 代替がきかない要員、とくに一人だけのプロマネや開発の**キーマン**<sup>55</sup>が不慮の病気や事故で欠員となった場合の代替要員について他部署・協力会社等広く人材の目星をつけておく。
4. 万が一の障害発生に備えて、年末年始や連休中における待機計画を作成する。

<sup>51</sup> コンティンジェンシープラン(Contingency Plan) 予期せぬ事態に備えて、予め定めておく緊急時対応計画・不測事態対応計画。

<sup>52</sup> **戦略的思考** 計画を成功させるための方策を考えること。

<sup>53</sup> **ボトルネック** (Bottle neck) 進行の障害や妨げとなるもの。

<sup>54</sup> **デッドロック** 行き詰ってしまった状態。いわゆる、にっちもさっちも行かない状態のこと。

<sup>55</sup> **キーマン** プロジェクトの成否のかなめとなる中心人物。キーパーソンとも言う。

## 【モノ】

1. 予備の開発機器・テスト機器の確保の可能性の検討を行う。

機器の不足による生産性の低下に効果があります。

2. 顧客交渉により成果物の優先順位の見直しを行う。

顧客に約束した成果物は約束した日にすべてリリースすべきですが、それが不可能だと判断された場合は速やかに顧客に報告すると同時に、顧客と相談の上、再度成果物の優先順位の見直しによる分割リリースの道を探らざるを得ないこともあります。一定のペナルティを覚悟する必要があるでしょう。

3. システム移行時の万一の失敗に備えて、旧システムに戻すための手順を考えておく。

## 【カネ】

1. 予備資金を用意しておく。

2. 戦略的リスク物件プロジェクトにおいては、経営層からの支援資金の約束を取り付けておく。

## 【時間】

1. 余裕日・余裕時間を計画に織り込んでおく。

2. 開発進捗遅延発生に備えて、延期が可能な業務、中止が可能な業務の目星をつけておく。

## 【情報】

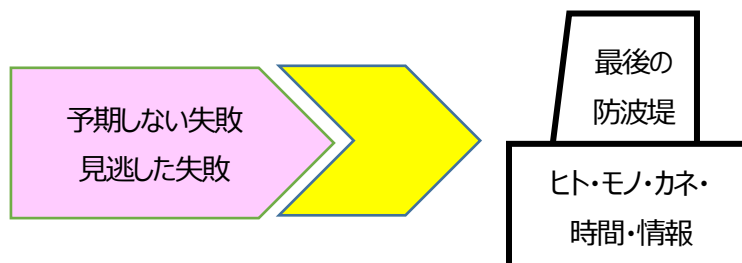
1. 情報漏洩に備えたセキュリティ対策を講じておく。顧客データ・開発データを持ち出せない仕組み、外部からの未承認データを持ち込ませない仕組みの構築を行っておく。

リスクのないプロジェクトなど存在しませんが、中小のプロジェクトはリスクが低いと見くびっていると、思わぬ失敗に見舞われないとも限りません。ソフトウェア開発全般においてどのようなコンティンジェンシープランがあるのかについて知っておけばきっと役に立つ機会があるでしょう。

実行

◎コンティンジェンシープランは、あなたのプロジェクトを救う最後の手段となる。

### 【コンティンジェンシープラン】（図4-1-5）



## 4-2. ドキュメントベース開発の実行

ドキュメントベース開発とは、開発業務のあらゆる面において、必ず何らかの書類・書面に基づいた情報の伝達や情報の共有を実行するという考え方のことを指しています。

あらゆる構造物を作るためには設計図が必要です。ソフトウェアの開発も当然のことに設計図に基づかなければプログラムは作成できません。

### ソフトウェア開発に必要なドキュメント

開発におけるドキュメントには主に、設計関連、開発管理関連、およびビジネス関連の三種類があります。設計関連ドキュメントには、何を作るかを示した要件定義書（要求仕様書）、どういう風に作るのかを表した設計書やガイドラインなどが含まれます。また開発管理表は、設計書にて定義されたものを完成させるためのリスクの排除・課題の解決・開発の段取り・進捗を管理するためのもので、リスク管理表、課題管理表、進捗管理表、予算管理表などがあります。以下に一覧で示します。

#### 【設計ドキュメント】

- 要求仕様書（要件定義書）
- 設計書： 基本設計書、詳細設計書、業務運用フロー、データフロー、システム論理構成図、プロセスフロー、ソフト構造図、修正影響度表、インターフェース仕様書、など。
- 手順書： 見積りガイドライン、プロセスガイドライン、設計手順書、コーディング規約書、評価手順書、構成管理手順書、など。
- ソースコード（コンピュータ言語記述文）
- 技術メモなど。

#### 【開発管理ドキュメント】

- 計画書： プロジェクト計画書、テスト計画書、など。
- 開発管理表： プロセス管理表、リスク管理表、課題管理表、予算管理表、進捗管理表、労務管理表、機材管理表、成果物管理表、障害管理表、など。

#### 【ビジネスドキュメント】

- 見積書
- 企画提案書

#### ◎ 整備状況・精度レベルの確認が必要な3種類のドキュメント

①設計ドキュメント ②開発管理ドキュメント ③ビジネスドキュメント

実行

## 開発ドキュメントの現状

開発者たちはドキュメントの現状について次のように語っています。

### 【開発者たちの生の声】

- ☆ システムの全体を表した資料がない。
- ☆ 品質は上流工程の要件定義書や設計書で決定する。テストだけでは品質は改善しない。
- ☆ 客先要件の実現方法が明確に書かれていない。
- ☆ 客先要件や仕様内容の記述が乏しく、ロジック記述に偏っている。運用テストに使えない。
- ☆ 開発内容に対するコンセプトや背景や経緯についての資料がない。
- ☆ 機能を実現するために必要な情報が設計書に記述されていない。
- ☆ 誰のために書かれたものか分からないドキュメントが多い。
- ☆ 作成者以外が見ても分かる内容・レベルになっていない。
- ☆ 内容が更新されていない。
- ☆ 納期が間に合わなくなったら、テスト作業やドキュメントを省略してしまう。
- ☆ 整理・管理されていないため調査・検索ができない。
- ☆ モジュール間の連携部分の説明が貧弱。
- ☆ 障害解析に使用できるレベルの内容になっていない。
- ☆ 障害報告書で技術的側面から図表等を使用した報告ができていない。
- ☆ 既存ソフトの流用の可否を判断できる資料になっていない。

このような状況を招いている原因については次のように語っています。

### 【できない理由】

- ☆ 前任者がドキュメントを更新していなかったから、自分もできなかった。
- ☆ 協力会社に設計以降の工程を丸投げする方が安く・納期短縮もできたから。
- ☆ 発注元が外注に丸投げする状況が続き、発注元の担当者が自ら設計する技術がなくなってきた。
- ☆ 発注元からドキュメントを要求されることもなく、ドキュメント作成工数が削減できた。
- ☆ バージョンアップ開発時にはソースベースで調査・改修も何とかなった。
- ☆ 時間に余裕がなく、とりあえず自分が分かるレベルの内容記述だけになった。客先・S Eや他の開発者・協力会社が理解できる内容にはなっていない。
- ☆ 時間も人もなく、限定した内容だけの記述しかできず、更新が必要な資料にも手をつけられなかった。

◎プロマネは自分のチームの本当の状況を現物確認した方がよい。

実行



## ドキュメントの不良が招いた失敗事例

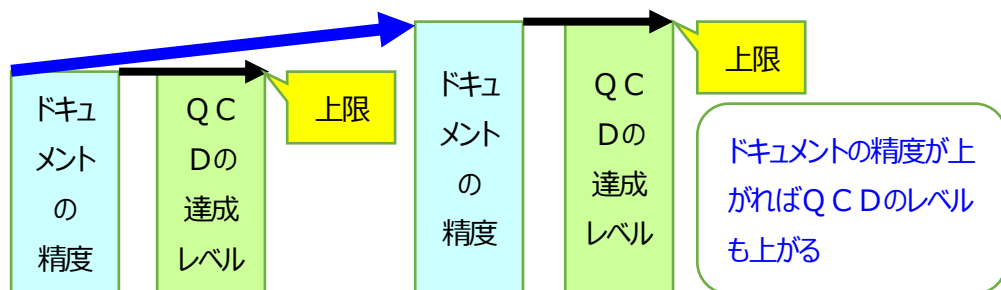
貧弱なドキュメントがどのようにひどい状況をもたらすのかを知るために、IPA・SEC<sup>56</sup>が公開しているプロジェクトの失敗事例の中からドキュメントの不備による失敗事例を紹介します。

- ① 「正式な要求仕様書がなく仕様検討を始めたが、二転三転する要求により変更が多発した。  
短納期のため、本来開発着手前に提示、または合意すべき要求仕様書がなく、また、ベンダーも強く要求していなかった。その結果、些細な点で要求仕様のズレが発生。完成度の低いシステムとなった。」
- ② 「開発ガイドラインがないために保守性、流用性が悪化してしまった。  
開発ルール、特にプログラミングの標準化、構造化のルールが未確定、またはメンバーに未浸透なため、プログラムが分かり難く、問題発生時の修正困難に加え、追加変更、移植性に多くの費用と工数を要した。」
- ③ 「手抜きドキュメントによるオフショア開発  
概要設計だけを示して委託したが、海外ソフトハウスはまだ設計できるほどのレベルにまでなっていなかった。そのため、海外ソフトハウス側でも勝手な思い込みで作り込みを続けた。プロジェクト側も、スキル不足で上がってくる中間成果物を十分に把握できず、システム規模が増え、作業遅れが発生した。」
- ④ 「計画書も手順書もない評価業務  
作業手順の誤りやテスト・データの誤りによる再テスト・修復という無駄な作業が多発し、テスト用マシンのCPU時間の半分以上を消費していた。これにより、テスト作業も遅延していた。」

◎ 妥当なレベルのドキュメントなしでは、QもCもDも達成することは不可能。

実行

【ドキュメントの精度とQC Dの達成レベル】（図4-2-1）



<sup>56</sup> IPA・SEC 独立行政法人情報処理推進機構（IPA）技術本部ソフトウェア高信頼化センター（SEC）の略称。

## ドキュメント精度の簡易チェック法



### 基本設計書の簡易チェック方法

簡易的に基本設計書の完成度をチェックする方法として、その基本設計書を使ってある重要な仕様の一部分について詳細設計書を書いてみればはっきりするでしょう。ちゃんとした詳細設計書が書けなければ、その基本設計書はまだ不完全だということです。詳細設計が書けない部分が、基本設計書における欠落部分や矛盾している部分なのです。

また基本設計書は要求された機能の基本的な骨格を定義した設計図であり、基本設計書と対をなしている仕様書としては結合テストにおいて使用される結合テスト仕様書があります。結合テスト仕様書は開発されたソフトウェアの基本的な機能が正しく動作することをテストするためのドキュメントです。基本設計書の一部分について結合テストの担当者に、結合テスト仕様書が書けるかどうかを試してみても基本設計書の精度が確認できるでしょう。

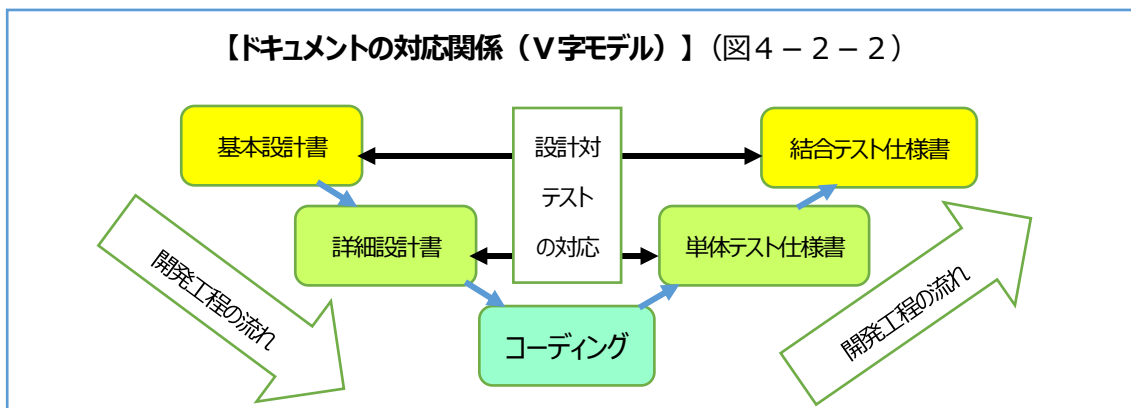


### 詳細設計書の簡易チェック法

簡易的に詳細設計書の完成度をチェックする方法として、その詳細設計書を使ってある重要な仕様の一部分についてプログラムを書いてみればはっきりするでしょう。ちゃんとしたプログラムが書けなければ、その詳細設計書はまだ不完全だということです。プログラムが書けない部分が詳細設計書における、欠落部分や矛盾している部分なのです。

また詳細設計書は要求された機能を最小機能単位に分割し、それを定義した設計図であり、詳細設計書と対をなしている仕様書としては単体テストにおいて使用される単体テスト仕様書があります。単体テスト仕様書は開発されたソフトウェアの最小機能単位が正しく動作することをテストするためのドキュメントです。詳細設計書の一部分について単体テストの担当者に、単体テスト仕様書が書けるかどうかを試してみても詳細設計書の精度が確認できるでしょう。

【ドキュメントの対応関係（V字モデル）】（図4-2-2）



## リアルタイムなドキュメントのメンテナンスが必要な訳

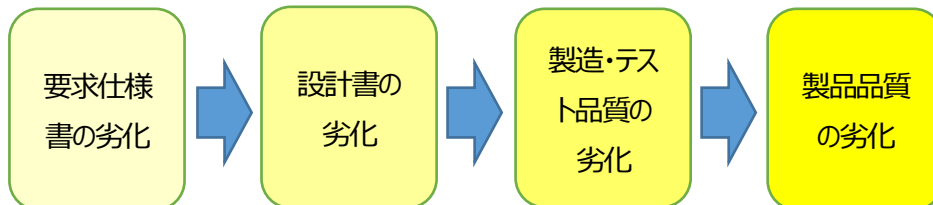
黙っていてもドキュメントは更新されません。更新されていないドキュメントの内容は何が「正」で、何が「誤」なのかを判断することができません。段々とメンテされなくなったドキュメントは、ついには誰にも使用されない大量の紙クズとなってしまいます。開発ドキュメントは常にその鮮度を保っておく必要があり、生鮮食品の一種だと思った方がよいのです。常に鮮度を保つ管理をすることが、不要な労力を省き、スムーズな開発を実現することにつながります。

信頼できる設計書がなければソースコードの解析による開発となり、まちがいなくその設計品質は劣化します。設計品質が劣化すれば製造品質も評価品質も劣化します。いつまでたっても設計書は更新されないまま、品質劣化の悪循環が繰り返されることとなります。設計書の前の要求仕様書がなければもっと大変な結果となります。

◎ドキュメントの誤記・抜け・追加は、気づいた時点でリアルタイムな修正が必要。

実行

【ドキュメントの劣化が招く、品質の劣化】（図4-2-3）



## 開発管理表の概要

プロマネにとって開発管理表はプロジェクトの進行をコントロールする重要なドキュメントです。  
下記は主要な開発管理表についての役割を整理したものです。

### 【開発管理の三種の神器】（図4-2-4）

#### ◎ プロセス管理表

主要イベント・成果物の実行チェック

#### ◎ リスク管理表

全工程にわたるリスクの掘り起こしと解消

#### ◎ 課題管理表

日々発生してくる課題対策の立案と解決



### プロセス管理表

プロジェクト管理の基本中の基本はプロセス管理です。プロセス管理は分かりやすく言うと仕事の段取りであり、プロセス管理表はプロジェクトの重要なイベントを時系列順に並べた開発手順のチェックシートだともいえます。各々の詳細な工程に対して、やるべき作業内容、誰が実行するのか、作成される成果物は何か、などを記述し、実施予定日・実施日や備考などを記録することで、個々の作業が適切な内容で確実に実施されたことを保証する証拠となるものです。

第1章の「1-4. プロセス管理」および「付録図表1. プロセス管理表」を参照のこと。

各工程が複数の会社にて分業化されている場合は、各工程の担当会社ごとに担当工程のプロセス管理表を作成・運用し、発注元の会社においては全体をまとめた統合プロセス管理表の作成・運用が必須です。



### リスク管理表

リスク管理表は開発全工程にわたるリスクを管理するものです。プロジェクトにおけるリスクとしては次のようなものがあります。

仕様凍結遅れ、仕様変更、進捗状況、責任の所在、開発規模、他組織との関連、要員の確保、外注コントロール、短納期、プロセスの遵守、見積り精度、性能問題、保守性問題、プログラム構造、設計ミス、製造ミス、評価ミス、構成管理ミスなど。

これらの情報を網羅したリスク管理表の作成およびリスクの解消はプロジェクトの成否を分ける必須の業務です。

第4章の「4-1. リスクの解消」および「付録図表2、4、5の各リスク管理表」を参照のこと。



## 課題管理表

リスクが実際に問題化したものが課題です。課題管理表は、日々発生してくる課題をその対策アクションとともに記録するための管理表です。課題管理のポイントは、効率的な課題解決のために各々の課題に優先度をつけること、およびその実行を確実なものにするために実行担当者名および期限を明記することです。

課題とリスクの違いがよく分かってない人が意外と多いのですが、リスクはまだ問題が表面化してはいないが表面化する危険性のある潜在的な課題のことです。リスクはそれが表面化する前にその芽を摘み取っておくような対策や行動が必要です。例えば仕様凍結が遅れそうな心配があるなら、こちらから先に仕様提案をすとか、仕様を決める責任部署に人員強化の依頼を出しておくとか、をすることがリスク回避のアクションとなります。

一方課題は必ず解決されなければならない問題です。課題はあなたの都合を待ってはくれません。期日までにその問題を解決しなければ、その問題はさらに大問題へと発展してしまい、Q C Dに大きな傷を負わせ、プロジェクトを止めてしまう結果を招くかもしれません。

プロマネの毎日の仕事は、課題の解決だと言っても言いすぎではないでしょう。プロジェクトにおいてはリーダーから担当者にいるまで実に多くの課題・問題を抱えています。何も問題を抱えていませんと発言する人は、何も仕事をしていない人だと思った方がいいでしょう。課題管理表なしではまっとうに開発は、やり遂げられないものです。

### 【課題管理表】（表4-2-1）

下記の課題管理表の記述欄は左から課題No.、優先度、警告、発生日、期限、完了日、記入者、発生工程、課題項目、課題内容、課題対応策、処理担当者名、対応状況、備考欄となっています。

No	優先度	警告	発生日	期限	完了日	記入者	発生工程	課題項目	課題内容	課題対応策	担当	発生作業/進捗状況/結果	備考
1													
2													
3													
4													
5													
6													
7													

サンプルの拡大版は「付録図表6. 課題管理表」を参照のこと。

### 4 - 3. 優先順位ベース開発の実行

#### 開発業務における実行の優先順位

一般的なものごとの実行の優先順位は次の通りです。

1. 顧客価値の高い順。
2. 次に、効果が大きく実行が容易なものの順。
3. タイムリミットが必須のものはその期限までに実行しなければならない。

優先順位に従った開発業務の実行手順は次のようになります。



#### 要求仕様を顧客価値の重要度順に整理すること

顧客価値の重要度の順位は、顧客との密接なコミュニケーションを図ることで、概要要求仕様の検討段階で判断可能です。発注側・受注側において仕様の概要が分かった時点で、仕様開発の優先度の順位を決定・合意することが重要です。優先度の順位は、リリース時期の先・後を考慮に入れた上での顧客価値の重要度の順とすべきでしょう。

#### ◎ 無駄な開発の排除のために

- ① 顧客価値のない要件を排除すること。
- ② 顧客価値の低い要件の優先順位を下げること。

実行



### 顧客価値の重要度順に仕様を凍結していくこと

仕様凍結にあたっては、仕様を全部一時に決めようとしないで、重要度順に凍結していくことが必要であり、更に顧客価値を満足させる要件定義の実行、すなわち開発目標の明確化が必要です。これらは要件定義にあたって、顧客と開発チームの密接なコミュニケーションをベースに、顧客の要求について顧客価値の優先順位付けを行い、優先度の高いものを顧客の必要とする時期に提供していく計画を立て、さらには顧客価値の低い開発行為の優先順位を下げることで、顧客にとっても必要性の低い開発行為を排除すること、などを通して顧客の要求と開発行為を一致させることによって実現されます。

これらの要件定義における基本的な取り組みの実行は、要件およびその範囲（スコープ）の明確化をメリハリのあるものにし、その変更管理のコントロールを容易にします。さらに要件の仕様化にあたっては仕様の階層構造的記述（WBS）が、仕様ミス防止・解釈違い防止・矛盾仕様の発見・正確性・分かりやすさに寄与するでしょう。

#### ◎有効な開発に要するエネルギーの最小化を図るために

実行

- ① 顧客の要求と開発行為を一致させること。
- ② 顧客価値の優先度の順・必要とされる時期に成果物を提供していくこと。



### 集中仕様凍結会議の実施

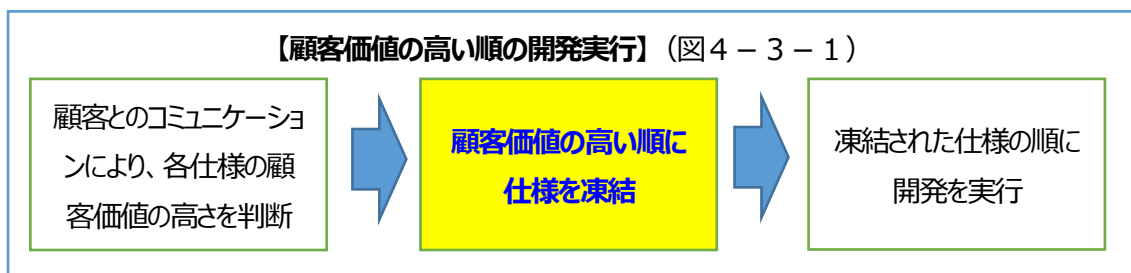
仕様検討に時間がかかりそうな場合は、日時を決めた上で顧客と直接顔を付き合せ、こちらからも対案を出し、検討会や宿泊で一気に仕様決定を行う必要があります。相手が決めてくれるのを待っているような受け身の姿勢では何も改善できません。



### 顧客価値の重要度順に凍結した仕様から開発着手を行うこと

顧客との間で合意された優先度の順位に従って、詳細仕様の決定、WBSの作成に基づいたスケジュールリングを行い開発に着手します。顧客との間で合意された開発の実行順位に関する情報は、逐次プロジェクトチーム内の全員に共有される必要があります。

【顧客価値の高い順の開発実行】（図4-3-1）



## QCDには優先順位はつけられない

プロジェクトの成功と失敗について、日経BP社による2009年の「プロジェクト実態調査800社」によればプロジェクトの成功率は31.1%だったという分析結果が出ています。

(Q 品質は成功 : 51.9%、C コストは成功 : 63.2%、D 納期は成功 : 54.6%)

この調査結果から見えることは次の通りです。

開発会社は表向き品質や納期優先だとは言っていますが、多くの失敗プロジェクトにおけるQCDの結果を見る限りにおいて、その実態はコスト、すなわち利益第一優先で、見かけ上納期を守ったように装い、品質を犠牲にしている姿が見えてきます。

すなわち、現実のQCDの優先順位はコスト(利益) > 納期 > 品質の順になっています。

これで良いのでしょうか？これで良いわけがありません、顧客は怒ります。

会社を潰さないようにする視点で見れば、コスト優先になるのでしょう。コストはマネージャ以外には見えにくい指標ですから、一般の開発者に見える部分では納期優先となってしまいます。約束の時期を外すことはできないために、最後の評価・デバッグの時間が不十分だったとしても、タイムアウトということになり、バグ含みで出荷されるソフトウェアが後を絶ちません。コストを守るために、必要な開発時間をカットして納期の帳尻を合わせ、そのために品質を犠牲にしているという図式です。もしこの通りの優先順位で実行したら、顧客の満足や品質第一が優先だと言っていることは嘘になるでしょう。悪い品質のものを買わされて被害を受けるのは顧客自身です。

品質・コスト・納期に優先順位はなく、本来同列の価値を持つものです。皆必須条件なのです。どれかだけを優先し他を犠牲にすることはできません。人間において、脳を取りますか心臓を取りますか選んで下さいというのと同じです。

やるべきことは、その様な究極の選択をしなければならぬような状況に陥らないようにすることです。失敗するプロジェクトの主な原因は、仕様凍結遅れ・開発プロセスの悪さ・リスク回避の失敗などにあります。つまり「何をつくるか」「どうつくるか」が不明確なため、大きな時間ロスを生んでいるということです。現実的に色々な悪条件が存在するプロジェクトにおいては、上記のことを十分に承知した上で、妥当な品質・妥当なコスト・妥当な納期を追及することがポイントになります。

◎ QCDに優先順位を設けてはいけない

実態は

コスト(利益) > 納期 > 品質

実行



## 第5章

### 管理工程のリスク

#### 5-1. 要求仕様の変更管理

一旦、顧客と開発側において開発すべき仕様および開発条件（開発費・開発期間・品質条件）などが合意されたとしても、その後に顧客側ないしは開発側の都合によって変更や追加の要求が出された場合、最初に約束した内容を守ることが難しくなってしまいます。特に仕様の度重なる変更や追加は開発費・開発期間・品質に重大な悪影響を与えてしまいます。

#### ベースラインの意味

見積り回答によって顧客と約束した、開発費・開発スケジュール・品質条件および開発仕様の内容・範囲のことを**ベースライン**<sup>57</sup>と呼び、このベースライン確定後の開発条件の変更に対する顧客との交渉を伴う処置を**変更管理**と呼んでいます。

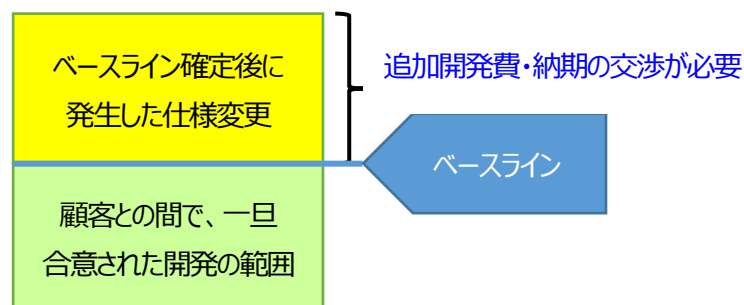
開発側においては、大きな変更によるQ C D等に対する重大な悪影響を避けるために、最初に約束したものの（ベースライン）とその後に発生したもの（変更・追加）を分けて処理する必要があります。追加・変更分は基本的に追加開発費・追加開発期間で処理される必要があります。

管理

◎**ベースライン（基線）とは、**

**プロジェクトマネジメントにおいては、顧客と開発側で合意した成果物の範囲および基準の事を指す。両者間で約束した仕様とそのQ C Dに関する基準は重要なベースラインとなる。**

【ベースラインと仕様の変更管理】（図5-1-1）



<sup>57</sup> **ベースライン** 見積り回答によって顧客と約束した開発費・開発スケジュール・品質条件および開発仕様の内容・範囲のこと。

## 変更管理が行われにくい理由

変更管理が行われにくい理由として最初に考えられることは、開発のスコープ・予算・期間を決定してしまうベースラインの確定そのものを顧客側が嫌うということがあります。

一般的に顧客側は限られた予算と期間の中で、できるだけ多くの仕様を盛り込みたいと思っていますが、ベースラインの確定はそれに対する縛りとなってしまいます。

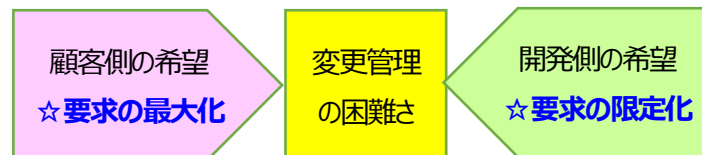
顧客側の考え方は、できるだけ良いものをできるだけ安く手に入れたいという所から出ているのですが、そのように都合の良いことがいつでもあるわけがなく、相応の価値あるものに相応の対価を払うという健全なビジネスの基本からは逸脱する考え方と言わざるを得ません。

余談になりますが、優れたビジネスを行っているある顧客は「品質にはそれ相応の対価を支払う」と言っており、その顧客の値引き要求に対する回答に対しても削減の根拠の提示を求められました。根拠のない値引き提案など受け付けられないということなのです。

◎健全なプロジェクトには、ベースラインの確定が必要である。

管理

### 【変更管理の困難さ】（図5-1-2）



## 要求仕様の変更管理不在がもたらす災い

要求仕様のベースラインの確定も変更管理も実行されないならば、顧客側は白紙委任状<sup>58</sup>を手にしたのも同然で、早期の仕様まとめの動機も失われ、顧客のあちこちの部署からさみだれ的に集まってくる要求をさみだれ的に開発会社に投げ続けることになります。これが二転三転する要求仕様の本当の原因なのです。要求仕様を考える能力がないのではなく、要求仕様を決められた時期までにまとめる意思がないだけのことなのかも知れません。

このように際限なく続く仕様の変更や追加は、ついにはプロジェクトの処理能力を超えてしまい、多くの手戻り作業や追加開発により、膨大な時間を失い、品質劣化、コスト高、生産性低下を招いてしまいます。

この結果は顧客側にもブーメランのように戻ってくることになり、顧客は納期遅延や障害多発に見舞われ、最悪の場合は顧客のビジネス自体が止まってしまうこともあります。

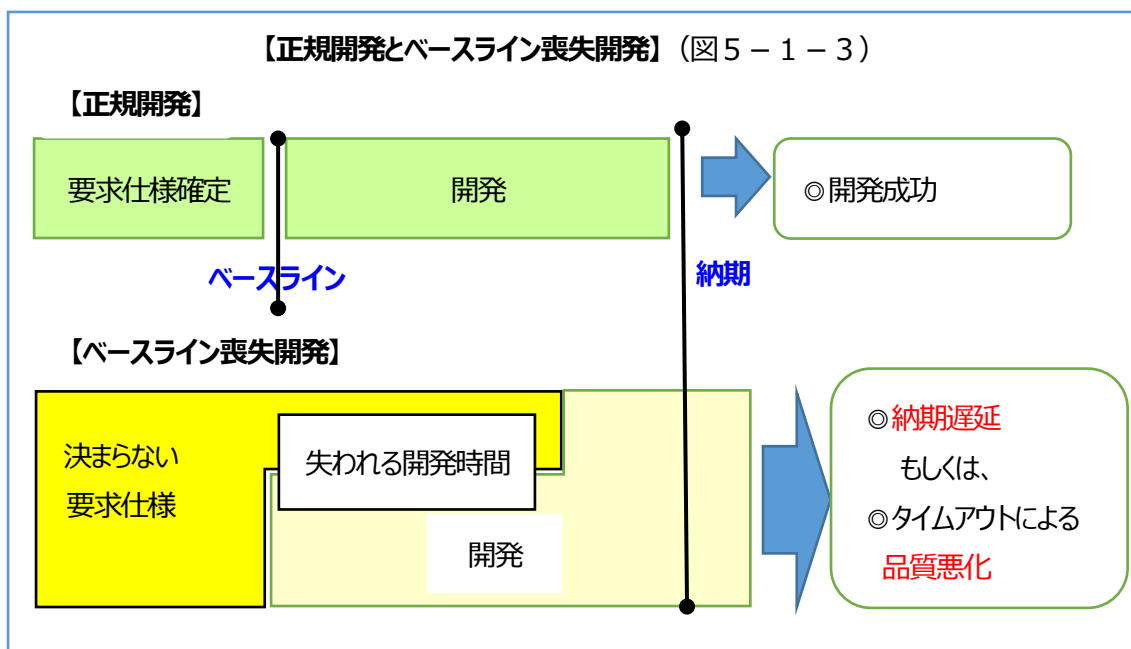
ベースラインの確定も変更管理の実行もなされないプロジェクトは、顧客側・開発側双方において致命的な結果を招くという認識が必要です。

管理

### ◎ベースラインの確定・変更管理の不在がもたらすリスク

☆プロジェクトの失敗：QCDの大幅な未達

☆顧客ビジネスの停止：市場障害の多発



<sup>58</sup> 白紙委任状 ものごとの決定権を相手に委ねてしまうことの例え。

## 仕様変更管理表がもたらす効果

安易な仕様変更にしっかりと歯止めをかけるためには、仕様変更管理表の運用が不可欠です。

この管理表の主な目的は、仕様変更の影響を管理し、顧客側の無定見な仕様変更・追加を抑制することにあります。特に顧客と開発側で同意した仕様凍結日以降に発生する仕様変更を厳重に管理することが重要であり、同意された仕様凍結日以降に発生した変更・追加仕様は基本的には別途見積り（追加コスト・追加開発期間）であることを顧客側が明確に認識するために使用されます。仕様変更管理表の運用は下記のような効果をもたらします。

### 【仕様変更管理表の効果】

1. 放縦・無定見な仕様変更の防止
2. 顧客・開発側両者に対する仕様凍結期日の厳守化（開発仕様のベースラインの確定）
3. 情緒的な要件定義工程のプロセス化
4. 仕様変更の質、量および費用の厳格な管理

管理

◎ベースライン確定後の仕様変更・追加は別途見積りとする交渉が必要。

仕様変更管理表には、管理番号、発生日、対応期限、対応日、対応状況、業務名、変更内容、変更区分、修正担当、影響度範囲、難易度、リリースバージョン、見込み対応工数、実績対応工数などを記録し、仕様の変更状況を明確にしておく必要があります。仕様変更管理表のフォーマットは次のようなものになります。

【仕様変更管理表】（表5-1-1）

【仕様変更管理表】														
No.	管理番号	発生日	対応期限 (ベースライン)	対応日	対応状況	業務名	変更内容	変更区分	修正担当	影響度 範囲	難易度	リリース バージョン	対応工数 (見込)	対応工数 (実績)
1	PJ-001										A			
2											B			
3											C			
4														
5														
6														
7														
8														
9														
10														
合計														

拡大図表は、付録図表7. を参照のこと。

付録図表

1. プロセス管理表：第1章1-4. プロセスをマネジメントする、表1-4-1

【プロセス管理表】(サンプル)			*PL (or PM) , SPL (or SPM)			担当 チェック	PL チェック	実施 予定日	実施日	レビュー 結果
NO	手順	進捗	作業	担当者	成果物					
事前準備	1 顧客・ステークホルダーの情報収集		顧客の要求内容・予算・納期等の事前把握	開発部	・調査報告書					
	2 新OS・新言語・新システム対応の準備		新ソフトウェア・新ハードウェア等に対する技術知識のマスターおよびプロトタイプによる事前調査	開発部	・調査報告書					
計画プロセス	1 見積もり依頼 見積もり回答		仕様、スケジュール、要員検討、見積依頼	PL, SPL	・見積回答書 ・リスク管理表					
	2 要求仕様の明確化		要求事項の明確化	PL	・要求事項の明確化チェックリスト作成					
	3 仕様検討、承認		SEとの仕様打ち合わせ	SE, PL, 開発メンバー	・議事録 / 要求仕様書の承認					
	4 不具合歯止め策の選定、決定		開発に対する有効な歯止め策を	PL, SPL, 開発メン	・歯止めチェックリスト					
	5 スケジュール		大、小日程の作成	PL, SPL	・大日程表、小日程表					
	6 プロジェクト品質目標の作成		プロジェクト内品質目標の作成	PL, SPL	・プロジェクト品質目標					
	7 プロジェクト計画書		プロジェクト計画書の作成 リスク管理表更新	PL, SPL	・プロジェクト計画書					
			プロジェクト計画レビュー 開発開始会議	SQAグループメンバー、PM, PL	・レビュー記録					
	8 外注見積委託作成		外注見積委託DBに登録	PL, SPL	・見積書					
	9 開発管理		開発管理資料の作成	PL, SPL	・プロセス管理表 ・スケジュール表 ・課題管理表 ・規模進捗					
10 デザインレビュー①		要求事項の明確化チェックリスト	SE→開発部	・要求仕様の採点確定						
実行・管理プロセス	1 仕様書		仕様書 仕様書レビュー	担当者 PL, SPL, 開発メン	・仕様書各種 ・レビュー報告書					
	2 初期設計		初期設計 初期設計レビュー、スケジュール	各担当 PL, SPL	・変更モジュール一覧表 ・レビュー報告書 / 変更モジュール一覧表					
	3 デザインレビュー②		SE確認；要求事項の明確化 チェックリスト確認	開発部→関係各位	・仕様書 ・設計書					
	4 詳細設計		詳細設計 詳細設計レビュー	各担当 SPL, 開発	・詳細設計書 / リリース影響度表					
	5 部材の確認		部材の変更	SPL	・新規部材					
	6 製造		製造 ソースレビュー	各担当 SPL, 開発	・ソースファイル / 影響度表 / 変更モジュール ・ステージ別チェックリスト / 処理条件マトリクス					
	7 評価 (単体)		チェックリスト作成 チェックリストレビュー デバッグ	各担当 SPL, 開発 各担当	・チェックリスト ・レビュー報告書 ・チェックリスト更新 / 試験結果					
	8 評価 (結合)		デバッグレビュー チェックリスト作成 チェックリストレビュー 開発チーム評価	各担当 各担当 SPL, 開発 各担当	・レビュー報告書 ・テストチェックリスト ・レビュー報告書 ・試験結果					
	9 トキメント		・仕様書フィードバック ・設計書フィードバック	各担当	・レビュー報告書 ・仕様書 ・設計書					
	10 総合テスト、第三者検証 品証検証 客先検証		チェックリスト作成 チェックリストレビュー 評価 評価レビュー	評価 SPL, 評価 評価担当、品証担当 PL, SPL, 評価	・チェックリスト ・レビュー報告書 ・チェックリスト更新 / 試験結果					
	11 デザインレビュー③		客先検証 要求事項のチェックリスト確認	開発部→SE、客先 品質保証部→関係部署	・客先検証報告受領 ・総合テスト報告書 / 第三者検証報告書					
	12 店舗リリース後確認 不適合の監視		実験店稼働確認チェック 実験店全ログチェック 実験店全ログチェック結果レ 全国展開	SPL, 開発 SPL, 開発 PL, SPL	・実験店稼働チェックリスト ・実験店チェックリスト ・レビュー報告書 ・PG交換障害報告書					
	13 トキメント整理		トキメント整理	開発部						
振り返り・継承	1 開発完了作業		開発完了報告	開発部→業務部	・開発完了報告書					
			プロセスの測定、分析、改善、歯止めの成果確認 開発予算との整合性チェック	PL, SPL PL, SPL	・プロセスの測定、分析、改善結果 ・プロセス管理表に見積もり金額の記入					
			他プロジェクトへのノウハウの継承	開発部	・重要ノウハウの他部署への情報共有の実行 ・重要ノウハウのDBへの登録					

2. 事前準備工程のリスク管理表：第2章 2-3. 事前準備工程のリスク、表2-3-1

リスク管理表（事前準備工程）	リスク要因分類	Q	C	D	初版発行日	更新日	文庫番号	担当姓名	リスク管理項目	対策アクション	対策日	備考
	①他職依存性リスク（自律性の放棄）	○	○	○	リーディングの欠如 ・マルチベンダー下における各担当責任のあいまいさ ・他社ベンダーへの未検証採用 ・業務技術の遅延（おのれ任せ） ・開発組織の自律性不足 ・顧客交渉戦略・能力不足 ・不適切なリソースの選任							
	②上位マネジメントの関与不足	○	○	○	・顧客の参加・協力度が低い ・不適切な進捗管理（見込み納め） ・新旧システムの同時並行開発（開発量・時間の重複）							
	③リーダーの経験・能力不足（未知な組織文化、戦略の欠如）	○	○	○	・顧客との名目だけの共同研究開発 ・能力不足のプログラマーの選任 ・明確な更新によるシステムの劣化・スバクタイ化 ・安易な移行機能搭載の保証							
	④見直し能力	○	○	○	・フロント・バックアップ能力 ・要件定義能力不足 ・顧客交渉戦略・能力不足 ・不適切な進捗管理 ・新旧システムの同時並行開発							
	⑤要件定義能力 （プログラマーのシステムマネジメント能力 （外資交渉、タスクマネジメント、現場 主義、見える化能力など）	○	○	○	・正面能力不足 ・追加の業務依頼開発能力不足 ・問題の放置 ・未検証言語採用の準備不足 ・開発目的の誤り（受注確保） ・技術者のトレーニング不足 ・開発体制の不備（経験者・能力・知識・人材） ・未検証分野／業務参入の準備不足							
	⑥プログラマーの技術能力、ヒューマンエラー （ラッパミス）	○	○	○	・新技術採用の準備不足 ・バックアップ機能におけるフロント&バックアップ開発不足 ・技術方針選定・対応における柔軟性 ・オープンシステム知識の不足 ・新技術知識不足 ・顧客業務知識不足							
	⑦プロセスマネジメントの有無	○	○	○	・顧客とのコミュニケーション能力 ・顧客のパートナー化失敗 ・顧客とのコミュニケーション能力 ・組織間の協調性・コミュニケーション							
	⑧計画・チャットリスト・手順書など	○	○	○	・開発ガイドラインの不備（開発プロセス、設計手順書、コーディング規約、単体・総合テスト手順書等）							
	⑨開発のベネフィットの有無	○	○	○	・開発のベネフィットの有無 ・開発費不足（見積りの失敗、開発の失敗） ・赤字発生							
	⑩開発環境の不備	○	○	○	・開発環境の不備 ・顧客投入戦略の誤り							
	⑪顧客投入戦略の誤り（開発費投入 時期ミス）	○	○	○	・顧客投入戦略の誤り							
	⑫私利私欲の発生（スコープ）	○	○	○	・あいまいな要件定義 ・あいまいな要件仕様							
	⑬情報の不備・不足（マネジメント情 報、技術情報、過去の失敗情報）	○	○	○	・情報の不備・不足 ・未検証分野／業務の業務知識・技術情報不足 ・新技術の技術情報不足							

3. 要求仕様書 採点表：第3章 3-1. ●要求仕様書の精度検証、表3-1-2

項番	分類	大項目	中項目	要求仕様内容	記述必須項目	記述の有無	記述内容得点	備考
1		1	1	表紙・変更履歴・目次	1	1	3	
2		2	2-1	システム化の理由	1	1	3	
3		3	2-2	システム化前後の運用	1	1	3	
4		4	2-3	システム化のメリット	1	1	3	
5		5	3	システム構成図	1	1	3	
6		6	4	ハード構成				
7		7	5	ソフト構成	1	1	2	
8		8	6	運用一覧				
9		9	7	運用スケジュール	1	1	3	
10		10	8	概要ビジネスフロー (運用フロー)				
11		11	9	業務一覧	1	1	3	
12		12	10	ビジネスルール定義				
13		13	11	詳細ビジネスフロー				
14		14	12	画面一覧				
15		15	13	画面設計	0			
16		16	14	画面項目説明				
17		17	15	帳票一覧				
18		18	16	帳票設計				
19		19	17	帳票項目説明				
20		20	18	機能定義書	1	x	x	
21		21	19	画面遷移図				
22		22	20	要求データ一覧				
23		23	21	要求データ仕様書	1	1	2	
24		24	22	オンライン一覧	1	1	2	
25		25	23	要求性能	1	1	2	
26		26	24	その他				
<b>合計</b>					<b>12</b>	<b>11</b>	<b>29</b>	

開発担当部署：XXXXX  
採点者：XXXXX

- 0：記述なし  
1：仕様として不十分  
2：多少不明点はあるが内容はわか  
3：要件として十分

\* 記載が80点未満の場合、基本的には見積不可とする。  
\* 記述内容得点が60点未満の場合、基本的には見積不可とする。

満点ポイント	12	36	記述内容得点 = 12×3 = 36点
得点ポイント	11	29	
得点/100点得点換算	92	81	記述内容得点 = (29/36)×100 = 81点

記載率

4. 要件定義工程のリスク管理表：第3章 3-2. 要求仕様に関するリスク、表3-2-1

リスク管理表 (要件定義工程)	リスク名	更新日	文書番号	担当	リスク管理項目	
					リスク内容	対策日
リスク管理表 (要件定義工程)	要因分類	① 他者依存的要因 (自律性の放棄)	① C D	①	要件定義工程の進捗、納期遅延	①
		② 上位マネジメントとの関与不足	② C D	②	顧客交渉戦略、能力不足 大規模な仕様追加要求の対応 大規模な納期前追加要求の対応 大規模な納期前追加要求の対応 顧客の参加、協力が低い 見直しプロセスの手続きルール違反	②
ヒトに関するリスク	要因分類	③ ユーザーの参加・協力が低い	③ C D	③	顧客交渉戦略、能力不足 大規模な仕様追加要求の対応 大規模な納期前追加要求の対応 顧客の参加、協力が低い 見直しプロセスの手続きルール違反	③
		④ 組織能力不足 (柔軟な組織文化、戦略の欠如)	④ C D	④	顧客交渉戦略、能力不足 大規模な仕様追加要求の対応 大規模な納期前追加要求の対応 顧客の参加、協力が低い 見直しプロセスの手続きルール違反	④
ヒトに関するリスク	要因分類	⑤ 見直し能力	⑤ C D	⑤	顧客交渉戦略、能力不足 大規模な仕様追加要求の対応 大規模な納期前追加要求の対応 顧客の参加、協力が低い 見直しプロセスの手続きルール違反	⑤
		⑥ 要件定義能力	⑥ C D	⑥	顧客交渉戦略、能力不足 大規模な仕様追加要求の対応 大規模な納期前追加要求の対応 顧客の参加、協力が低い 見直しプロセスの手続きルール違反	⑥
ヒトに関するリスク	要因分類	⑦ リーダーのプロジェクトマネジメント能力 (外部交渉、タイムマネジメント、現場主義、見える化能力など)	⑦ C D	⑦	顧客交渉戦略、能力不足 大規模な仕様追加要求の対応 大規模な納期前追加要求の対応 顧客の参加、協力が低い 見直しプロセスの手続きルール違反	⑦
		⑧ マネージャーの技術能力、コミュニケーション能力	⑧ C D	⑧	顧客交渉戦略、能力不足 大規模な仕様追加要求の対応 大規模な納期前追加要求の対応 顧客の参加、協力が低い 見直しプロセスの手続きルール違反	⑧
ヒトに関するリスク	要因分類	⑨ プロセス管理の精確	⑨ C D	⑨	顧客交渉戦略、能力不足 大規模な仕様追加要求の対応 大規模な納期前追加要求の対応 顧客の参加、協力が低い 見直しプロセスの手続きルール違反	⑨
		⑩ コミュニケーション能力 (相違・ギャップ)	⑩ C D	⑩	顧客交渉戦略、能力不足 大規模な仕様追加要求の対応 大規模な納期前追加要求の対応 顧客の参加、協力が低い 見直しプロセスの手続きルール違反	⑩
モノ	要因分類	⑪ 関連部署との連携不足	⑪ C D	⑪	顧客交渉戦略、能力不足 大規模な仕様追加要求の対応 大規模な納期前追加要求の対応 顧客の参加、協力が低い 見直しプロセスの手続きルール違反	⑪
		⑫ システムの不備 (要件定義書・設計書・チャートリスト・手順書など)	⑫ C D	⑫	顧客交渉戦略、能力不足 大規模な仕様追加要求の対応 大規模な納期前追加要求の対応 顧客の参加、協力が低い 見直しプロセスの手続きルール違反	⑫
モノ	要因分類	⑬ 開発のベースの精確	⑬ C D	⑬	顧客交渉戦略、能力不足 大規模な仕様追加要求の対応 大規模な納期前追加要求の対応 顧客の参加、協力が低い 見直しプロセスの手続きルール違反	⑬
		⑭ 開発環境の不備	⑭ C D	⑭	顧客交渉戦略、能力不足 大規模な仕様追加要求の対応 大規模な納期前追加要求の対応 顧客の参加、協力が低い 見直しプロセスの手続きルール違反	⑭
力	要因分類	⑮ 開発者不足	⑮ C D	⑮	顧客交渉戦略、能力不足 大規模な仕様追加要求の対応 大規模な納期前追加要求の対応 顧客の参加、協力が低い 見直しプロセスの手続きルール違反	⑮
		⑯ 開発者投入戦略の誤り (開発費投入時期ミス)	⑯ C D	⑯	顧客交渉戦略、能力不足 大規模な仕様追加要求の対応 大規模な納期前追加要求の対応 顧客の参加、協力が低い 見直しプロセスの手続きルール違反	⑯
情報	要因分類	⑰ あいまいな関係範囲 (スコープ)	⑰ C D	⑰	顧客交渉戦略、能力不足 大規模な仕様追加要求の対応 大規模な納期前追加要求の対応 顧客の参加、協力が低い 見直しプロセスの手続きルール違反	⑰
		⑱ あいまいな要求仕様	⑱ C D	⑱	顧客交渉戦略、能力不足 大規模な仕様追加要求の対応 大規模な納期前追加要求の対応 顧客の参加、協力が低い 見直しプロセスの手続きルール違反	⑱
情報	要因分類	⑲ 情報の不備・不足 (マネジメント情報、技術情報、過去の失敗情報)	⑲ C D	⑲	顧客交渉戦略、能力不足 大規模な仕様追加要求の対応 大規模な納期前追加要求の対応 顧客の参加、協力が低い 見直しプロセスの手続きルール違反	⑲
		⑳ 情報の不備・不足 (マネジメント情報、技術情報、過去の失敗情報)	⑳ C D	⑳	顧客交渉戦略、能力不足 大規模な仕様追加要求の対応 大規模な納期前追加要求の対応 顧客の参加、協力が低い 見直しプロセスの手続きルール違反	⑳



## 5. 設計工程のリスク管理表

: 第4章 4-1. ●開発現場にこそがるリスクの事例、表4-1-1

リスク管理表 (設計工程)	リスク名	更新日	初版発行日	文庫番号	担当名	リスク管理項目		リスク内容	リスク発生	対策日	備考
						リスク管理項目	対策日				
ヒトに関するリスク	①他者依存的姿勢 (自律性の放棄)	○	○	○	○	○	○	○	○	○	○
	②上位マネジメントの関与不足	○	○	○	○	○	○	○	○	○	○
	③ユーザーの参加・協力度不足	○	○	○	○	○	○	○	○	○	○
	④組織能力不足 (未熟な組織文化、戦略の欠如)	○	○	○	○	○	○	○	○	○	○
	⑤見聞力能力	○	○	○	○	○	○	○	○	○	○
	⑥要件定義能力	○	○	○	○	○	○	○	○	○	○
	⑦リーダーのプロジェクトマネジメント能力 (外部交渉、タイムマネジメント、現場主義、見える化能力など)	○	○	○	○	○	○	○	○	○	○
	⑧メンバーの技術能力、コミュニケーション能力	○	○	○	○	○	○	○	○	○	○
	⑨プロセス管理の精確	○	○	○	○	○	○	○	○	○	○
	⑩コミュニケーション能力 (阻害、キヤップ)	○	○	○	○	○	○	○	○	○	○
モノ	⑪関連要素との連携不足	○	○	○	○	○	○	○	○	○	○
	⑫ドキュメントの不備 (要件定義書・設計書・チェックリスト・手順書など)	○	○	○	○	○	○	○	○	○	○
	⑬関係のベネフィットの精確	○	○	○	○	○	○	○	○	○	○
	⑭開発環境の不備	○	○	○	○	○	○	○	○	○	○
	⑮開発者不足	○	○	○	○	○	○	○	○	○	○
カネ	⑯資源投入情報の誤り (開発費投入時期ミス)	○	○	○	○	○	○	○	○	○	○
	⑰あまりにも開発期間 (スコープ)	○	○	○	○	○	○	○	○	○	○
	⑱あまりにも要求仕様	○	○	○	○	○	○	○	○	○	○
	⑳情報の不備・不足 (マネジメント情報、技術情報、過去の失敗情報)	○	○	○	○	○	○	○	○	○	○

6. 課題管理表：第4章 4-3. ●課題管理表、表4-3-1

【課題管理表】		優先度	警告	発生日	期限	完了日	記入者	発工程	課題項目	課題内容	課題対応策	担当	発生作業/進捗状況/結果	備考
No	1	2	3	4	5	6	7							

7. 仕様変更管理表：第5章 5-1. ●仕様変更管理表がもたらす効果、表5-1-1

【仕様変更管理表】														
No.	管理番号	発生日	対応期限 (ベースライン)	対応日	対応状況	業務名	変更内容	変更区分	修正担当	影響度 範囲	難易度	リリース バージョン	対応工数 (見込)	対応工数 (実績)
1	PJ-001										A			
2											B			
3											C			
4														
5														
6														
7														
8														
9														
10														
	合計													

## チェックリスト一覧表

チェックリスト名	章・節の番号	記載ページ
◆プロジェクトマネジメントリスク一覧	1 - 3.	3
◆早期仕様凍結のチェックポイント	3 - 1.	13
◆仕様凍結のチェックポイント	3 - 1.	15
◆仕様調査のチェックポイント	3 - 1.	16
◆要求仕様書の詳細チェックリスト	3 - 1.	19
◆見積り回答書の形式的なチェックリスト	3 - 4.	28
◆見積り精度向上のチェックリスト	3 - 4.	29
◆プロジェクト体制構築の要件	3 - 5.	35
◆優れた開発チームの能力特徴	3 - 5.	36
◆開発体制不備のリスク	3 - 5.	36
◆開発の入り口で押さえるべきリスク	4 - 1.	43
◆開発中に押さえるべきリスク	4 - 1.	43
◆開発の出口で押さえるべきリスク	4 - 1.	43
◆現場におけるリスクの事例	4 - 1.	44

## 著者プロフィール

佐野洋（さのひろし）

現在、フリーコンサルタントとして PM ファクトリーを主宰。

数十年にわたり POS システムのファームウェア開発およびプロマネ業務に従事。

好きな言葉は「Boys be ambitious!」。

信条は「弱き者も強き者も共にその生涯を生き抜くこと」。

## コンタクト

eMail : pmf\_hsano@yahoo.co.jp

URL : <https://pmfactory-hsano.jimdofree.com/>

## お問い合わせ

本書に記載されている内容についてのみのお問い合わせとさせていただきます。

またお問い合わせにつきましては、eMail : pmf\_hsano@yahoo.co.jp 宛てにお願いいたします。

なお、ご質問の際には、書名、該当ページ、氏名、返信先を明記していただきますようお願いいたします。

お送りいただいたご質問には、可能な限りお答えできるように努力いたしますが、当方にて不適切だと判断されるご質問には回答を差し控させていただく場合もあります。あらかじめご了承のほどお願いいたします。

## 著作等

『SE 稼業は忘己利他（もうごりた）—現場に転がる箴言集』技術評論社webサイトにて連載

URL : <http://gihyo.jp/dev/serial/01/engineer-proverbs>

---

本書の無断複写複製（コピー等）は、  
著作者の権利侵害になります。



## CONTENTS

第1章 プロジェクトマネジメントにおける問題とリスク

第2章 事前準備工程のリスク

第3章 計画工程のリスク

第4章 実行工程のリスク

第5章 管理工程のリスク