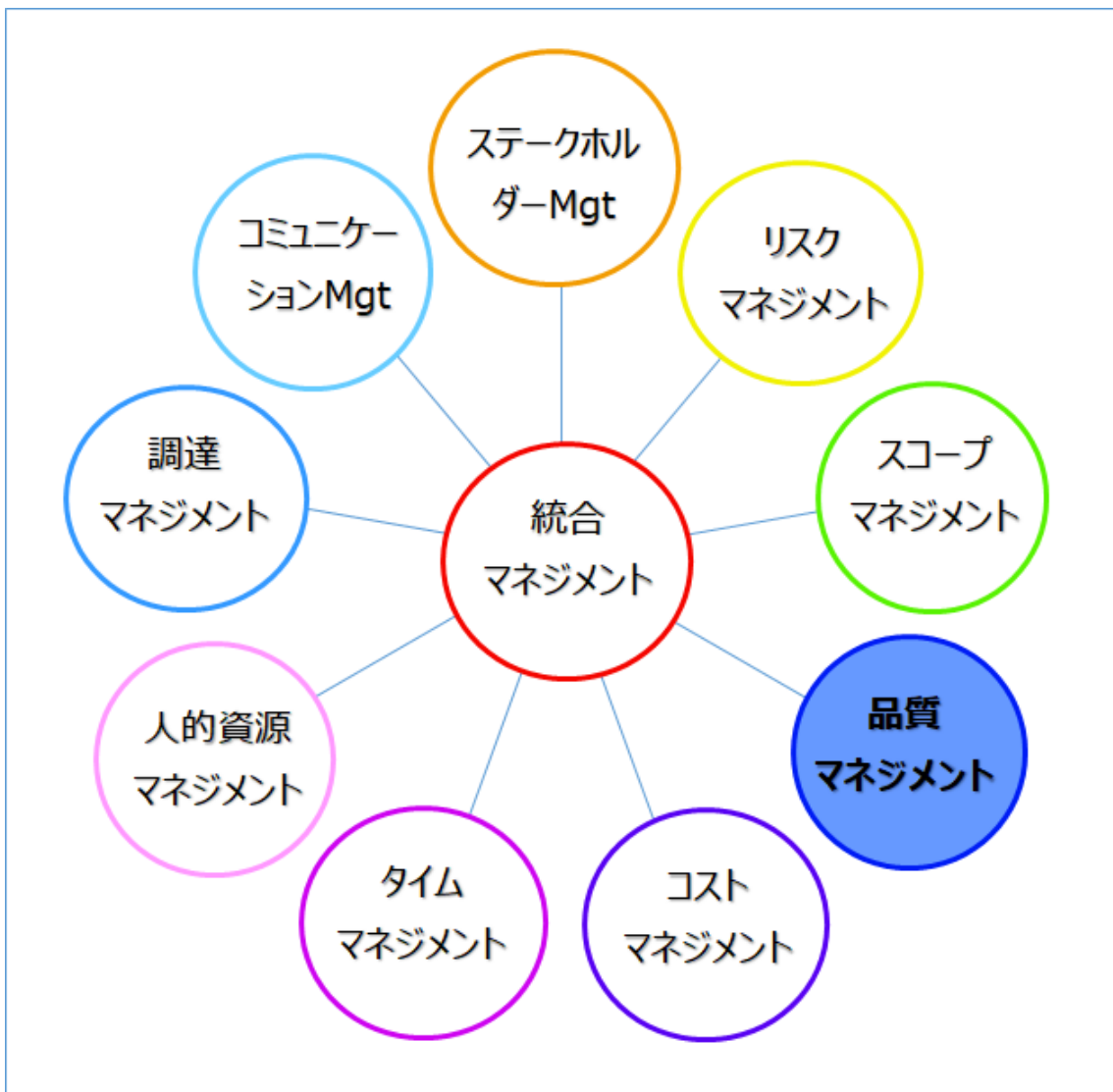


実用プロマネ わかる現場の

# 品質マネジメント





## 目次

### 第1章

#### 品質マネジメントの役割 p 1

- 1-1. 品質マネジメントの役割 p 1
- 1-2. プロジェクトの成功要件 p 2

### 第2章

#### 品質マネジメント 計画 p 5

- 2-1. 品質目標値の設定 p 5
  - プロジェクト目標の設定 p 5
  - 品質目標値の設定 p 6
    - 品質について p 6
    - 業務品質（ヒトの品質）の設定 p 6
    - 製品品質（モノの品質）の設定 p 7
    - その他の製品品質指標 p 8
    - 品質目標の達成に向けて p 9
  - 納期と品質 p 10
  - 目標値設定による期待効果 p 11
  - 目標をあきらめない p 12
- 2-2. ソフトウェア品質保証計画 p 13
  - プロジェクト計画書の作成 p 13
    - ◆プロジェクト計画書要件チェックリスト p 14

## 第3章

# 品質マネジメント 実行 p 15

- 3-1. 三つの時点で押さえるリスク p 15
- 3-2. QCD目標値の達成☆データドリブン開発の実行 p 16
  - データドリブン開発とは何か p 16
  - データドリブン開発の基本的なプロセス p 16
  - データドリブン開発を構成する三つの要素 p 17
  - データドリブン開発の効用 p 17
- 3-3. 品質目標値の達成☆ドキュメントベース開発の実行 p 19
  - 品質目標値の達成 p 19
    - 業務品質（ヒトの品質）の向上 p 20
    - 製品品質（モノの品質）の向上 p 20
  - ソフトウェア開発に必要なドキュメント p 21
  - 開発ドキュメントの現状 p 22
  - ドキュメントの不良が招いた失敗事例 p 23
  - ドキュメント精度の簡易チェック法 p 24
    - 基本設計書の簡易チェック法 p 24
    - 詳細設計書の簡易チェック法 p 24
  - 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 25
  - 開発管理表の概要 p 26
    - プロセス管理表 p 26
    - リスク管理表 p 26
    - 課題管理表 p 27
- 3-4. QCDには優先順位はつけられない p 28
- 3-5. 改善活動の実行 p 29
  - 改善活動の意義 p 29
  - 仕事に対する認識 p 30
  - 改善活動の基本コンセプト p 30
  - 改善活動の事例 p 31
  - 改善活動計画書 p 32
  - 改善活動を阻害するもの p 34

## 第4章

### 品質マネジメント 管理 p 3 5

#### 4-1. 品質管理 p 3 5

品質管理に使用される管理表の例 p 3 6

品質状況管理表 p 3 6

不具合習熟曲線表 p 3 6

レビューの品質と実行 p 3 8

◆全レビュー共通の効果的レビューのポイント p 3 8

◆設計レビューのポイント p 3 9

効果的なレビューを実行するためのドキュメントの条件 p 3 9

外注納品物の品質管理 p 4 1

書類審査 p 4 1

受入テスト審査 p 4 2

## 第5章

### 品質マネジメント 振り返り p 4 3

#### 5-1. 品質管理の考察（プロジェクト完了報告書） p 4 3

#### 5-2. 失敗の原因を他に求めないこと p 4 4

できにくい反省 p 4 4

振り返り p 4 4

ラップアップ（開発完了会議） p 4 5

付録表 p 4 6

著者プロフィール等 巻末



# 第1章

## 品質マネジメントの役割

### 1-1. 品質マネジメントの役割

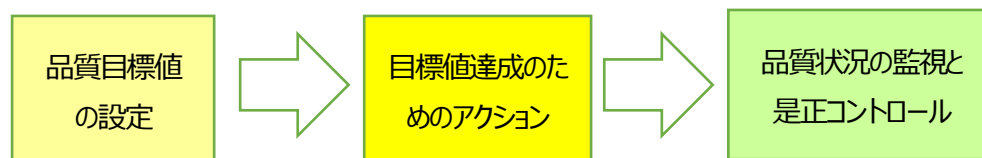


**品質マネジメント<sup>1</sup>**：作業と成果物の品質目標を達成するようにマネジメントする

#### ◎ 品質マネジメントの役割

業務品質および製品品質を数値化し、達成すべき品質の目標値を設定し、品質目標を達成するための改善活動を伴った開発業務を遂行すること。

【品質マネジメント】（図1-1-1）



<sup>1</sup> **品質マネジメント** 業務品質および製品品質を数値化し、達成すべき品質の目標値を設定し、品質目標を達成するための改善活動を伴った開発業務を遂行すること。

## 1 - 2. プロジェクトの成功要件

一般的にソフトウェア開発プロジェクトの成功要件は、次の三つの目標の達成だと言われています。

### 【プロジェクトの成功要件】

1. 品質目標 Q : Quality
2. コスト目標 C : Cost ⇒利益目標
3. 納期目標 D : Delivery ⇒生産性目標

コスト目標は同時に利益目標となり、また納期目標は同時に生産性目標となります。

これらのQ C Dの目標は、三つが同時に達成されて初めて成功プロジェクトだと言えます。

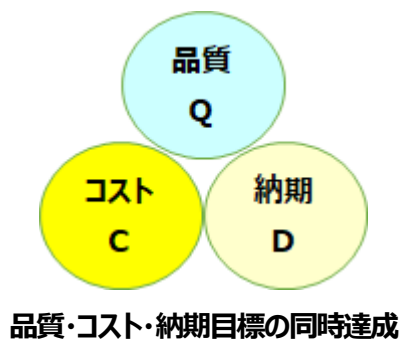
例えば下記の三つのケースにおける目標の達成はほとんど意味がないということが分かります。

- ① 品質目標は達成したがコスト・納期目標は未達成の場合
- ② コスト目標は達成したが品質・納期目標は未達成の場合
- ③ 納期目標は達成したが品質・コスト目標は未達成の場合

コスト目標は、開発組織内においては利益目標として表現されることもあります。

Q C D目標の達成は顧客と開発組織との間の契約によって約束されるもので、どれか一つでも未達成ならば契約違反となり、民事的に公的な処罰の対象となることもあります。

### 【プロジェクト成功の3要件】 (図1 - 2 - 1)



<sup>2</sup> 生産性 時間当りに生み出される成果物の量および質を表わす指標。



プロジェクトを成功に導くためにはプロマネは下記の実行が必要になります。

### 【プロジェクトを成功させるポイント】

- ① 妥当な見積りによる妥当な開発期間および開発工数の確保を行うこと。
- ② 要求仕様の早期凍結を行うこと。
- ③ リスクの早期解消を行うこと。
- ④ リーダーが開発仕様の全体像を押さえておくこと。
- ⑤ 開発メンバーに対して適切な仕事の配分を行い、十分に仕様の理解をさせておくこと。
- ⑥ 開発効率化・失敗防止のための継続的な改善活動<sup>3</sup>を実行すること。

効率的開発、良い品質、納期の確保、利益の創出を実現するためには上記の活動が必要になり、これらの活動を主導するのがプロマネの中心的な役割だと言えます。

◎プロジェクトの成功要件は、目標Q C Dの三つを同時に達成すること。

PM

<sup>3</sup> 改善活動 Q C Dに関する問題の改善を図りプロジェクトを成功に導くと同時に、人材の育成を図る活動のこと。特に失敗に学ぶ活動は有効な結果を生む。



## 第2章

# 品質マネジメント 計画

## 2-1. 品質目標値の設定

### プロジェクト目標の設定

プロジェクト目標の代表的なものとして、Q C Dの目標すなわち品質目標、コスト・利益目標、納期・生産性目標の三つをあげることができます。

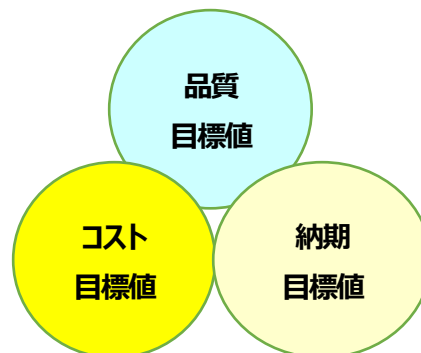
プロジェクトのQ C D目標値は、見積り回答書にて示した数値内で設定することが絶対的な条件となります。次に、所属する上位組織のQ C D目標に沿ったものが要求されますが、いずれにしても、その組織ないしはプロジェクトが持っている実力値以上の目標を達成することは極めて困難です。組織などの目標値の設定は、例えば三か年計画などに基づいて徐々に引き上げられていくことが現実的であり、達成の可能性も高くなります。そのような意味で、プロジェクトの目標の設定にあたっては、最初に開発組織のQ C Dの実力値を知るところから始める必要があります。

またQ C Dの各目標は、同時に達成される必要があり、どれか一つないしは二つが達成されたとしても他の目標が達成されなかった場合は、成功プロジェクトとは言えません。

計画

◎ Q C Dの目標値は顧客および社内経営層に約束したものであり、  
どれか一つでも未達成ならば失敗プロジェクトとなる。

【Q C D目標値の設定】 (図2-1-1)



スローガンではなく、目標数値の設定を！

## 品質目標値の設定



### 品質について

品質には**製品品質**<sup>4</sup>と**業務品質**<sup>5</sup>の二つがあります。製品品質とはモノ自体の品質であり、業務品質とはそのモノを作成するヒトの業務行為の品質のことです。

製品品質（モノの品質）は、業務品質（ヒトの品質）によって規定されてしまいます。ヒトがモノ（ソフトウェア）を作っている限りにおいては、業務品質レベル以上の製品品質の結果は生まれません。

計画

- ① 品質には業務品質と製品品質がある。
- ② 業務品質が上がらなければ製品品質はあがらない。



### 業務品質（ヒトの品質）の設定

業務品質とは、ヒトの業務における行動の切り口から見た品質であり、プロジェクトチームの開発能力のことを意味します。端的に言えばヒトの能力であり、具体的業務能力としてはプロジェクトマネジメント能力、プロセス遂行能力、技術能力、生産性レベルなどがあります。

業務品質の主な指標例としては以下のものがあります。

#### 【業務品質の指標例】

- ① 仕様品質：仕様ミス件数／率（／Kstep）、深刻度A B C分布、変更影響度表作成率
- ② 設計品質：設計ミス件数／率（／Kstep）、深刻度A B C分布
- ③ 製造品質：製造ミス件数／率（／Kstep）、深刻度A B C分布
- ④ 評価品質：評価ミス件数／率（／Kstep）、深刻度A B C分布、バグ発見数／率
- ⑤ **構成管理**<sup>6</sup>品質：構成管理ミス件数／率（／Kstep）、深刻度A B C分布
- ⑥ プロセス管理品質：プロセス管理表実施率
- ⑦ リスク管理品質：リスク対策件数／率（／Kstep）、深刻度A B C分布、リスク管理表実施率
- ⑧ 課題管理品質：課題対策件数／率（／Kstep）、深刻度A B C分布、課題管理表実施率
- ⑨ 各レビュー品質：レビュー指摘件数／率（／Kstep）、深刻度A B C分布、レビュー実施率

<sup>4</sup> **製品品質** 物自体の品質。例：ソフトウェアの不具合発生率等。

<sup>5</sup> **業務品質** 人間が行う業務能力の品質。例：作業ミス率等。

<sup>6</sup> **構成管理** ソフトウェア開発におけるソースコードや設計書などの成果物に対して変更履歴を管理し、必要に応じたバージョンの成果物を取り出すことを可能にする。一般的に構成管理ツールが用いられ、ビルド管理、リリース管理、バージョン管理などを通して成果物の一貫性を保つようになっている。

深刻度ABC分析<sup>7</sup>は、障害内容を顧客の視点から見て、その深刻度のレベルを高いものの順に、A：重度障害、B：中度障害、C：軽度障害と定義し、その発生件数・率などによって品質の傾向を把握するものです。

計画

◎ヒトの業務品質を向上させるためには、継続的な業務改善活動を！



### 製品品質（モノの品質）の設定

製品品質とは、モノ（ソフトウェア）の切り口から見た品質であり、開発能力（業務品質）以上の製品品質が生まれることはありません。製品品質の主な指標例としては以下のものがあります。

#### 【製品品質の指標例】

- ・ 単体テスト品質：不具合件数／率（／Kstep）、深刻度ABC分布
- ・ 結合テスト品質：不具合件数／率（／Kstep）、深刻度ABC分布
- ・ 総合テスト品質：不具合件数／率（／Kstep）、深刻度ABC分布
- ・ 市場品質：市場障害件数／障害率、深刻度ABC分布

製品品質における深刻度ABCの例を以下に示します。

A 重度障害の例：全システムダウン、会計数値違算、大規模な取引・発注・在庫管理機能不正等

B 中度障害の例：一部システムダウン、一部端末のロック、一部取引・発注・在庫管理機能不正等

C 軽度障害の例：会計数値に影響しないレベルの不適切または誤った印字・表示・操作シーケンス等

計画

◎製品品質は、不具合発生件数・密度および深刻度の3点により評価すること。

#### 【2つの品質で支えるシステム】（図2-1-2）



<sup>7</sup> ABC分析 障害内容を顧客の視点から見て、その深刻度のレベルを高いものの順に、A：重度障害、B：中度障害、C：軽度障害と定義し、その発生件数・率などによって品質の傾向を把握するためのもの。



### その他の製品品質指標

製品品質については、ソフトウェアの信頼性の視点を中心に示してきましたが、バグが出ないだけでソフトウェアの品質が高いとは言えません。ハードウェアの世界において知られている優れた製品の5つの特性についても、指標化および目標値の設定が必要になります。本来、総合テストはこれらの5つの特性検証の場であって、単なるバグ潰しの最終工程ではないのです。

#### 【優れた製品の5つの特性：RASUI<sup>8</sup>】

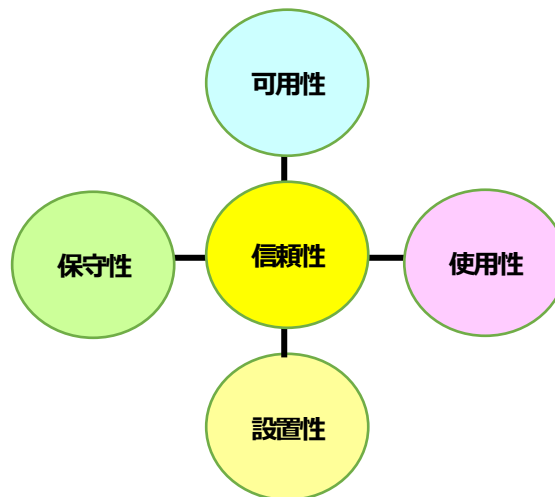
- ・ リアビリティ Reliability：信頼性（不具合・障害率の低さ）
- ・ アベイラビリティ Availability：可用性（継続的稼働能力）
- ・ サービスビリティ Serviceability：保守性（保守・メンテのしやすさ）
- ・ ユーザビリティ Usability：使用性（使いやすさ）
- ・ インストーラビリティ Installability：設置の容易性

計画

#### ◎優れた製品の5つの特性：

- ①信頼性、②可用性、③保守性、④使用性、⑤設置容易性

【優れた製品の5つの特性】（図2-1-3）



<sup>8</sup> **RASUI** 優れた製品の5つの特性である、Reliability：信頼性（不具合・障害率の低さ）・Availability：可用性（継続的稼働能力）・Serviceability：保守性（保守・メンテのしやすさ）・Usability：使用性（使いやすさ）・Installability：設置の容易性、の略称。



## 品質目標の達成に向けて

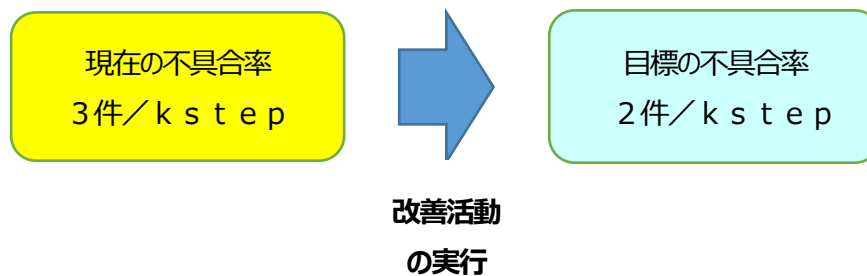
いずれの指標もソフトウェア業界全体を網羅できるような統一的基準値で表すことは困難であり、同一組織における過去・現在の実績値の変動の比較によって目標値を設定するのが現実的です。

業務品質を上げるためには、プロジェクトマネジメント能力の向上、プロセス遂行能力の向上、技術能力の向上、生産性の向上などが必要になりますが、このようなヒト依存的な能力の向上を実現するためには、上記に示した品質の達成目標を目指した改善活動を、個人の自主性にまかせるのではなく、組織的な取り組みとして実際のプロジェクトの中で実行する必要があります。

計画

◎ 品質の目標値は過去・現在の実績値の平均を見て設定すること。

### 【不具合率の削減に向けての活動】 (図 2 - 1 - 4)



## 納期と品質

納期の目標とは、単に納期の期日を守るということではないと言うことは誰でも分かっていることですが、切羽詰まって時間切れになってしまったプロジェクトが、多くのバグを内在したままソフトウェアをリリースしてしまうことが少なからずあります。

納期が顧客にとって本当の価値を持つ大前提は、リリースされる製品が顧客の要求を満たしていると同時に品質が確保されていると言うことです。言い換えれば、納期の目標とは、納期の期限内に品質を確保するための目標を設定することだと言えます。このことを実現するためには開発の各工程においても同様のことが実現される必要があります。すなわち次のようなことです。

- ・ 要件定義の納期までに、品質が確保された要件定義を実行しなければならない。
- ・ 基本設計の納期までに、品質が確保された基本設計を実行しなければならない。
- ・ 詳細設計の納期までに、品質が確保された詳細設計を実行しなければならない。
- ・ コード製造の納期までに、品質が確保されたコーディングを実行しなければならない。
- ・ 総合テストの納期までに、品質が確保された総合テストを実行しなければならない。

◎納期は品質が確保されなければ意味がない。

計画



## 目標値設定による期待効果

目標値の設定は、データに基づいた開発（データドリブン開発<sup>9</sup>）がプロジェクト遂行の第一歩と言え、ソフトウェア開発を科学的・工学的に遂行する基本的な手段となります。

数値に基づく開発は、プロジェクトの代表的な問題である、あいまいさや情緒的な判断を一掃する役割を果たします。

目標値の設定は次のような効果を生み出します。

### 【目標値設定による期待効果】

- ・ 現状の開発力が数値として明確になる。
- ・ 改善すべき問題を数値的に把握できる。
- ・ 改善結果が数値的に明らかになる。
- ・ 到達すべきゴールが数値的に見え、モチベーションの維持につながる。
- ・ あいまいさや情緒性を排し、合理的ないしは妥当性のあるチームを育成する。

計画

◎ 目標は数値として表すことで、実行可能な目標となる。

<sup>9</sup> データドリブン開発 データを根拠にした開発活動のこと。

## 目標をあきらめない

仕事の目標は品質・コスト・納期の達成であると言われていたにもかかわらず、現実にはコスト・納期の限界で品質を犠牲にしている場合が少なからずあります。コスト・納期の限界で多くの開発工程が時間切れとなり、それぞれの仕事在中途半端な未完成の状態での次の工程に回され、最後に市場において多くの障害を生んでいるのです。これは仕方ないといっておきましても良い問題ではありません。中途半端な仕事でコストや納期を守ったつもりでも、結果として障害対応によってさらに多くのコストや時間を浪費することになり、最も大切な顧客の信用を失う結果を招いてしまいます。

なぜコスト不足・時間不足になっているのか真剣に考える必要があります。多くの問題は、要件定義や見積りなど開発の前工程に存在していることは誰でも気づいていることですが、多くの人はその改善に着手しようとはしません。改善活動は仕事には含まれていないと考えている開発者やマネージャが意外に多いのです。改善活動こそ仕事の中核業務であるという意識を持つ必要があります。

失敗プロジェクトの一番の問題は、妥当なコストや納期条件での受注ができていないこと、二番目の問題は明確な要求仕様が提示されないこと、三番目の問題は合理的なプロジェクトマネジメント能力および技術力の不足にあります。この問題を解決するためには現在までの自分たちの失敗を直視し、その真因を把握し、その改善目標を設定し、改善活動を実行・継続する必要があります。

目標の達成をあきらめたらその時点で失敗が確定してしまうことになります。

### 【失敗プロジェクトの原因ワースト3】（図2-1-5）

◎ 妥当なコストや納期条件で受注ができていない

◎ 明確な要求仕様が提示されない

◎ プロジェクトマネジメント能力および技術力の不足

改善活動  
を始めよう

計画

◎ 目標は、あきらめなければ必ず達成可能である。

## 2-2. ソフトウェア品質保証計画

### プロジェクト計画書の作成

プロジェクト計画書<sup>10</sup>はプロジェクト計画プロセスの総まとめであり、開発実行の行動計画書となるもので、プロジェクトをスムーズに運営し、QCD目標を達成するために必要な活動内容を計画書という形で明文化したものです。品質保証に関しては、そのガイドライン、レビュー計画、テスト計画およびテスト範囲についての記述が必要です。またプロジェクト計画書には、見積り内容、プロジェクトのスコップ、スケジュール、コスト、リスク、ステークホルダーとのコミュニケーション方法などの記述が含まれます。

プロジェクト計画書の目次の例を以下に示します。

#### 【プロジェクト計画書】

##### 1. 概要

- (1) プロジェクト計画書の目的
- (2) プロジェクト概要（開発目的、スコップ、納品物、リリース基準、開発プロセス、プロジェクト環境）
- (3) 見積り内容（見積り結果等）

##### 2. ソフトウェア開発計画

- (1) プロジェクトチーム（開発体制、役割と責任）
- (2) リソース計画（プロジェクトに必要なリソース、トレーニング計画等）
- (3) 日程計画（マイルストーン）
- (4) プロジェクト進捗管理（小日程表、進捗報告、定例会議、レビュー、顧客への報告等）
- (5) リスク管理計画

##### 3. ソフトウェア品質保証計画（ガイドライン、レビュー計画、テスト計画、テスト範囲等）

##### 4. ソフトウェア構成管理（SCM）計画

- (1) 構成の確定（管理対象、管理文書、管理ツール等）
- (2) 構成管理のプロセス
- (3) 構成管理状態の把握（ステータス報告、状況把握の頻度、報告書の配布）
- (4) 構成管理ステータス監査（監査方法、監査の頻度）

##### 5. 外部委託管理計画

- (1) 外部委託選定理由及び見積り
- (2) 外部委託計画内容

##### 6. システム導入計画（システム導入日程等）

##### 7. その他の情報（製品情報、書類番号等）

<sup>10</sup> プロジェクト計画書 開発実行の行動計画書となるもので、プロジェクトをスムーズに運営し、QCD目標を達成するために必要な活動内容を計画書という形で明文化したものです。プロジェクト計画書には、見積り内容、プロジェクトのスコップ、スケジュール、コスト、リスク、ステークホルダーとのコミュニケーション方法などの記載が必要。

プロジェクト計画書作成にあたって下記の要件をチェックすることで漏れを防ぐことができます。

◆【プロジェクト計画書要件チェックリスト】

- 開発目的は明確になっているか。
- プロジェクトの開発範囲は明確になっているか。
- 納品物は明確になっているか。
- 納品物の検収基準（リリース基準）は明確になっているか。
- 開発プロセスは明確になっているか。
- プロジェクト環境（開発環境・テスト環境・実稼働環境）が明確になっているか。
- 参考にした過去のプロジェクトのデータが明確になっているか。
- 見積結果が添付されているか。
- 商品企画書、開発指示書が添付されているか。
- 開発体制及び役割と責任は明確になっているか。
- 顧客から提供されるリソースは明確になっているか。
- プロジェクトに必要なリソースは明確になっているか。
- 他の組織から提供されるリソースは明確になっているか。
- 日程計画（スケジュール表、プロセス管理表）は立てられているか。
- 進捗管理方法（進捗管理表、定例会議、レビュー、進捗報告）は明確になっているか。
- リスク計画は立てられているか。
- ソフトウェア品質保証計画は立てられているか。
- ソフトウェア構成管理計画は立てられているか。
- 外注管理計画（外注委託選定理由、見積り、計画内容）は立てられているか。
- システム導入計画は立てられているか。

上記に示されたようにプロジェクト計画書は、プロジェクトの全体像をその根拠と共に表したもので、社内経営層の承認のもとに、顧客をはじめとした重要なステークホルダーと共有されるべき資料です。

## 第3章

# 品質マネジメント 実行

### 3-1. 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。とりあえず開発工程を前・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。
- QCDの数値目標の設定を行うこと。



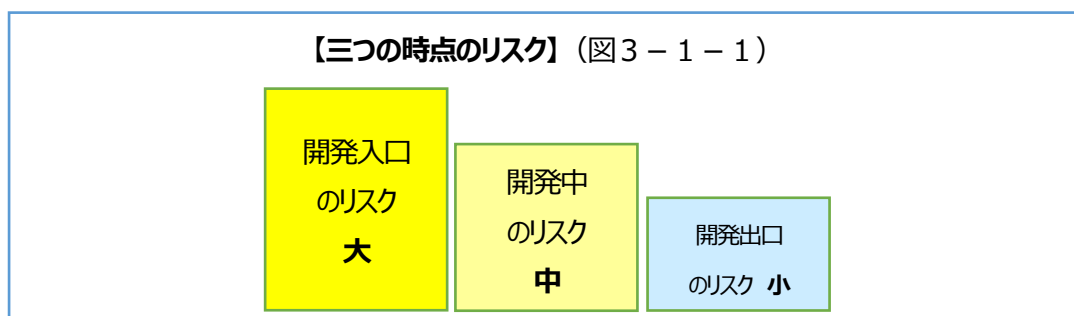
#### ◆ 開発中に押さえるべきリスク

- 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>11</sup>の実行)

- QCD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。



<sup>11</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。

## 3-2. QCD目標値の達成☆データドリブン開発の実行

### データドリブン開発とは何か

データドリブン開発とは、開発の推進にあたってデータを根拠にした活動を押し進めるという考え方です。

データを根拠にするとは、事実を証明する数値に基づいた資料や論拠を使うということです。具体的には開発における目標の設定、問題の分析、および開発行為のすべてに必要なコミュニケーションや、要件定義書・設計書などのドキュメントの記述内容は、全て数値・数字の裏付けが必要だということです。

難しく複雑な開発という仕事を無事に完了させるためには、感情論を排し、合理的な方法が必要になります。合理的な仕事のやり方の基本が、データに基づくやり方およびドキュメントに基づいたやり方です。数値で示されるものでなければ、目標も結果も適切に判断することはできません。

データに基づいて開発を遂行するということは、自分ないしは自チームの過去のQCD等のデータおよびそれらから導き出された目標値データに基づいて仕事をするということです。データに基づかない仕事のやり方は、出たとこ勝負のやり方でプロの仕事とは言えません。

実行

◎ 基本的なデータはQCDのデータ。

### データドリブン開発の基本的なプロセス

開発の基本的な数値はQCDに関する数値です。開発チームにおける現在のQCDの実力値を知り、目標とするQCDの値を設定し、プロジェクト終了時に達成されたQCDの数値の振り返りを行うことがデータドリブン開発の基本的なプロセスだと言えます。

プロジェクトを開始する前に、過去のプロジェクトにおける品質に関するデータを収集しておく必要があります。この過去のデータの平均値が現時点における開発チームの実力値を表わしています。これまでの失敗の反省に基づき、改善対策を実行した結果、得られる期待成果<sup>12</sup>としての数値がプロジェクトの目標値となります。この目標値はプロジェクトの終了時に得られた実績数値との比較を行うことで、次のプロジェクトにおける目標指標を算出することができます。

実行

◎ 過去のデータ、目標とするデータ、結果のデータの三つでドライブする。

<sup>12</sup> 期待成果 達成したいと考えている目標数値。

## データドリブン開発を構成する三つの要素

データに基づく開発は、次の三つの開発**コンセプト**<sup>13</sup>において推進されることで、Q C D 目標値の実現を確かなものにします。

- |                                      |                  |
|--------------------------------------|------------------|
| 1. <b>ドキュメントベース開発</b> <sup>14</sup>  | Q : 品質目標値の達成     |
| 2. <b>プロフィットドリブン開発</b> <sup>15</sup> | C : 利益・コスト目標値の達成 |
| 3. <b>優先順位ベース開発</b> <sup>16</sup>    | D : 納期・生産性目標値の達成 |

それぞれの開発コンセプトの実行はQ C Dのすべてに好影響を与えますが、上記はQ C Dのどれに強力に効果をあらわすのかということを表わしたものです。

## データドリブン開発の効用

データに基づいた開発を言うことは、最短時間・最小エネルギーで物事を成し遂げる最善の方法として、合理的なやり方を採用するということです。合理性とは、物事の筋道を数値によって明らかにし、その数値の意味するところによって現状を把握し、それに従った行動を行うということです。日常業務の多くの物事は数値によって表現が可能であり、合理性の効用として以下のものが考えられます。

### 【合理性の効用】

- ① 種々に異なる問題の解釈を、だれの誤解も招かない一つの形式で表すことができる。
- ② 問題の量や質など、度量衡に関する全てのことを具体的かつ明確に示すことができる。
- ③ 数値化された原因や結果は、衆人の検証に耐えうる証拠となり得る。
- ④ 数値化することで見えない事象、すなわち見えない原因や結果を見えるようにできる。
- ⑤ 最短の時間と最小のエネルギーで問題を解決できる。

多くの問題は、物事を数値・数理で表すことによって解決可能です。特に開発上の問題は、品質・コスト・時間に関するデータや数値の把握によらなければ絶対に解決することはできません。数値でとらえるべき問題を情緒的な判断でとらえようとしても適切な解は得られず、時間とエネルギーの無駄使いとなるだけです。

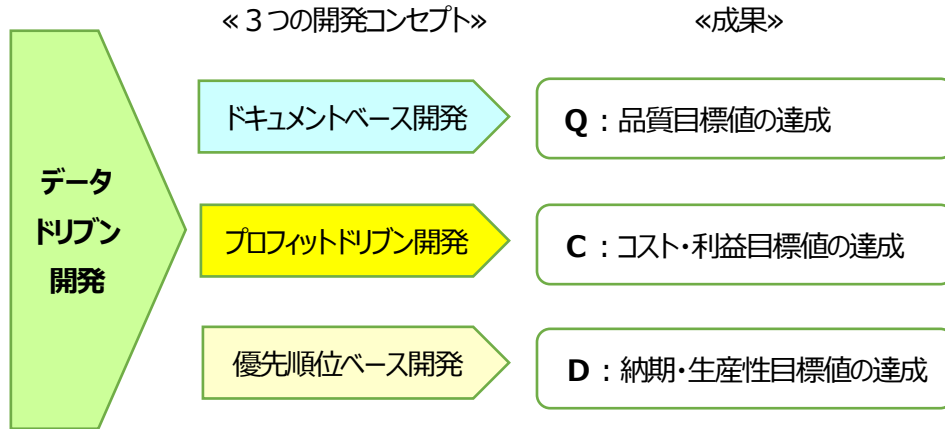
<sup>13</sup> **コンセプト** ものごとの基本的な概念を言い表したもの。ものごとの骨格を表わす考え方。

<sup>14</sup> **ドキュメントベース開発** 口頭だけに依存せず、書類・書面に基づいて実行される開発。

<sup>15</sup> **プロフィットドリブン開発** 利益目標の達成を主眼として、同時にコスト目標の達成を実現する開発。

<sup>16</sup> **優先順位ベース開発** 顧客価値の高い要求仕様の開発を優先させ、不要不急な開発を避けること。

【データドリブン開発】（図3-2-1）





### 3-3. 品質目標値の達成☆ドキュメントベース開発の実行

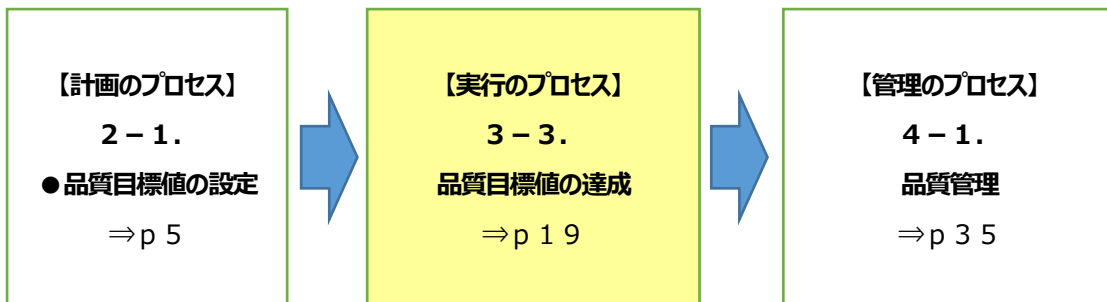
ドキュメントベース開発とは、開発業務のあらゆる面において、必ず何らかの書類・書面に基づいた情報の伝達や情報の共有を実行するという考え方のことを指しています。

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。

品質目標値の達成は、主にこのドキュメントベース開発の実行および後述の「3-5. 改善活動の実行」によって実現されます。

#### 品質目標値の達成

2-1. において、品質目標値の設定について解説を行いました。本章においてはその目標値達成のために必要なプラクティスについての解説を行います。品質に関する解説の流れは下図のようになっています。



どのような目標値であれ、それを達成するために必要なことは以下の3点だけです。

#### ◎品質目標値達成のための3原則

実行

- ① 現在の品質状態を数値で表すこと。
- ② 改善の目標値を決めること。
- ③ 目標値の達成に必要な改善活動を実行すること。



## 業務品質（ヒトの品質）の向上

プロジェクトの業務能力を高めるためには、プロジェクトの構成員である開発者個人の業務能力を高める必要があります。そのためには個々の開発者における現在の業務能力のレベルを把握し、その改善目標値を設定し、必要な改善活動を実行する必要があります。

ヒトの業務品質を数値で把握するために適した指標としては、下記の二つが一般的なものです。

- ①業務実行ミスの件数および率
- ②業務内容の実行件数および率

第2章で取り上げた業務品質の指標例は次の通りでした。

- ①仕様品質、②設計品質、③製造品質、④評価品質、⑤構成管理品質、⑥プロセス管理品質、⑦リスク管理品質、⑧課題管理品質、⑨各レビュー品質

これらの指標値は、チーム全体の値および個人のデータの両方を収集する必要があり、それぞれの弱点を明確にし、それらを改善するための対策を組織的な改善活動として実行する必要があります。改善活動を個人レベルに任せていては、ほとんど改善は実行されないと思ったほうが良いでしょう。

また個人のデータに関しては「個人スキル管理票」などとしてまとめておけば、個人のスキル育成計画やキャリア形成計画の資料として活用できます。いずれの業務品質の指標も、ソフトウェア業界全体を網羅できるような統一的基準値は公表されていませんので、自組織における過去・現在の実績値の変動の比較によって判断する必要があります。



## 製品品質（モノの品質）の向上

チームおよび個人の業務品質が向上してくれば当然のことに製品の品質も向上してきます。

製品の品質は、ヒトの業務品質が製品の上に現れてきたもので、品質状況を端的に表すのに適した指標だと言えます。

また品質の改善対策を検討する場合、業務品質の指標と製品品質の指標の両面から問題の真因を追求すれば、より具体的な対策案にたどり着くことができます。

例えば、製品品質の単体テスト品質に問題があった場合、単に単体テストのやり方だけに原因を求めるのではなく、関連する業務品質である仕様品質・設計品質・製造品質も含めた原因追及が、真因をつきとめるのに役に立ち、適切な改善活動につながることができます。

第2章で取り上げた製品品質の指標例は次の通りでした。

- ①単体テスト品質、②結合テスト品質、③総合テスト品質、④市場品質

実行

◎すべての品質指標は改善活動のテーマであり、  
業務品質も製品品質も、組織的な改善活動を実行しなければ上がることはない。

## 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）、どういう風にす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、ソフト構成図、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

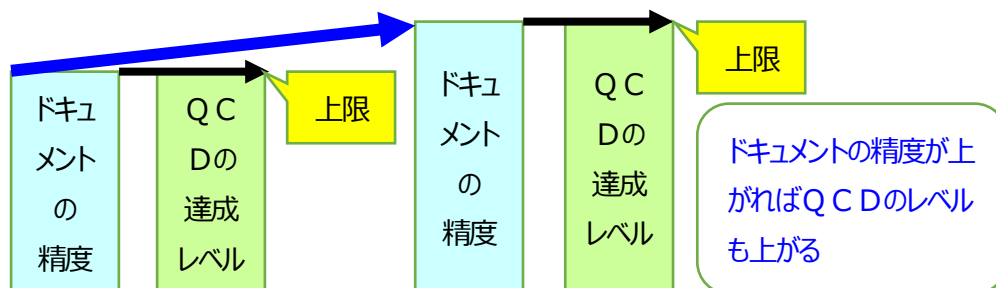
貧弱なドキュメントがどのようにひどい状況をもたらすのかを知るために、IPA・SEC<sup>17</sup>が公開しているプロジェクトの失敗事例の中からドキュメントの不備による失敗事例を紹介します。

- 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QCDの達成レベル】（図3-3-1）



<sup>17</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。

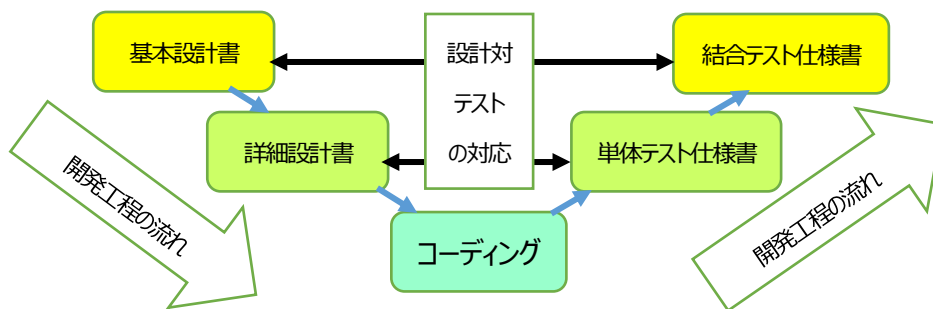


### 詳細設計書の簡易チェック法

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3-3-2）



## 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

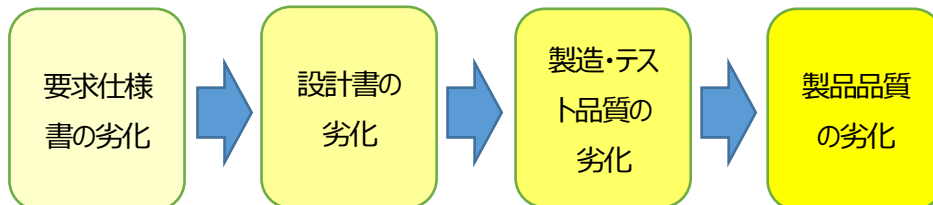
黙っていてもドキュメントは更新されません。更新されていないドキュメントの内容は何が「正」で、何が「誤」なのかを判断することができません。段々とメンテされなくなったドキュメントは、ついには誰にも使用されない大量の紙クズとなってしまいます。開発ドキュメントは常にその鮮度を保っておく必要があり、生鮮食品の一種だと思った方がよいのです。常に鮮度を保つ管理をすることが、不要な労力を省き、スムーズな開発を実現することにつながります。

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3-3-3）



## 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 3 - 3 - 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（参照：付録表 3 - 3 - 1）

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。

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（参照：付録表 3 - 3 - 2 ~ 3 - 3 - 4）

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。





### 課題管理表（参照：付録表 3 - 3 - 5）

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 3 - 3 - 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													

### 3-4. 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

### 3-5. 改善活動の実行

改善活動と言えば、大方の人たちは時間の余裕がある時に仕事のついでに行うQCサークル活動のようなものだと思います。時間がないから、予算がないから改善活動ができないという声をあちこちで聞きます。では、時間やお金があったら本当に改善活動ができるのでしょうか。そのような人たちに時間と金を与えても、結局形ばかりの活動になり、何らの実利も得られないでしょう。成長の意欲のある人は、他人から言われずとも知恵を出して、時間を生み出し、利益を生み出すための改善活動をすでに実行しています。成長意欲のない人には改善の意義も理解できず、改善の動機も湧かないでしょう。

そういうわけで、時間がない、資金もない、要求仕様も二転三転して定まらないというような、愚痴ばかりを並べたてるばかりで、一向にそれらの改善活動も実行せず、仕事の内容は手抜きや丸投げが蔓延するばかりで、製品は市場でバグだらけで、損益も万年赤字続きが続いているのです。

#### 改善活動の意義

このような荒廃した環境を作るのも復興させるのも人間、とくに組織をリードする人たちです。プロマネの役割と責任を自覚する必要があるでしょう。個人および組織に、実利の獲得と人間的成長を同時にもたらす運動の一丁目一番地は、コミュニケーションの活性化を基盤とした改善活動に他なりません。改善活動は開発業務の中核業務であり、暇な時についでにやるようなものではないのです。

改善活動がもたらす成果には次のようなものがあります。

#### 【改善活動がもたらす実り】

1. QCD数値の大幅な改善
2. チームメンバーの技術能力の向上
3. チームリーダーのプロマネ能力の向上
4. コミュニケーションの活性化による組織能力の向上
5. 全員の自律（自立）能力およびモチベーションの向上
6. 会社の成長・発展

改善活動の苦痛を嫌って現在の地獄的な状況を我慢し続けるのか、一時の新たな負担を背負う覚悟を決めて改善活動の実りを手に入れるのかを決めるのは、みなさんの選択次第なのです。

実行

◎改善活動は、開発業務の中核業務であり、暇な時についでにやるようなものではない。

## 仕事に対する認識

ルーチンワークをこなしているだけでは、品質の高い業務や製品を生み出すことはできません。ルーチンワークと合わせて、自他の過去の失敗に学ぶ改善活動の実行が、高品質な業務および製品を生み出すことはトヨタのカイゼン活動で実証済みであり、世界中に知られていることです。

改善活動を行っていない開発組織は、開発という創造的な仕事から、いつの間にかルーチンワークという単純労働集約型の仕事へと劣化していきます。良い仕事を継続的に実行していくためには、改善活動を中核にすえた仕事を日常的に実行する必要があります。

◎ **良い仕事 = 改善活動 + ルーチンワーク**

**開発業務 = 創造的な仕事 + ルーチンワーク**

実行

## 改善活動の基本コンセプト

改善活動の基本コンセプトは以下の通りです。

### 【改善活動の基本コンセプト】

- ① **個人戦**<sup>18</sup>ではなく**組織戦**<sup>19</sup>（チームプレー）を行うこと。
- ② 一人だけが成長するのではなく、みな共に成長すること。
- ③ 能力に長けたものは、後進の者にその知恵（ノウハウ）を譲ること。
- ④ 後進の者もその成長に従って、さらに後に続く後進の者に、その知恵（ノウハウ）を譲ること。
- ⑤ 先に進んでいる者は、その持てるあらゆる価値あるもの、資産・資金・知恵（ノウハウ）の全てを後進のものに順に譲り渡し、永続的な繁栄の循環を実現させること。

◎ **改善活動の基本コンセプト**

**共に成長し、持てる智・財を継承し、永続的な繁栄の循環を実現すること。**

実行

<sup>18</sup> **個人戦** 組織に拠ることなく、個人単独で問題解決に当たること。

<sup>19</sup> **組織戦** 複数の人間で組織を形成し、それぞれに役割分担を決め、協調・連携のもとに問題解決に当たること。

## 改善活動の事例

改善活動の具体的な例としては以下のものがあります。

### 【改善活動の事例】

- |                  |                            |
|------------------|----------------------------|
| 1. コミュニケーションの活性化 | 1 2. 評価仕様書の精度向上            |
| 2. 開発プロセスの確立と励行  | 1 3. 無駄の排除                 |
| 3. プロジェクトの数値目標設定 | 1 4. 開発の効率化                |
| 4. 要求仕様書の精度向上    | 1 5. レビュー品質問題の改善           |
| 5. 要求仕様の早期凍結化    | 1 6. プロジェクトの振り返り（ラップアップ）励行 |
| 6. 要求仕様の変更管理     | 1 7. 開発ノウハウの継承             |
| 7. 仕様変更影響度表の作成   | 1 8. リーダーシップとチームプレー問題の改善   |
| 8. 見積り手法問題の改善    | 1 9. モチベーションの喚起            |
| 9. 見積りリスクの排除     | 2 0. 手抜き問題の撲滅              |
| 1 0. 開発リスクの排除    | 2 1. 組織風土文化の改善             |
| 1 1. 設計書の精度向上    |                            |

いずれも身近にある問題ばかりです。これらの問題を全てとは言わないまでも10%、いや30%程度改善するだけで、どれ程の成果がプロジェクトのみならず会社にもたらされるのかは容易に想像できるでしょう。

◎改善活動の具体的なテーマは、  
繰り返される失敗や、いつもリスク管理表に取り上げられる項目の中にある。

## 改善活動計画書

改善活動計画書の基本的な記述内容を下記に示します。

この計画書はいわゆる **DMAIC**<sup>20</sup>手法に準じたもので、DMAICの意味は次の通りです。

- D (Define : 問題点記述)
- M (Measure : 現状の指標値データ)
- A (Analyze : 現状のデータ分析)
- I (Improve : 解決策)
- C (Control : 成果の維持・定着方法)

### 【改善活動計画書フォーム】

1. プロジェクトテーマ名
2. プロジェクト体制 : リーダー名記述、メンバー名記述
3. プロジェクト期間 : 開始日、終了予定日
4. 問題点記述 (Define) : \* 解決すべき課題を現在の状況と共に複数記述する。
5. 顧客に提供される価値 (顧客のCTQ Critical To Quality)  
\* この改善によって顧客に提供される価値 (QCD等) を記述する。
6. プロジェクトの成果目標
  - ・ 目標の記述 \* 何をどのようにするかを記述する。
  - ・ 成果目標値の記述 \* 改善活動によるQCD等の成果期待値・金額等を記述する。
7. 現状の指標 (値) データ (Measure)
  - ・ 収集した現状の問題等を整理・分類し、データ化する。改善目標について現状の数値を示す。
8. 現状のデータ分析 (Analyze)
  - ・ フィッシュボーンによる分析 ; 問題を構成している要因分析を行い、主要要因をあぶり出す。
  - ・ データに基づく分析 ; 問題を構成している複数の要因を数値分析し、改善ターゲットを特定する。
9. 解決策 (Improve)
  - ・ 複数の解決策を立案し、実行の優先順位をつける。
  - ・ 実行の優先順位は、QCD改善効果 (費用対効果) + 実行難易度 + 緊急度により決定する。
10. 成果の達成見込み (\* 改善実行後には実績値に書き換える)
  - ・ 成果物、改善金額、投資コスト、成果金額 (= 改善金額 - 投資コスト) 等。
11. 成果の維持・定着方法 (Control)
  - ・ 実行された解決策が将来に渡って組織に定着する方法・手段を記述する。

---

<sup>20</sup> **DMAIC** Define (定義)、Measure (測定)、Analyze (分析)、Improve (解決策)、Control (管理) のステップからなる経営変革手法であり、VOC (Voice of Customer、顧客の声) を基にして事業活動を分析し、データドリブンでプロセスの改善を進める。

下記は改善計画書簡易版のサンプルです。

【DMAICシート簡易版】(図3-5-1)

承認 印		<b>DMAICシート(簡易版)</b>	報告年月日 : PJリーダー名 : PJメンバー名 :				
<b>テーマ：仕損費コストの削減</b>							
<b>Define</b> *問題の明確化		<b>Analyze</b> *問題を派生している根因の追及					
<p>有るべき姿・こうありたいと思う状態</p> <ol style="list-style-type: none"> <li>ソフトウェア製品における市場品質の向上</li> <li>上流工程での品質確保</li> </ol>		<ol style="list-style-type: none"> <li>注文書発行の遅延等で、本来のタイミングでの公式レビューが実施されておらず、問題点が発見されても後戻り出来ない工程になっている事が多い。又、短時間の公式レビューでは、開発プロセス及びリスクに関する細かなレビューが出来ない為、公式レビューを補完する他のレビューが必要。</li> <li>委託外注からの成果物受入検査を、総合テストで行っている為、総合テスト時に単体テスト不足による不具合が多発。又、外注より成果物を受入れる為の品質基準が不明確(数値基準が無い)であり、受入時のレビューが不足。</li> </ol>					
<p>現在の状態</p> <p>XX年下期仕損費 : xxxx千円</p>							
<p>問題点 (PJが取り組む課題・テーマ)</p> <ol style="list-style-type: none"> <li>デザインレビューだけでは上流工程での品質の確保は難しい。(プロセス及びリスクに関するレビューが不足。)</li> <li>単純な不具合(プログラム製造段階での不具合)が相変わらず多い。</li> </ol>							
<b>Measure</b> *問題の分解と優先順位		<b>Improve</b>					
<p>一次分解</p> <p style="text-align: center;">レビューが不足      製造不具合が多い</p>		<p>改善策</p> <ol style="list-style-type: none"> <li>第三者によるプロセス監査・リスク監査の継続的实施。</li> <li>外注受入の為の品質基準の作成及び、品質基準に基づく外注受入検査・レビューの実施。</li> </ol>					
<p>二次分解</p> <table border="1" style="width: 100%; text-align: center;"> <tr> <td style="width: 25%;">                 リスクの抽出が充分出来ない。 (1)             </td> <td style="width: 25%;">                 開発プロセスの問題抽出が充分出来ない。 (2)             </td> <td style="width: 25%;">                 請負外注先での単体テスト不足。 (3)             </td> <td style="width: 25%;">                 外注受入検査が充分出来ない。 (4)             </td> </tr> </table>		リスクの抽出が充分出来ない。 (1)	開発プロセスの問題抽出が充分出来ない。 (2)	請負外注先での単体テスト不足。 (3)	外注受入検査が充分出来ない。 (4)	<p>改善効果金額(年間) : H/S : xxxx千円</p>	
リスクの抽出が充分出来ない。 (1)	開発プロセスの問題抽出が充分出来ない。 (2)	請負外注先での単体テスト不足。 (3)	外注受入検査が充分出来ない。 (4)				
		<b>Control</b> *改善を定着化するために行ったこと					
		<p>プロセス監査・リスク監査を公式レビューと同様に開発プロセスの一部として定義付け、CMMにおけるSQA活動の一貫として監査を実施していく。</p>					
*解決効果の大きい順に(1)~(4)							

## 改善活動を阻害するもの

改善活動を阻害する要因として、心理的なものと環境的なものの二つがあります。



### 心理的な要因

心理的な要因としては、たとえば改善活動によって将来大きな利益が得られると言われても、その失敗リスクを考えると現状のままの方がまだ安全だという心理が働き、たとえ、品質が悪い・コストが高い・生産性が悪いという状況であったとしても、何とかやっていくことが出来ているという、低レベルではあるが、それなりに安定している現状を失いたくないという心理が働きます。要するに、悪いなりに安定している現状を失いたくないために、将来利益を生み出すかも知れない改善活動を避けようとしているものと考えられます。また現状のQCDのレベルはたとえ余り良くないとは思っていても、顧客からの苦情も一過性で済んでおり、何年もこのような状況に慣れてしまっているため、改善などの手間ひまがかかる活動を開始する動機がわからない、という心理的な傾向の影響もあるでしょう。

**【改善活動の阻害要因】 1. 現状を替えたくない心理 2. 費用・時間の不足**



### 環境的な要因

社内の改善活動にとっての最初の障害は費用および時間の不足にあります。出たとこ勝負的な開発を行っているような組織は、慢性的な赤字と時間不足に悩まされています。仕事に投入される時間は、妥当な品質を確保するために必要な時間の60%以下なのかも知れません。現実的な問題として、このようなチームに自分たちの時間の5%を割いて改善活動を開始するように説いたところで、所詮そんな時間はどこにもないから、やはり改善活動はできないということになります。

このようなチームを復活させるためには、そのチームの予算外で動ける人材の投入が必要になるということを最初に理解しておく必要があります。チーム内に改善活動をリードする適任者がいたならば、その人材の予算を別途確保し、そのチームには背負わせないような手当が必要となります。

経験から言えることは、改善活動によってある程度の実利が得られる見通しさえついていれば、このような初期投資の金額の数倍もの改善益はすぐにでも得られますが、ないない尽くしの中で改善活動ができないのは当然のことであるとも言えます。

実行

◎それでも改善活動には、初期投資のヒト・カネ・時間が必要になる。



## 第4章

# 品質マネジメント 管理

### 4-1. 品質管理

品質管理の対象になる品質指標はすでに示した通りですが、これらの目標値を達成するためには当然のことその裏付けとなる改善活動が必要になってきます。改善活動を行うためには、その指標の数値について、現在の数値の把握、改善目標数値の設定および改善結果数値の把握およびコントロールが必要になってきます。例えば製品品質の代表的な指標である単体テスト・結合テスト・総合テストにおける不具合密度の例を次に示します。

**【各テストの不具合密度の現在値・目標値・改善実績値】** 不具合密度：（件数/kstep）

	現在値	目標値	実績値
単体テスト	5	3.5	3
結合テスト	3	2	2.5
総合テスト	2	1.4	1.6

各テストにおける不具合密度について現在値を把握した上で、何らかの改善施策により30%の改善目標を立てた場合には、上記のような数値目標の設定となります。各テストの実行時に、実績数値を把握することで達成率を把握し、改善施策の不足部や欠点を特定し、次なる改善活動の目標とします。

改善施策としては、設計書の精度向上・評価仕様書の精度向上・仕様変更影響度表の作成などが有効でしょう。

## 品質管理に使用される管理表の例



### 品質状況管理表

品質状況管理表は、単体テスト・結合テスト・総合テストにおける不具合発生件数・原因・**バグ密度**<sup>21</sup>・深刻度レベルなどの把握に使用されます。

【品質状況管理表】（表 4 - 1 - 1）

【品質状況管理表】													
工程	業務名	プログラム ステップ数	テスト項目数	不具合原因内訳				不具合件数 合計	バグ密度 (件数/kstep)	不具合深刻度レベル			テスト工数
				仕様バグ	設計バグ	製造バグ	管理ミス			A (軽度)	B (中度)	C (軽度)	
単体テスト	A												
	B												
	C												
	D												
単体テスト 合計													
結合テスト	A												
	B												
	C												
	D												
結合テスト 合計													
総合テスト	A												
	B												
	C												
	D												
総合テスト 合計													

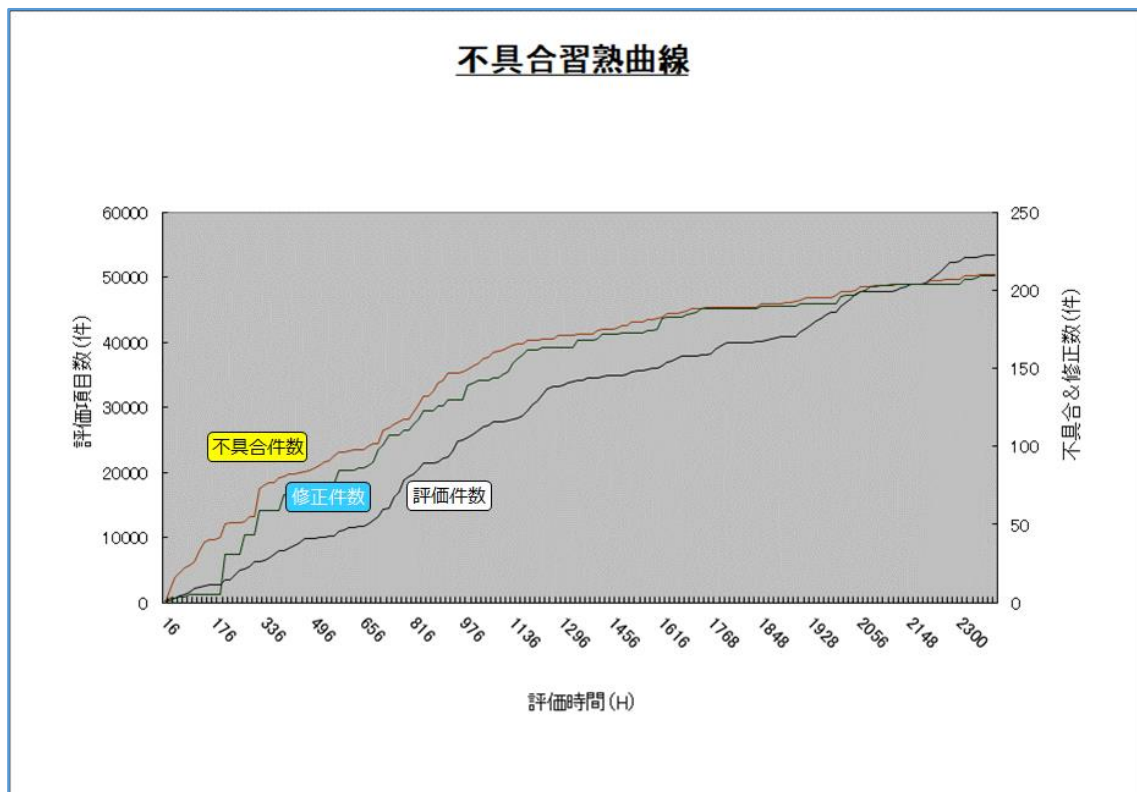
<sup>21</sup> **バグ密度** 作成されたソフトウェアの 1 k s t e p 当りに発生したバグの数で表される。



## 不具合習熟曲線表

不具合習熟曲線表<sup>22</sup>は、信頼度成長曲線またはゴンペルツ曲線などと呼ばれるもので、横軸にテスト時間、縦軸に累積バグ発見数・修正件数・評価件数をとったグラフで表されます。S字の成長習熟曲線を描くことが多く、決定論的モデルとして、現在の状況から今後の予想を立て、テスト進捗管理、バグ収束率の予測、残バグ数の予測などに用いられます。

【不具合習熟曲線表】(図4-1-1)



<sup>22</sup> **不具合習熟曲線表** 信頼度成長曲線またはゴンペルツ曲線などと呼ばれるもので、横軸にテスト時間、縦軸に累積バグ発見数をとったグラフで表される。S字の成長習熟曲線を描くことが多く、決定論的モデルとして、現在の状況から今後の予想を立て、テスト進捗管理、バグ収束率の予測、残バグ数の予測などに用いられる。

## レビューの品質と実行

品質のモニタリングおよびコントロールにおいて、品質管理表と同時に大切なのが、主要な成果物に対する**レビュー**<sup>23</sup>です。レビューの目的はそれぞれの成果物の品質評価による誤りの排除にあります。

レビューのポイントおよび設計レビューにおける主なチェック内容を示します。

### ◆【全レビュー共通の効果的レビューのポイント】

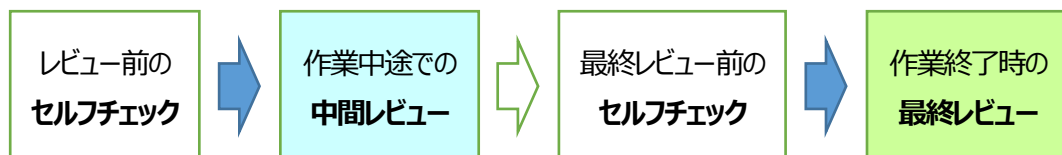
- あらかじめレビュー計画を立て、時間を確保しておき、レビューのポイントを考えておく。
- レビュー対象物が一目で分かるような一覧表を作成しモレをなくす。
- 共通レビュー項目の雛形を作成しておく。また既存のレビュー資料で流用できるものは流用する。
- 各担当者のスキルに合わせたレビューのチェックポイントおよびレビュー内容を考える。
- セルフチェックの時間を設け、レビューしきれない詳細部分については、事前にチェックを済ませておく。
- セルフチェックおよびレビューに要した工数実績データを記録・累積しておき、セルフチェックやレビューにかかるべき妥当な時間を割り出し、効果的なセルフチェック資料を作成する。
- 中間レビューおよび本レビューの二段階レビューを行う。中間レビューでは基本的な理解や方向性の間違いをチェックし、本レビューでは重要機能やリスク機能に絞ったレビューを行う。  
何から何まで全てをレビューすることは不可能であるということを認識しておく。
- 影響度の確認レビューを行う。
- ミスや不具合の傾向をデータに基づいて分析・把握し、関連する重度障害や再発の多い不具合を重点的にレビューする。

管理

### ◎効果的なレビューのポイント

事前の準備、事前のセルフチェック、二段階レビュー、影響度の確認、過去の失敗

### 【レビューの基本的な流れ】（図4-1-2）



<sup>23</sup> レビュー 工程毎の成果物に対して、その成果物が意図された通りに作成されたかどうかを検証し誤りを発見する行為のこと。

#### ◆【設計レビューのポイント】

- 前提として、設計は重要な仕様順に実行すること。
- 事前に説明する設計骨子を整理しておくこと。
- 事前に自己レビューを行うこと。
  - 詳細にわたるレビューは自己レビューの段階で済ませておき、単純なミスはこの段階で排除しておく。
  - 頻度の高い設計ミスなどを網羅した、自己レビュー用の基本的なチェックシートを用意しておく。
- 設計骨子の重要な順に、理解した要求仕様内容および実現方法としての設計内容を説明する。
- 中間レビューと本レビューの二段階レビューを行う。
  - 中間レビューにおいて、重要仕様の理解が過不足なく設計に反映されているかどうかをチェックする。
  - 本レビューにおいて、重要仕様の最終チェックと同時に、時間の許す限り他の仕様のチェックを行う。
- レビュー内容のポイント
  - 不明仕様を自分の想定によって設計している部分はないか。想定部は速やかに確定させる。
  - 要求仕様の意味を正しく理解しているか。
  - 入力条件は正しいか、処理のロジックは正しいか、出力は正しいか。
  - 異常処理は適切か。
  - レスポンス・パフォーマンスの考慮は適切か。
- 疑問点・不明点は都度レビュアーに判断を仰ぐこと。



#### 効果的なレビューを実現するためのドキュメントの条件

レビューがうまく行かない原因として、ドキュメントの内容が貧弱であることおよび各ドキュメント間の仕様のひもづけがされていないことがあります。主要なドキュメントである、要求仕様書・基本設計書・詳細設計書・テスト設計書においては同じ仕様について要求仕様番号と同様な番号を付番することで一貫通貫的にひもづけを行い、要求仕様書と同様な構造（WBS）を持つような記述をする必要があります。

すべてのドキュメントが、同じ思想のもとで同じ構造化が行われていれば、どの工程の担当者においても仕様の理解が容易になると同時に、ドキュメントのメンテナンスも容易になり、当然のことに誤記・記入抜け・バラバラな記述・分かりにくい記述の発見も容易になり、それぞれのレビューもスムーズになります。

このように全てのドキュメントが共通の構造のもとに統一されていれば、作成されるプログラムも自ずと構造化されやすくなり、プログラム作成の精度が向上すると同時にメンテナンスも容易になるはずです。

◎ドキュメントの条件

要求仕様書・基本設計書・詳細設計書・テスト設計書は、共通なWB S 構造を持つこと。

【共通なWB S 構造のイメージ】（表4-1-2）

レベル1：システムA						
	レベル2		レベル3		レベル4	
	項番	成果物	項番	成果物	項番	成果物
要求仕様書	1	要求仕様書	R 1 - 1	業務アプリ1	R 1 - 1 - 1 R 1 - 1 - 2 ...	業務1-1. 要求仕様 業務1-2. 要求仕様 ...
基本設計書	1	基本設計書	B 1 - 1	業務アプリ1	B 1 - 1 - 1 B 1 - 1 - 2 ...	業務1-1. 基本設計 業務1-2. 基本設計 ...
詳細設計書	1	詳細設計書	D 1 - 1	業務アプリ1	D 1 - 1 - 1 D 1 - 1 - 2 ...	業務1-1. 詳細設計 業務1-2. 詳細設計 ...
テスト設計書	1	テスト設計書	T 1 - 1	業務アプリ1	T 1 - 1 - 1 T 1 - 1 - 2 ...	業務1-1. テスト設計 業務1-2. テスト設計 ...

要求仕様書からテスト設計書に至るまで、同じ仕様や機能に関するもの同士は、同様の項番を附番することによって、次のことが可能になる。

- ドキュメントの構造化
- ドキュメント相互の関係性を容易に追跡可能にする。
- 仕様の矛盾やモレなどの発見を容易にする。

## 外注納品物の品質管理

ソフトウェア開発の元請ベンダーにおいては、近年ますます外注ソフト開発会社に対する依存度が高まって来ており、それらの外注会社から納品されるソフトウェアの品質確保が重要な課題になっています。

外注化は、主に設計・製造・評価テストなどの工程において進んでいますが、本章では外注会社の請負範囲を設計・製造および結合テストまでとした、ソフトウェアの受入れ検査について解説をします。

外注化されるものに対する品質基準は、社内における品質基準と同等レベルのものを設定する必要があり、それらは「2-1. 品質目標値の設定」において記載した品質指標と同等なものです。

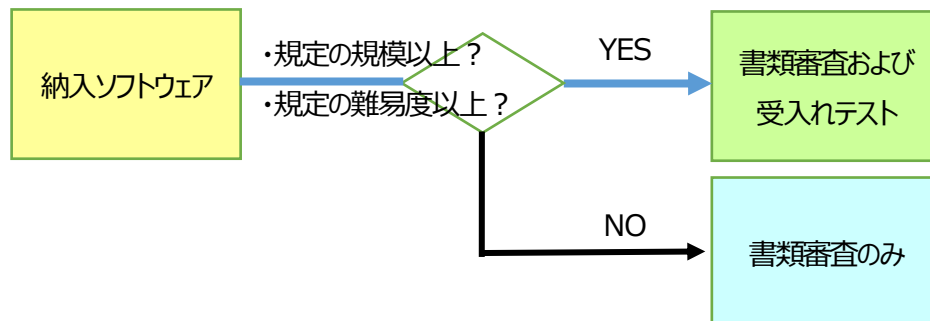
外注から納品されるソフトウェアは、「ソフトウェア受入検査基準管理表」などの書面により、納入されるソフトウェアの品質が、要求された基準を満たしていることが証明される必要があります。

納入ソフトウェアの受入判定は、書類審査と受入テストの二つの方法で行われますが、開発規模の小さなものおよび難易度が低いと判断されるものは、規定に従って書類審査のみの場合もあります。

管理

◎外注に委託した開発作業の品質基準は、社内基準と同等のものを適用すること。

【納入ソフトウェアの受入れ判定】（図4-1-3）



## 書類審査

外注会社は、以下の資料の提出が求められます。

### 1. ソフトウェア受入検査基準管理表

単体テストおよび結合テスト、それぞれのテスト件数、テスト時間、テスト密度、テスト不具合密度を記入し、受入基準を満たしていることを証明するもの。

### 2. テスト確認表（単体テスト・結合テスト）

### 3. テスト仕様書（テストチェックリスト、単体テスト・結合テスト）

### 4. テスト不具合一覧表（単体テスト・結合テスト）

### 5. テスト進捗表（単体テスト・結合テスト）

### 6. 詳細設計書

### 7. ソースコード

受入検査基準管理表のサンプルは次の通りです。

【受入検査基準管理表】（表4-1-3）

テスト密度・不具合密度	合格基準値	単体テスト	結合テスト
テスト密度 (テスト件数/kstep)	〇〇件以上		
不具合密度 (不具合件数/kstep)	〇〇件以下		
テスト件数			
テスト時間			
<b>付帯納入成果物の確認</b>	受領日		
テスト確認表			
テスト仕様書			
テスト不具合一覧表			
テスト進捗表			
詳細設計書			
ソースコード			

## 受入テスト審査

所定の開発規模以上ないしは難易度レベル以上のものは、書類審査に合格した上で、受入テストによる審査を必要とします。受入テストの実施要領は次のようになります。

重要仕様部分の動作確認テストを行い、受入検査基準管理書に記載された不具合密度の妥当性を検証し、納入された付帯成果物における不具合の再発や類似不具合の発生がないか等の確認を行う。

提出されたエビデンス<sup>24</sup>と相違ない結果で新たな不具合が発見されなければ合格とされます。

<sup>24</sup> エビデンス 証拠または根拠を示すもの。



# 第5章

## 品質マネジメント 振り返り

### 5-1. 品質管理の考察（プロジェクト完了報告書）

プロジェクト完了時に実行される振り返り会議（ラップアップ・ミーティング）においては、仕様変更管理、品質管理、コスト管理、進捗管理それぞれの目標と実績の差の理由および今後の課題を総括し、次に続く開発への申し送り情報としてプロジェクト完了報告書にまとめます。

 品質管理の考察（表5-1-1）

品質管理の考察			不具合原因内訳				実績値		目標値	不具合深刻度レベル			テスト工数
テストフェーズ	プログラムステップ数	テスト項目数	仕様バグ	設計バグ	製造バグ	管理ミス	不具合件数	バグ密度 (件数/kstep)	バグ密度 (件数/kstep)	A (重度)	B (中度)	C (軽度)	
単体テスト													
結合テスト													
総合テスト													
市場障害													
合計													
考察	1. バグ密度の実績値と目標値の差異の理由および今後の改善課題 2. 不具合原因内訳・不具合深刻度レベルの結果についての反省 3. 品質管理全般についての反省												

品質管理においては、プロジェクトが目標としたバグ密度と実績の差異を把握し、目標未達の場合は何が不足していたのかを考察し、新たな改善施策を立て、次の開発において実行する必要があります。また不具合原因の内訳や不具合の深刻度レベルを分析し、チームの弱点をしっかりと認識し、これらについても改善の実行が必要になります。

## 5 - 2. 失敗の原因を他に求めないこと



### できにくい反省

失敗の結果に対する人の反応として3種類があります。1 つめは何も悔やまず平気な人。2 つめは悔やむだけの人。3 つめは悔やんだ後で、同じ失敗をしないために失敗の原因を探し対策を考える人。

失敗しても自分のせいにはしたくないと思えば真剣な反省もできません。反省したり謝罪したりすることは本当に難しいことです。言い訳から始まる謝罪は顧客や相手の怒りを倍増させます。

失敗の原因を追究しない限り、その失敗を生み出している状況や現実の変化を読み取ることはできません。現実直視力や柔軟性は失敗に学ぶことによって養成されていきます。

自分の失敗責任は最小化し、他人の失敗責任を最大化するような人は、自己の成長もリスクの回避もできないでしょう。失敗の現実を直視できない人のリスクは高くなります。失敗の原因を他者や物に求める人のリスクは更に高くなります。



### 振り返り

レビューが振り返り行為の一つだという認識のない人においては、忙しいという理由で、しばしばレビューがスキップされてしまうことが往々にしてあります。

レビューも振り返り会議も実行されなければ、失敗の経験に学ぶ機会が全くないということになります。そのようなプロジェクトは規模の大小を問わず、手抜きプロジェクトなのです。日常的にこのようなやり方を続けていけば品質も利益も悪化し、人材も全く成長できません。このような**学習能力**<sup>25</sup>のない組織に所属していたとしても、自分が気づいた時点でプロジェクトリーダーやマネージャに問題提起をする必要があります。開発の規模にかかわらず、すべてのプロジェクトにおいて、自己レビューや社内レビューおよび振り返り会議等を必ず実行する必要があります。

「振り返り」を通じた「改善活動」を日常的に実行することで、技術者個人のスキル向上や組織力の向上が確実に実現され、バグに追い回されることもなく、確実な利益を確保できる開発組織を生み出すことができるでしょう。

---

<sup>25</sup> **学習能力** 失敗に学ぶ能力ないしは成長したいと思う意欲のこと。



### ラップアップ<sup>®</sup>（開発完了会議）

ラップアップはプロジェクトを振り返る重要な最後のけじめであると同時に、次なる仕事へのノウハウの継承という重要な役割を担う行為であるともいえます。この仕事はプロマネの重要な責務の一つであり、メンバーの自主性に任せておくような性質のものではありません。プロマネ主導で必ず実行する必要があります。またラップアップは、業務遂行中に採取し分析を済ませておいたQ C Dの指標数値に基づいた振り返りの実行、および次に続くプロジェクトへの申し渡し事項をまとめておく必要があります。単に、〇〇の失敗をしないように“がんばろー”と言うような情緒的な会議にすることだけは避けたいものです。

振返

◎ 振り返りは、失敗の経験に学び個人・組織ともに成長する有効な手段であり、プロジェクトのけじめであると同時にノウハウ継承の役割をもっている。

付録表

3-3-1. プロセス管理表

【プロセス管理表】(サンプル)			*PL (or PM)、SPL (or SPM)		担当	PL	実施	レビュー		
NO	手順	進捗	作業	担当者	成果物	チェック	チェック	予定日	実施日	レビュー結果
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 / 要求仕様の承認					
	4 不具合箇止め策の選定、決定		開発に対する有効な箇止め策を	PL, SPL, 開発メン	・箇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	担当者	・仕様書各種					
	2 初期設計		仕様書レビュー 初期設計	PL, SPL, 開発メン 各担当	・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		初期設計レビュー、スケジュール SE確認：要求事項の明確化 チェックリスト確認	PL, SPL 開発部→関係各位	・レビュー報告書 / 変更モジュール一覧表 ・仕様書 ・設計書					
	4 詳細設計		詳細設計	各担当	・詳細設計書 / リリース影響度表					
	5 部材の確認		詳細設計レビュー 部材の変更	SPL、開発	・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 ・ステータス別チェックリスト / 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 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 ドキュメント		評価レビュー ・仕様書フィードバック ・設計書フィードバック	SPL、開発 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 SPL、評価	・チェックリスト ・レビュー報告書					
			評価レビュー 客先検証	評価担当、品証担当 PL、SPL、評価 開発部→SE、客先	・チェックリスト更新 / 試験結果 ・レビュー報告書 ・客先検証報告受領					
	11 デザインレビュー③		要求事項のチェックリスト確認	品質保証部→関係部署	・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		実験店舗稼働確認チェック 実験店全ログチェック 実験店全ログチェック結果レ	SPL、開発 SPL、開発 PL, SPL	・実験店舗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 ドキュメント整理		全国展開 ドキュメント整理、	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、箇止めの成果確認	PL, SPL	・プロセスの測定、分析、改善結果					
			開発予算との整合性チェック	PL, SPL	・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						



3-3-3. 要件定義工程のリスク管理表

リスク管理表 (要件定義工程)	プロジェクト名:	更新日:	初版発行日:	文書番号:	担当姓名:	リスク管理項目	対策日	備考
要因分類	Q	C	D	要因分類	要件定義工程リスク	リスク内容	管理番号	備考
①他者依存性姿勢 (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	○	○	○	○	・顧客交渉時期、能力不足 ・顧客の仕様追加要求の対応力 ・大規模な開発期間延長の対応力 ・大規模な納期短縮要求の対応力 ・顧客の参加・協力度が低い ・身振り手振りの手続遅れ・混乱			
③ユーザーの参加・協力度不足 (突然な組織文化、戦略の変化)	○	○	○	○	・関係者選定時の可否判断 ・顧客要件の発見 ・アカウントも、メンバー間能力			
④見積り能力	○	○	○	○	・要件定義能力不足 ・顧客交渉時期・能力不足 ・大規模な仕様追加要求 ・大規模な納期短縮要求 ・適切な交渉依頼依頼能力不足 ・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)			
⑤要件定義能力 リーダーのロジックの外マネジメント能力 (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)	○	○	○	○	・要件定義・検閲・知識不足の単員で要件定義を実施 ・要件定義基礎のため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			
⑥メンバーの技術能力、ローマシラー (ソフトウェア)	○	○	○	○	・メンバーの管理能力不足 ・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欠如	○	○	○	○	・プロセス管理の不足 ・仕様変更・確認の非設計計画 ・基本設計の凍結後、詳細設計着手 ・顧客のパートナー化失敗			
⑧コミュニケーション能力 (顧客・キヤパ)	○	○	○	○	・顧客とのコミュニケーション能力 ・プロセス上のコミュニケーション能力 ・営業と開発部門間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 ・要求仕様書なし			
⑨開発チームとの連携不足 ⑩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)	○	○	○	○				
⑪開発のペースの管理	○	○	○	○				
⑫開発環境の不備	○	○	○	○				
⑬開発費不足	○	○	○	○				
⑭資源投入戦略の誤り (開発費投入時期ミス)	○	○	○	○				
⑮あいまいな開発範囲 (スコープ)	○	○	○	○	・あいまいなスコープ ・あいまいな要件 ・あいまいな仕様			
⑯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			

### 3-3-4. 設計工程のリスク管理表

リスク管理表 (設計工程)	初版発行日	更新日	担当名	文書番号	文書種別	管理番号	リスク内容	対策アクション	対策日	備考
リスク管理表 (設計工程)										
原因分類	QC/D	設計工程リスク		√						振り分け
①作者依存性姿勢 (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コーディネータの参加・協力不足	○									
④組織能力不足 (未熟な組織文化、経験の欠如)	○									
⑤品質管理能力	○	・アクション開発								
⑥要件定義能力	○									
⑦リーダーのプロフェッ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 (ラッパミス)	○	・フレームワーク構築能力 ・異常系技術力の不足 ・タイムリミット設計能力の不足 ・非同期制御設計能力の不足 ・リアルタイム設計能力の不足								
⑨プロセス管理の有無	○	・ログ技術力の認識不足 ・アクション対応技術力不足 ・メモリアル対応技術力不足 ・性能 (パフォーマンス/レスポンス) 技術力不足 ・プロセス管理不足 ・基本設計レビューの未実施 ・仕様要件 (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)	○	・組織間の協同性・コミュニケーション								
⑪関連部署との連携不足	○	・協力会社・ベンダー先などのコミュニケーション能力								
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)	○	・要求仕様書誤りの設計 ・誤謬なドキュメントやガイドライン								
⑬開発のベースの相無	○									
⑭開発環境の不備	○									
⑮開発投入不足	○									
⑯開発投入戦略の誤り (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								
⑳情報不足 (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								

### 3-3-5. 課題管理表

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



## 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 品質マネジメントの役割**

**第2章 品質マネジメント 計画**

**第3章 品質マネジメント 実行**

**第4章 品質マネジメント 管理**

**第5章 品質マネジメント 振り返り**