

# プロマネ再生工場

## アクションガイド

### ◆スコープマネジメント編

開発すべき仕様の内容および範囲にかかわる問題解消のためのガイドラインです。



## 目次

### スコープマネジメント アクションガイド p 1

- s 1 課題：要求仕様問題の解決 p 1
  - ◎誤解を招く仕様書の日本語記述 p 1
    - ◆【アクションガイド：日本語表記の注意点】 p 1
  - ◎要求仕様書のチェックポイント p 1
    - ◆【アクションガイド：要求仕様書のチェックポイント】 p 1
  - ◎仕様問題解決策の概要 p 3
    - ◆【アクションガイド：要求仕様問題解決策の概要】 p 4
    - ◆【アクションガイド：ユーザー目線に立つこと】 p 4
    - ◆【アクションガイド：適切ではない要求仕様の見つけ方】 p 4
    - ◆【アクションガイド：提案行動】 p 4
- s 2 課題：仕様凍結 p 5
  - ◎【仕様凍結を阻害する三つの障壁】 p 5
    - ◆【アクションガイド：早期仕様凍結】 p 5
- s 3 課題：無理な顧客要求への対応 p 6
  - ◎なぜ代案を提案する行動に移れないのか p 6
    - ◆【アクションガイド：不条理な顧客要求への対抗策】 p 6
- s 4 課題：要求仕様理解の手順 p 7
  - ◎仕様内容の情報共有 p 7
  - ◎仕様調査段階で仕様不具合を発見することの重要性認識 p 7
    - 【手戻りコスト100倍の法則】 p 7
    - ◆【アクションガイド：要求仕様理解の手順】 p 7
- s 5 課題：要求仕様の変更管理 p 8
  - ◎ベースラインの意味 p 8
  - ◎仕様の変更管理が行われにくい理由 p 8
  - ◎要求仕様の変更管理不在がもたらす災い p 9
  - ◎仕様変更管理表がもたらす効果 p 9
  - ◎仕様変更の確認方法 p 10
    - ◆【アクションガイド：仕様Q & A表の要件】 p 10
    - ◆【アクションガイド：要求仕様の変更管理】 p 11
  - ◎仕様変更対応工数の請求について p 11
    - ◆【アクションガイド：仕様変更対応工数の請求】 p 11

- s 6 課題：異常系仕様の合理的な決定 p 1 2
  - ◆【アクションガイド：異常系仕様のあるべき姿】 p 1 2
- s 7 課題：性能要件 p 1 3
  - ◎パフォーマンス性能・レスポンス性能は重要なシステム機能要件 p 1 3
  - ◆【アクションガイド：適正なパフォーマンス・レスポンス時間の達成】 p 1 3
- s 8 課題：仕様変更影響度表の作成 p 1 4
  - ◎設計書に拠らない正確な影響度表の作成方法 p 1 4
  - ◆【アクションガイド：仕様変更影響度表の作成】 p 1 4

## s 1 課題：要求仕様問題の解決

### 誤解を招く仕様書の日本語記述

日本語の長々と続く要求仕様書を正確に読み解くことは至難の技です。このような要求仕様書の改善策としては次のようなものが考えられます。

- ① 図・表・論理記号を多用し誤解を招かない表現方法を採用する。
- ② できるだけ分かりやすい簡潔な文章にする。

#### ◆ 【アクションガイド：日本語表記の注意点】

- 主語がない文章には主語を書き加えること。
- 可能な限り短い文にすること。
- 並列的な表現は箇条書きにすること。
- 時間軸にて整理したチャートで現在・過去・未来の時制を明確にすること。
- 読み聞かせで聞き手がきちんと理解できるか確認してみること。
- 単位・期限・期間・対象範囲などの基準を明確にすること。
- %表記は何を 100%としたのかを明記すること。
- 程度表現（例：すごく、適当、ちょうど、およそ、等）を避け数値で表すこと。
- 強調表現（例：ちゃんと、しっかり、等）を避け数値で表すこと。
- 推測表現（例：～くらい、およそ～、等）を避け数値で表すこと。
- その他の曖昧表現（例：～の予定、できるだけ～、等）を避けること。

### 要求仕様書のチェックポイント

要求仕様書の品質を保つためには下記の 8 つのチェックポイントをおさえておく必要があります。

#### ◆ 【アクションガイド：要求仕様書のチェックポイント】

##### 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こと。使用する場合は、○月○日までに決めるとコメントをつけること。

### 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）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
- 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、範囲をあらわしていることを意識して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否か検証可能であること。
- 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。

### 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・分離して表現している。要求が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。

仕様の理解不足・誤解に起因する間違った記述をなくし要求仕様書の精度向上のためには以下のアクションが必要となります。



### 仕様問題解決策の概要

ベンダー側の要求仕様担当者にも優劣があり、仕事の開始前にその優劣を判断しておき、相応の対応をする必要があります。これは社内・社外を問わずに必要な、対人間におけるリスク対策といえるものです。次のような認識や対策が必要となります。

#### ①要求仕様には必ず誤りがあるという前提に立つこと

要求仕様内容が貧弱な原因は、仕様決定者において、その仕様の理解が不十分だということでしょう。理解が不十分なままで出される要求仕様は、それ自体の内容は必ず不完全であり矛盾や誤りを多く含んでいます。

#### ②要求仕様の疑問点・不明点を掘り起こすこと

全ての要求仕様に対して、その機能の意味や効果および論理性に疑問や不明点がないかどうかの自己チェックが必要で、最初の要求を受け取った時点でこの作業を必ず実行する必要があります。納得できないことは納得できるまで要求者に確認する必要があります。確認の方法はQ & A表に基づいたフェイス・トゥ・フェイスでの確認が最も効率的で正確です。相手の要求内容を鵜呑みにして、言われたことをそのまま開発すればよいというような受動的な姿勢は避けるべきでしょう。

#### ③要求仕様問題の解決策を実行すること

解決策については下記のアクションガイドの実行が必要です。

◆ **【アクションガイド：要求仕様問題解決策の概要】**

- 仕様凍結の期日を切り、お互いに合意しておくこと。
- 顧客価値・顧客要求度の高い仕様からの仕様凍結・開発を行うこと。
- あいまいな点・不明確な点・矛盾点などについて直接コミュニケーション・電話によるヒアリング・Q & Aシート等による確認を設計開始前に終了させること。待ちの姿勢や放置はいけない。
- 開発の初期工程においては要求元担当者との密な（週2～3回）コミュニケーションを重ね仕様変更・追加の有無を確認することで後工程での頻繁な仕様変更や仕様凍結の遅延を防ぐこと。
- 仕様の条件やノウハウである要求仕様の背景や理由について必ず確認し文書化すること。
- 経験不足な部分は有識者への質問や相談を行うこと。
- 待ちの姿勢ではなく、こちらからも仕様提案を行うこと。
- 仕様提案力をつけるためにベンダー側のS E・顧客打ち合わせ等へ参加すること。

◆ **【アクションガイド：ユーザー目線に立つこと】**

- さまざまなエンドユーザーの視点を持つこと。エンドユーザーの例：オペレータ・受発注担当者・商品配送者・商品開発者・運用管理者・損益管理者・クレーム管理者、など。
- ユーザー目線の本質は、顧客および開発側にて長年蓄積されてきたノウハウであると認識し、要求仕様の妥当性は自己の経験・観察やドキュメントおよび経験者に拠ること。

◆ **【アクションガイド：適切ではない要求仕様の見つけ方】**

- ①明らかにその仕様では正しい機能が実現できない場合（ビジネスロジックの欠陥）
- ②H/WおよびS/Wの性能上実現が不可能と思われる場合
- ③従来同じ業種・業態で広く受け入れられている入出力方式（操作手順、表示方法、印字方法等）と異なっている場合
- ④ソフトウェアの処理速度（パフォーマンス、レスポンス）に大きな悪影響を及ぼす場合
- ⑤他の方法でもっと簡単に顧客要求が実現できる場合

◆ **【アクションガイド：提案行動】**

- 代案を考えるだけの知識および相手を納得させるためのコミュニケーション能力を持つための学習や改善活動を継続実行すること。
- 忙し過ぎて提案する時間がないならば、効率的な提案を行うことで時間の余裕を生み出すこと。



## s 2 課題：仕様凍結

◎ 想定仕様ではなく確定仕様に基づく開発を！

### 【仕様凍結を阻害する三つの障壁】

1. 組織間の仕様決定能力の分断
2. 情報伝達の組織間のタイムラグ
3. 組織間のコミュニケーション・ギャップ

### ◆ 【アクションガイド：早期仕様凍結】

- 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、元請側と下請側で合意しておくこと。
- 顧客・元請・下請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。
- 受注者側においても提案型仕様凍結を行うこと。
- 顧客価値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。
- 仕様未凍結状態ないしは疑問点・不明点を残したまま先行開発着手は行わないこと。

### s 3 課題：無理な顧客要求への対応

#### なぜ代案を提案する行動に移れないのか

代案を提示する行動に移れない原因としては下記のようなことがあります。

##### 【能力的な面】

- ・代案を考えるだけの知識がないこと。
- ・代案を相手に納得していただくためのコミュニケーション能力がないこと。

##### 【心理的な面】

- ・コミュニケーションに自信がないこと。
- ・苦手な相手との会話を避けたがること。
- ・面倒くさいと思うこと。
- ・提案しても何も自分の得にはならないと思うこと。

##### 【環境的な面】

- ・忙し過ぎてやっている時間がないこと。

不条理な顧客要求への対抗策についてのガイドラインは下記の通りです。

#### ◆ 【アクションガイド：不条理な顧客要求への対抗策】

- 開発責任者と要求側責任者の密接かつ直接的な話し合いを実行する。
- 完全な要求仕様書の早期提示時期の確約をとる。
- 未凍結仕様は、双方の合意の上、要求者において早期凍結時期を確約していただく。
- 開発仕様はすでに凍結合意されたものに限定する。

《困難な要求に対しては、更に困難な条件の提示を行うこと》

- 要求元からPM、有識技術者・テストメンバー等の投入を行っていただく。
- 開発要員増強のためのプレミアムコストの要求を行う。
- 暫定版または優先度順のリリース、分割リリース等の条件提示を行う。
- 暫定版の不具合発生責任は要求者に負っていただく。
- 暫定版の市場投入は、限定数にて実験店舗のみで実施する。
- 全店稼働は正式版のみで実施する。

一見無理だと思われる要求を受けた場合の交渉における基本姿勢は、まずはYesと答えて、butで実行条件を示すことに尽きます。良く考えれば、さまざまな実行条件が出て来るはずで

自分の弱みにばかり気を取られていると敵が見えなくなります。敵の弱みにも注目するところからタフな交渉が可能となります。“All or Nothing”的なデジタル思考をやめ、敵をよく知ることから交渉を始めたいものです。

## s 4 課題：要求仕様理解の手順

### 仕様内容の情報共有

仕様調査用Q & Aリストは解決済み内容・未解決項目の解決予定日・変更日など必要事項を記入し、変更部については日次情報共有会議などで全員に周知徹底する必要があります。

またこれらの情報は、当該プロジェクトの掲示板に更新日の記載をもって全員に通知および公開しておく必要があります。メールベースだけではとても情報共有することは難しいでしょう。

### 仕様調査段階で仕様不具合を発見することの重要性認識

仕様の誤解などによるミスはQ C Dすべてに大きな損害を与えてしまいます。仕様ミスの発見が開発工程の初期の段階であれば被害は少なく済みます。仕様ミスによる手戻りコストについては一般的に次のように言われています。

#### 【手戻りコスト100倍の法則】

要求仕様の欠陥を要件定義工程内で修正するコストを1とした場合、

- ▶設計段階からの手戻り修正コストは**5倍**
- ▶コーディング段階からの手戻り修正コストは**10倍**
- ▶テスト段階からの手戻り修正コストは**20倍**
- ▶納入後からの手戻りコストは**100倍**

#### ◆ 【アクションガイド：要求仕様理解の手順】

- 仕様調査・検討に必要な時間を確保しておくこと。
- まずは仕様の全体像および骨子の把握から始めること。
- 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。
- 仕様追加・変更の影響度の事前調査を済ませておくこと。
- 要求仕様の背景や意味を必ず明確にしておくこと。
- 開発経験者から基本的な仕様のレクチャーを受けること。
- 過去の失敗事例に学ぶこと。
- 実機操作によって客先システムの動作を体感しておくこと。
- 疑問・不明点の発掘を行い、要求者側との直接コミュニケーションで即断即決を目指すこと。
- 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。
- 期限を切って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。

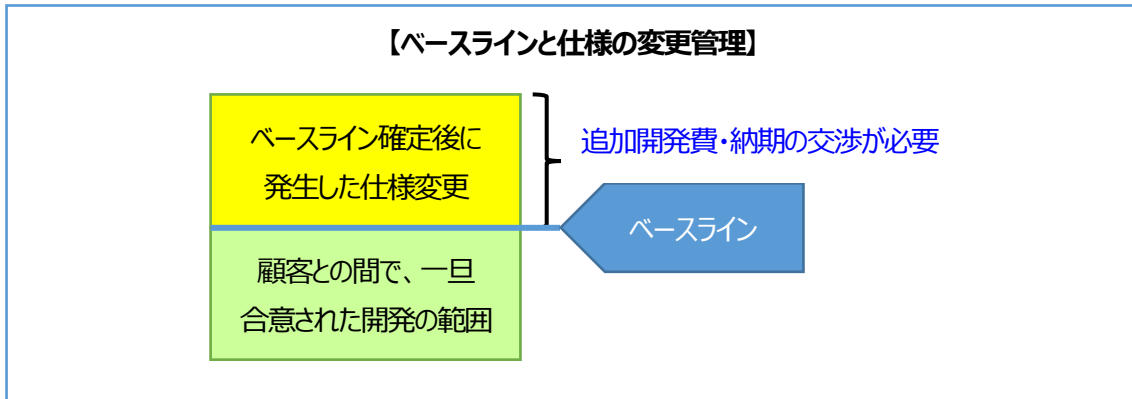


## s 5 課題：要求仕様の変更管理



### ベースラインの意味

- ◎ **ベースライン（基線）とは、**
  - ・ 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。

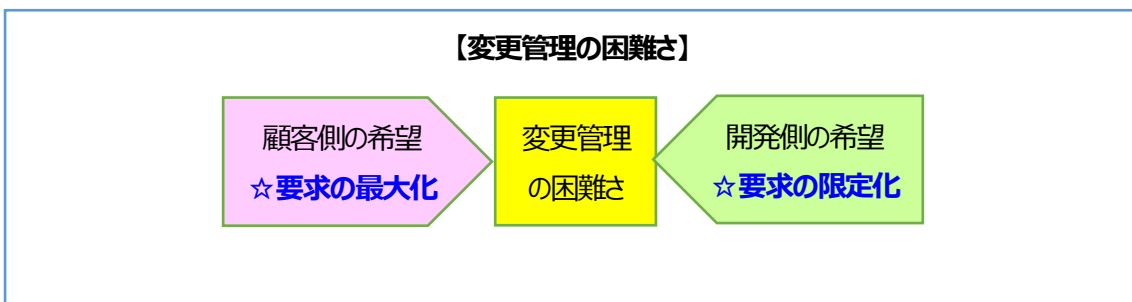


### 仕様の変更管理が行われにくい理由

仕様の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。



- ◎ **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。**

## 🔄 要求仕様の変更管理不在がもたらす災い

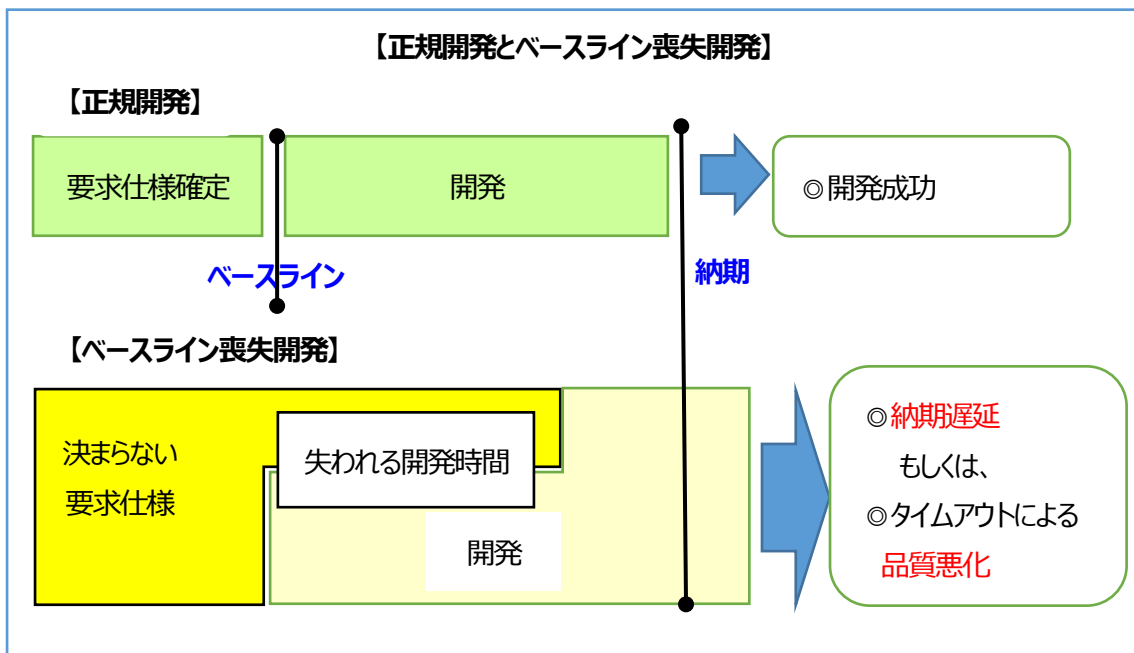
【ベースラインの確定・変更管理の不在がもたらすリスク】

1. プロジェクトの失敗 ⇒ Q C Dの大幅な未達
2. 顧客ビジネスの停止 ⇒ 市場障害の多発

## 🔄 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

◎ **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になります。

### 【仕様変更管理表】

【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														

### 仕様変更の確認方法

#### ◆ 【アクションガイド：仕様Q & A表の要件】

- 要求仕様に関する疑問点・不明点は仕様Q & A表等にて管理すること。
- Q & A表の記述内容は、誰が読んでも分かる記述になっていること。
- 同じ仕様に関するQ & Aは同じ場所に時系列順に並んでいること。
- 複数のキーワードで検索可能な書式になっていること。
- 仕様Q & Aで明確になった内容は、要求仕様書も同時にメンテナンスすること。
- 仕様Q & A表は要求仕様書の添付資料としてワンセットで保管・管理すること。

#### ◆【アクションガイド：要求仕様の変更管理】

- 要求仕様のベースラインには、顧客と開発側で合意した、開発仕様の内容・範囲の定義およびその開発に関するQCD（開発費、日程、品質）条件を含むこと。
- 開発着手前に要求仕様のベースラインを確定しておくこと。
- 要求仕様のベースラインを基準とした変更管理を仕様変更管理表に基づいて行うこと。
- 仕様変更管理表で管理される項目には、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・変更担当者名・影響度範囲・難易度・リリースバージョン名・見込み工数・実績工数などを含むこと。
- ベースラインから変更・追加されたものに要する追加開発費・開発期間は別途見積りとする。



#### 仕様変更対応工数の請求について

仕様変更を無料で押し付けようとする相手に対応するためには以下の対策を行う必要があります。

#### ◆【アクションガイド：仕様変更対応工数の請求】

- 見積り条件に、「見積り回答書に記載にないものについての追加・変更は全て再見積りおよび追加開発費・追加開発期間が必要である」旨の記述を必ず記載しておくこと。
- 「仕様変更依頼は必ず書面にて、その変更の理由および背景を明示して行うこと」が記載された仕様Q & A表のフォームを先に渡しておくこと。
- 顧客と仕様変更についての打ち合わせの都度、上記のことを確認しておくこと。



## s 6 課題：異常系仕様の合理的な決定

### 【異常系対応の判断基準】

- ・ 異常系対応の判断に過剰とか完璧とか想定外などという情緒的な考え方をしないこと。
- ・ 異常系対応の判断基準に、発生頻度・難易度・コストなどを持ち込むべきではない。

### 【例外処理の考え方】

- ・ 例外処理には、異常系処理だけではなく正常系処理も存在する。
- ・ 想定外や例外の状況になっても、ロックや永久ループの状態に陥ってはいけない。

### ◆ 【アクションガイド：異常系仕様のあるべき姿】

- あるべき異常系仕様には、「過剰な、完璧な、想定外」などの非ロジカルな言葉は使用しないこと。
- 例外処理には、異常系処理だけではなく正常系処理も存在する。
- 想定外や例外の状況になっても、装置側はロックや永久ループの状態に陥ってはいけない。
- レギュラーなデータおよび状態に対してはすべてエラー処理が必要となる。
- アプリケーションロジックについて、全ての正常系および異常系の処理が過不足なく網羅されていること。
- 外部機器との I / F においては、入出力条件のチェックおよび異常処理を網羅すること。
- 機器単独では決められない場合は、他の機器担当および顧客との合意を得る必要がある。

## s 7 課題：性能要件



### パフォーマンス性能・レスポンス性能は重要なシステム機能要件

レスポンスとパフォーマンスは重要なシステム機能要件の一つであり、全ての処理に対してレスポンス・パフォーマンスを考慮した設計や製造を行うことは常識中の常識です。業務ロジックが正しくても、適正な時間内に処理や応答がないようなプログラムは意味がありません。

#### ◆【アクションガイド：適正なパフォーマンス・レスポンス時間の達成】

- ソフトウェアのパフォーマンス時間はシステム要件を満たしているか。
- ソフトウェアのレスポンス時間はシステム要件を満たしているか。
- サーバー／DB／HDD等へのアクセスを行うプログラムにおいては、パフォーマンスおよびレスポンスは最も重要な要件となるという認識を全員で再度共有すること。
- 同様に、これらのアクセスを行うプログラムにおいては、要求仕様書や設計書にパフォーマンス目標値とレスポンス目標値についての記述を必ず記載し、レビューおよびテスト項目に加えること。

(具体的な達成数値目標の設定例)

- 表示の切り替わり速度：200ms以下
- タッチパネルの反応速度：100ms以下
- スキャナーの反応速度：10ms以下
- ファンクションキー押下から機器動作開始までの時間：200ms以下

これらのシステム性能要件はシステム仕様書および要求仕様書に最初から明記されるべきものです。さらにもし記述モレがあったとしても何らかの性能要件が必須であるという常識を開発者全員が持っている必要があります。自分が設計・製造したソフトが実際に実機でどのように動作するのかを見ることは必須のことでしょう。

## s 8 課題：仕様変更影響度表の作成

### 設計書に拠らない正確な影響度表の作成方法

既存仕様への影響度を調査する仕事は、第一義的には開発チームの義務であると言えます。既存仕様に対する影響度を調査しなければ不具合のない設計も製造もできません。派生開発においては、開発チームは既存仕様に対する悪影響を避けながら設計および製造を行わなければ正しい開発を行ったとは言えません。評価チームは開発チームから提供された既存仕様に対する影響度表を基に評価設計を行うのが普通のやり方です。その上で評価チームは今までの評価の経験を活かして開発チームが見落としそうな影響部分のテストを追加するのが正常なあり方でしょう。

影響度表の作成には正確な設計書が必要ですが、現在多くの設計書は適切な更新が行われていない場合が多く、影響度表の作成は困難を極めています。このような状況下において、ある程度使える影響度表を作成するためのポイントは下記の準備活動が必要です。

#### 【影響度表作成のための準備活動】

単体テスト、結合テスト、総合テストにおいて発見された他のモジュールに影響した不具合ならびに影響が及んでいると気づいたことを、全のテスト実行者が協力して影響度表に書き込むことを継続して実行すること。

当初は大雑把な影響度表しかなかった場合でも、派生開発の都度このことを実行していれば、完全な設計書がない場合でも、回数を重ねるごとに非常に精度の高い影響度表になっていくでしょう。開発チームにおける単体・結合テストおよび評価チームにおける総合テストにおいて両チームが連携してこの活動を地道に実行する必要があります。

影響度表は、本来ソフトウェア構造図・テーブル関連図・プロセスフロー等から作成されるべきものですが、多くのプロジェクトにおいてはこれらのドキュメントが整備されていることが少ないのが実態だと思われます。そのような状況下においては、下記のアクションガイドに示したように、開発の各工程の作業時にできるだけ情報を集めながら影響度表を徐々に作り上げていく必要があります。

#### ◆ 【アクションガイド：仕様変更影響度表の作成】

- 正確な構造設計書・テーブル関連図・プロセスフローから影響度表を作成すること。
- 見積り時の実装調査にて判明した他機能への影響情報を記録すること。
- 設計時において知りえた影響情報を記録しておくこと。
- 製造時において知りえた影響情報を記録しておくこと。
- デバッグ時に判明した他機能への影響情報を記録すること。
- 単体・結合・総合の各テストにおいて判明した他機能への影響情報を記録すること。
- 市場障害対応にて判明した他機能への影響情報を記録すること。