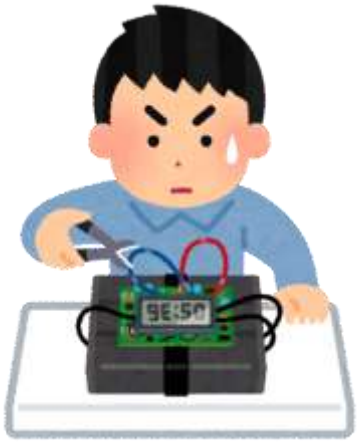
ITプロジェクト

要件定義リスク

クイック・リファレンス



PMファクトリー　2016年

目次

はじめに

下記は日経BP社による「プロジェクト実態調査800社（2009 年 2 月 2 日）」による失敗プロジェクトの原因分析結果です。

品質問題を起こしたプロジェクト
1. テストが不十分・移行作業問題　41．7%
2. 要件定義が不十分　36．7%
3. システムの設計が不正確　31．7%
4. システム開発の質が悪い　31．7%
5. エンドユーザーへの教育不十分　31．7%

コスト問題を起こしたプロジェクト
1. 追加の開発作業が発生　58．9%
2. 追加の設計作業が発生　47．5%
3. 追加の企画作業が発生　32．6%
4. テストが長引いた　16．3%
5. ハードの追加購入や高性能化　7．8%

納期問題を起こしたプロジェクト
1. 要件定義が長くなった　43．6%
2. 設計作業が長くなった　33．0%
3. 開発作業が長くなった　33．0%
4. テストが長くなった　26．8%
5. 企画作業が長くなった　16．2%

いずれにしても要件定義の生産性の悪さがプロジェクト全体に悪影響を及ぼしていることは一目瞭然です。
これらの問題点を開発工程順に並べると次のようになります。

（1） 準備段階において
コスト； 追加の企画作業が発生し、ハードの追加購入や高性能化がインパクトを及ぼした。
納期； 企画作業が長くなった。

（2） 上流工程（要件定義、システム設計）において
品質； 要件定義が不十分なため、システムの設計が不正確になった。
納期； 要件定義が長引いた。

（3） 中流工程（ソフトウェア設計、プログラミング、単体テスト）において
品質； システム開発の質が悪い。
コスト； 追加の設計作業や他の開発作業が発生した。
納期； 設計作業や他の開発作業が長引いた。

（4） 下流工程（結合テスト、総合テスト）において
品質；テストが不十分だった。システム移行作業に問題が出た。エンドユーザーへの教育が不十分だった。

コスト； テストが長引いたためコスト増を招いた。
納期； テストが長引いた。

これらから見えるプロジェクトの主な問題点は下記のように集約できます。
1. 要件定義が不十分であり長引いている。
2. システム設計が不正確であり質が悪い。
3. 追加開発が発生する。
4. テストが不十分。

これらの事実を一連の文章にすると次のようになります。
顧客およびベンダーともに下請け依存的な姿勢により要件定義能力が低下し、そのために要件定義工程は所定の期間内に終了できず、さらに要件定義内容も貧弱であり漏れも多い。
最上流工程におけるこの問題は中流工程の設計工程においては質の悪い設計や予定外の追加開発を発生させ開発工程を遅延させてしまっている。これら上流・中流工程の品質の悪さおよび工期遅延は下流工程におけるテスト時間の大幅な不足をまねき、結果として検証不足の粗悪な品質の製品を納期遅れでしかも赤字でリリースしている。

もう少し要件定義工程について考えてみたいと思います。
一般的に見積りや契約は要件定義が終了しない内に行われてしまっており、要件定義が終了する以前にすでに開発費や納期は決められている場合がほとんどでしょう。遅れて決まった要件定義内容は、すでに決まった開発費や開発期間より多くの開発費および長い開発期間を必要とする場合がほとんどです。
これはプロジェクトの成功条件である妥当なコストおよび時間の条件が成立していないことを意味しています。コストと時間の前提が崩れたプロジェクトの結果は品質低下しかありません。プロジェクトは開始される前から失敗を運命づけられているようなものです。

顧客とベンダー間の下請け依存的な姿勢による要件定義能力の低下は、開発のいろいろな局面でもプロジェクトの真の阻害要因としてその名前を変えて登場してきます。
例えばベンダー側から協力会社への仕事の丸投げが一括請負というもっともらしい名前で実行されています。またオフショア開発という名前で他国に丸投げされています。発注元は請負側に実務を丸ごと依存し、請負側は発注元に丸ごと売り上げを依存している構図になっています。そのような状況は間違いなく関係組織全体に仕事をやり抜く情熱や責任感の低下を招くことになるでしょう。ものづくりの指標を利益第一主義でしか見られないマネジメントの下では開発組織はまちがいなく衰退していく運命にあるでしょう。

本書は、プロジェクトの最大のリスクであり双子のリスクである見積りを含む事前準備工程のリスクと要件定義工程のリスクに焦点をあて、IPA／SEC（独立行政法人　情報処理推進機構ソフトウェア・エンジニアリング・センター）が公開している失敗事例に対して、それらのリスクの回避策およびアクションのチェックポイントを示したものです。
なお「要件定義」は「要求仕様」とも呼ばれることがありますが、本書においては「要件定義」という文言に統一しました。

第1章　プロジェクトにおける具体的なリスク要因

プロジェクトをとりまくリスク要因は下記の通りでしょう。本書におけるリスク要因の分類分けは、読者がイメージし易いようにリソース（ヒト／モノ／資金／情報）による分類と業務内容（マネジメント／技術）による分類で行いました。

1）ヒトに関するリスク要因一覧

ヒトに関するリスク要因は、下記に示した１１項目にグルーピングしました。

① 他者依存的姿勢（マネジメントリスク）
② 上位マネジメントの関与不足（マネジメントリスク）
③ ユーザーの参加・協力度不足（マネジメントリスク）
④ 組織能力不足（マネジメントリスク）
　＊未熟な組織文化（情緒的人間関係、非科学的な情緒的思考、情報軽視、データ軽視、認識力の弱さ、コミュニケーション機能不全、丸投げ体質）
　＊　戦略の欠如（問題に対する取り組みの姿勢、情報戦略）
⑤ 見積り能力（マネジメントリスク）
⑥ 要件定義能力（技術リスク）
⑦ リーダーのプロジェクトマネジメント能力（マネジメントリスク）
　＊外部交渉能力
　＊タイムマネジメント能力（時間の有効利用、アクションタイミングの適時性、ヒト・資金の資源投入時期・量のミスなど）
　＊現場主義の励行（部下との常時コミュニケーションの実行）
　＊見える化能力（見えない問題の顕在化能力）
⑧ メンバーの技術能力（技術リスク）、うっかりミスなどのヒューマンエラー（マネジメントリスク）
⑨ プロセス管理の有無（マネジメントリスク）
⑩ コミュニケーション能力（阻害・組織間ないしは人間間のギャップ）（マネジメントリスク）
⑪ 関連部署との連携不足（マネジメントリスク）

上記のヒトに関するリスク要因をマネジメント領域と技術領域で分類すると次のようになります。

【マネジメントに属するリスク】

① 他者依存的姿勢
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
⑤ 見積り能力
⑦ リーダーのプロジェクトマネジメント能力
⑨ プロセス管理の有無
⑩ コミュニケーション能力
⑪ 関連部署との連携不足

【技術に属するリスク】

⑥ 要件定義能力
⑧ メンバーの技術能力、ヒューマンエラー

2） モノに関するリスク要因一覧

モノに関するリスク要因は、下記に示した三つにグルーピングしました。

⑫ ドキュメント（要件定義書・設計書・チェックリスト・手順書など）の不備（技術リスク）
⑬ 開発のベースの有無（技術リスク）
⑭ 開発環境の不備（技術リスク）

モノに関するリスクは一部マネジメントリスクとも関係しますが主に技術的なリスクと言えます。

3） 資金に関するリスク要因一覧

資金に関するリスク要因は次の二つにグルーピングしました。

⑮ 開発費不足（マネジメントリスク）
⑯ 資源投入戦略の誤り・開発費投入時期ミス（マネジメントリスク）

資金に関するリスクは全てマネジメントリスクだと言えます。

4） 情報に関するリスク要因一覧

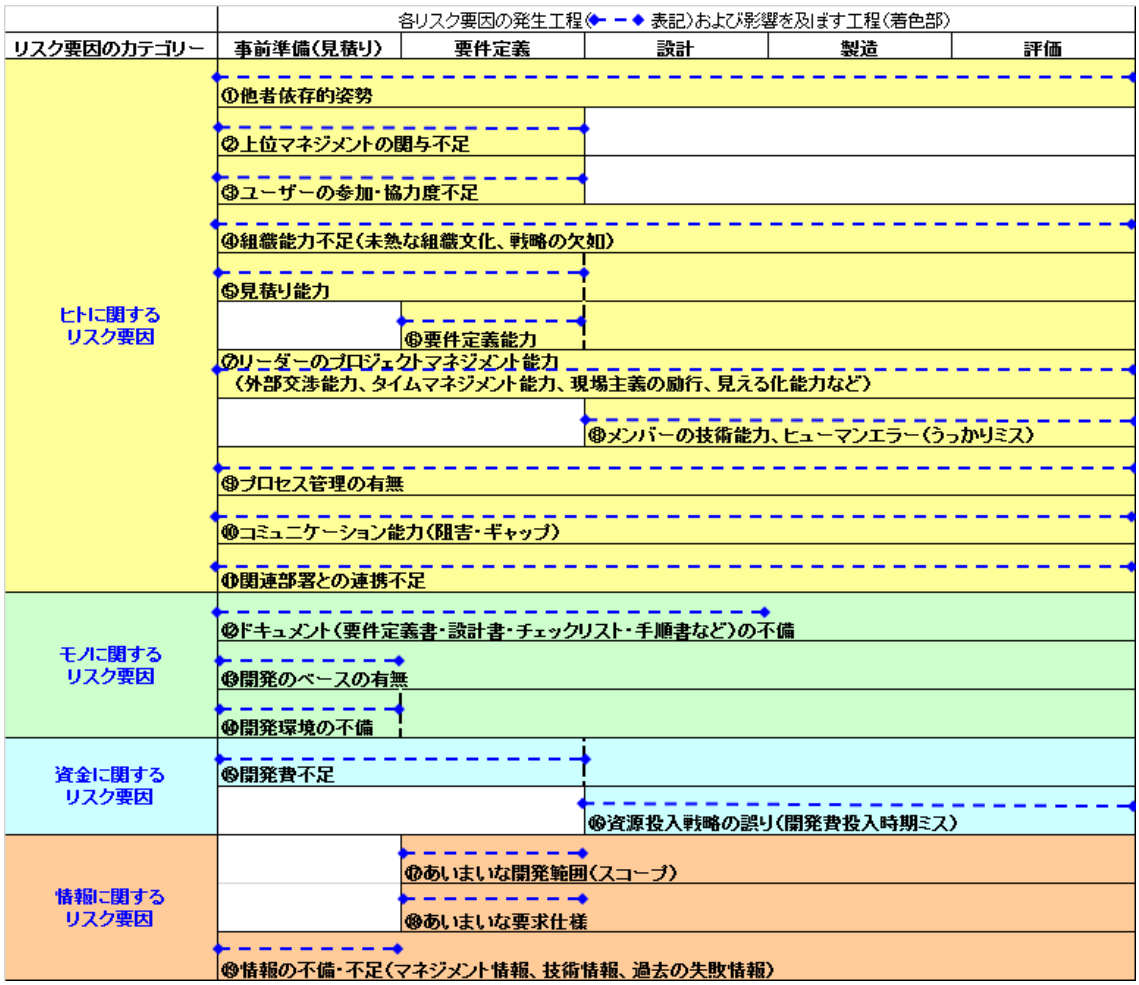
情報に関するリスク要因は次の三つにグルーピングしました。

⑰ あいまいなスコープ。目的・開発範囲の不明確さ・複雑さ。
⑱ あいまいな要件。要件定義の不明確さ。
⑲ 情報の不備・不足
　＊ マネジメント情報の不備、技術情報の貧困さ、失敗情報の蓄積不足

情報に関するリスクは全てマネジメントリスク＋技術リスクの複合リスクだと言えます。

以降の文中で使用されるリスク番号は、上記の①から⑲とします。

前記の19種類のリスクを要因別・工程別に表したものが、下記のリスク要因マップです。



【リスク要因マップ】

5） 重要失敗事例に学ぶリスクの回避または排除の方法

失敗事例に関してはIPA／SEC（独立行政法人　情報処理推進機構　ソフトウェア・エンジニアリング・センター）にて公開されているITプロジェクトの「見える化」の上流工程編・中流工程編・下流工程編の付録に掲載されている「プロジェクトにおける問題事象と対策の事例集」のうち事象概要および事例における「見切り判断」の内容部分を引用し、リスク分析および対策については筆者の経験に基づいて独自に記述しました。

すべての失敗事例から学べることは次の通りでした。

1. 失敗の80%はヒトに起因する。

2. 失敗要因の50%以上が上流工程（事前準備、要件定義）で発生している。

3. 失敗要因のワースト5は次の通りです。
   ① 組織能力不足
   ② リーダーのプロジェクトマネジメント能力不足
   ③ 他者依存的姿勢
   ④ メンバーの技術能力不足
   ⑤ ドキュメントの不備

次章から、実際に起きた失敗事例をあげてそのリスク要因および回避策を示すことにします。

第2章　事前準備工程におけるリスクと回避策

事例1．　情緒的思考による不適切な受注　[見積りリスク]

◎不条理な予算圧縮要請で開発に着手

【事象概要】

システム化の対象領域の業務知識に乏しかったが、受注しないと、この先顧客との取り引きが続かないという危機感があった。請負契約をしてしまったが、大幅な工数増になり、プロジェクトの想定していた金額よりも採算が大きく悪化した。[SEC]

【判断の誤り】

顧客からの「簡単にできる」という言葉にも反論できず、顧客予算に合わせた見積額の圧縮要請を受け入れて受注し、開発に着手した。赤字になっても受注するほうがよいと判断した。[SEC]

【発生問題】 損益悪化

【リスク要因】

キーリスクは、④組織能力不足（マネジメントリスク）にある。

関連リスクは次の通り。

⑤　見積り能力（マネジメントリスク）
⑥　要件定義能力（技術リスク）
⑦　リーダーのプロジェクトマネジメント能力（マネジメントリスク）
⑩　コミュニケーション能力（阻害・組織間ないしは人間間のギャップ）（マネジメントリスク）

【リスク回避策】

一言でいうと受注責任者の無責任さが問題の真因です。知識もない領域なのに契約欲しさ優先で何をどのように開発するのかの考えもなく受注してしまうことなど常識では考えられないことです。仕事の中身を追及せずに契約継続しか頭の中にないようで、この責任者は開発組織の責任者としては不適合でしょう。

この事例のリスク対策は、そもそもこの仕事自体を請けるべきではなかったということでしょう。もしどうしても請けるということなら、この開発領域について知識・経験のある他のベンダーなりの協力のもとに見積りおよび受注をするべきで、赤字がどの程度になるかを見極めた上で戦略的な物件として受注するのか、それとも辞退するのかを決めるべきでした。

【リスク回避のアクション・チェックリスト】

◎知識のない領域の請負開発

□　①　知識・経験を保有している他のベンダーと協力して受注すること。
□　②　協力者が見つからない場合は請負を辞退すること。
□　③　赤字額を予め見積り、戦略物件として受注するか辞退するかを決めておくこと。

事例2.　パッケージをベースにした顧客要件開発の落とし穴 ［見積りリスク］

◎作りながら仕様を確認

【事象概要】

ノウハウの無い業界相手にパッケージ製品を導入し、仕様の詳細は顧客にヒアリングしながら開発していく計画を立てていた。しかし、ヒアリングを進めると見た目や機能名からは想像できないような仕様が続々明確化され「無いと業務にならない」「常識だ」という言葉に踊らされて開発規模が増大。納期遅延を引き起こした。[SEC]

【判断の誤り】

知らない事を自分の都合の良い方に考え「仕様は不明確だがベースとなるパッケージが必要な部分を充足してくれているだろう」と期待してしまった。[SEC]

【発生問題】 損益悪化、納期遅延

【リスク要因】

キーリスクは、④組織能力不足（マネジメントリスク）、特に未熟な組織文化（非科学的な情緒的思考、認識力の弱さ）と戦略の欠如（問題に対する取り組みの姿勢の弱さ）にある。

関連リスクは次の通り。

⑦ リーダーのプロジェクトマネジメント能力（マネジメントリスク）
⑰ あいまいなスコープ。目的・開発範囲の不明確さ・複雑さ。
⑱ あいまいな要件。要件定義の不明確さ。

【リスク回避策】

業界仕様のノウハウがないことが最大のリスクではないでしょう。最大のリスクはベンダー側における”ヒアリングしながら開発していく”という考え方です。正しい仕事の仕方は、まず事前準備段階で顧客要件を十分にヒアリングしてベンダー保有のパッケージ機能とのフィット＆ギャップを精査することです。次に判明したギャップ要件について適正な見積りを顧客に提示する必要があります。開発着手は顧客が見積りを受け入れた後にすることでしょう。これはビジネスの基本中の基本です。すでに開発に着手してしまった後では顧客に追加見積りを出しても受け付けてもらえないのは当たり前のことです。

【リスク回避のアクション・チェックリスト】

◎パッケージをベースにした顧客要件の開発の落とし穴

□ 開発にいたるまでの適切な手順は、次の通り。
□ ① 顧客要件とパッケージ機能とのフィット＆ギャップを精査すること。
□ ② ギャップ要件についての見積りを顧客に提示すること。
□ ③ 見積りの顧客承認後、開発に着手すること。

最初に低予算ありきで顧客要件を聞きながら開発を進めていくのは大きな間違いです。

事例3.　資産流用に関する見積りミス ［見積りリスク］

◎既存資産が流用できると思っていたが、流用できないことが分かりコストオーバー

【事象概要】

ある地方自治体のシステム化案件。他の自治体で、同じ業務処理をするシステム案件を実施したことがあったので、そのときの成果を流用できると判断して見積もりをし、開発に着手した。

ところが、開発を進めていくうちに、業務処理が想定と大きく異なり、既存のシステムは流用できないことが判明。結果として大幅に開発規模が増大し、開発期間も長期化した。[SEC]

【判断の誤り】

類似システムが流用可能と考えて、それを前提に原価を低めに見積もり、プロジェクトを計画した。[SEC]

【発生問題】 損益悪化、進捗遅延。

【リスク要因】

キーリスクは、⑤見積り能力（マネジメントリスク）にある。

関連リスクは次の通り。

④　組織能力不足（マネジメントリスク）

⑰　あいまいなスコープ。目的・開発範囲の不明確さ・複雑さ。

⑱　あいまいな要件。要件定義の不明確さ。

【リスク回避策】

既存資産が流用可能との情緒的な思い込みにより顧客要件の精査を実施しなかったことによる見積りミスが失敗の真因です。既存資産の流用の可否がこのプロジェクトのキーリスクとなる場合は、顧客要件と既存資産とのフィット＆ギャップについて事前の精査が必須なことはだれにも容易に思いつく常識です。

【リスク回避のアクション・チェックリスト】

◎既存資産流用の可否

□①　顧客要件と既存資産とのフィット＆ギャップについて事前の精査を行うこと。

□②　想定あるいは推定仕様に基づく開発を行ってはいけない。

事例4.　見積りルール違反　[見積りリスク]

◎営業がプロジェクト・マネージャの承認のないまま概算見積もりで受注契約を結んでしまった
【事象概要】
　顧客側のシステム投資予算が決まっているようであり、ベンダー側の営業部門はその予算額をある程度把握しているようだった。営業部は開発部から見積りを取る前に、その予算に収まるよう、概算で予算を見積もって顧客に提示し、契約した。しかし、最終的には、営業部の見積もりを大きく上回る開発工数を要し、大幅赤字に追い込まれた。[SEC]

【判断の誤り】
　営業が詳細見積り未完了のまま請負金額を確定させてしまっており、プロジェクト・マネージャによる見積もりを実施しないまま開発を着手してしまった。[SEC]

【発生問題】 損益悪化

【リスク要因】
　キーリスクは、④組織能力不足（マネジメントリスク）にある。特に営業部における情緒的人間関係や非科学的な情緒的思考や開発部と営業部間のコミュニケーションの機能不全がプロジェクトを失敗に導いている。

　関連リスクは次の通り。
　② 上位マネジメントの関与不足（マネジメントリスク）
　⑩ コミュニケーション能力（阻害・組織間ないしは人間間のギャップ）（マネジメントリスク）
　⑪ 関連部署との連携不足（マネジメントリスク）
　⑲ 情報の不備・不足

【リスク回避策】
　営業部門が開発側の見積りも取らずに勝手に見積りを出したのなら開発側には何も責任はありません。この問題はベンダー側の社内コミュニケーションの問題です。
　開発側としては、営業側におけるルール破りが判明した時点で営業側とキッチリ話をつけておくべきです。事例内容からは読み取れませんが、もし赤字を開発部で聞き受けたとするならば、それも問題行為となります。 全社的な観点で戦略的物件の位置づけにある物件なら会社で損失を負担するなどの対応が必要となり、営業側の意識的な違反ならば営業部で負担してもらうのがビジネスの筋だと言えます。

【リスク回避のアクション・チェックリスト】

◎見積りのルール破り
□ ① ルール破りの責任の所在は明確にしておくこと。
□ ② 赤字などの損害はルール違反をおかした部署で背負うこと。
□ ③ ルール違反をおかした部署からは再発防止の約束を取り付けておくこと。

事例5.　マルチベンダー下における責任分担の不明確さ

◎仕切る気がないとりまとめ役

【事象概要】

大きな開発物件で、複数ベンダーが一緒に開発を受注した。開発範囲の中でどこまでをどのベンダーが構築するかという分担が中々決まらずに時間だけが経過。責任分担が不明確な機能の重複や、機能自体が無い物が発生してしまった。[SEC]

【判断の誤り】

会社間での調整が不十分であることがわかっていたが、開発が進むうちに開発分担が決まっていくだろうと思い、とりあえずプロジェクトは開始された。[SEC]

【発生問題】 品質不良および損益悪化

【リスク要因】

キーリスクは、①他者依存的姿勢（マネジメントリスク）にある。自律性・積極性のなさがプロジェクト管理のあいまいさやコミュニケーションの不全を生みプロジェクトを失敗に導いている。

関連リスクは次の通り。

② 上位マネジメントの関与不足（マネジメントリスク）
③ ユーザーの参加・協力度不足（マネジメントリスク）
④ 組織能力不足（マネジメントリスク）
⑥ 要件定義能力（技術リスク）
⑦ リーダーのプロジェクトマネジメント能力（マネジメントリスク）
⑩ コミュニケーション能力（阻害・組織間ないしは人間間のギャップ）（マネジメントリスク）
⑪ 関連部署との連携不足（マネジメントリスク）
⑰ あいまいなスコープ。目的・開発範囲の不明確さ・複雑さ。
⑱ あいまいな要件。要件定義の不明確さ。

【リスク回避策】

複数の担当ベンダー間での責任の押し付け合いによる協調性・コミュニケーションの悪さから、誰が何を担当するのかが不明瞭なままで、結局開発機能の重複ないしは欠落などの品質問題が発生している。

このような問題は実際の開発がテキトウに始まってからでは手の打ちようがない。プロジェクトの準備段階のうちにすぐに手を打つべきだった。だれもリーダーシップをとらないなら、これではダメだと気づいた者が何とかすべきだろう。最初の言い出しっぺが損をすることは絶対にない。最初の言い出しっぺはものごとの方向性を決定づける有利な立場を手に入れることができる。

まず顧客を巻き込んであなたがリーダーシップをとるべきだった。自分の立場が弱いのなら自分の上司にも参加してもらう必要がある。顧客側ないしは他のベンダーの誰がキーマンなのかもよく見極める必要がある。話しても効果の薄い相手を選んではいけない。

またこのような問題が発生する背景に顧客側が開発の範囲（スコープ）を明確にできていないことがある。全体が見えていなければ、将来の負担を軽くするために現在の負担をなるべく減らしておこうと各ベンダーは考える。だからといっていつまでも綱引きのお見合い状態を続けるわけにはいかない。タイムリミットになる前に開発範囲の全貌を決定するように自社の提案も含めて顧客側にアプローチする必要がある。これが自律性のある組織のやり方だと言える。

また話し合いをうまくまとめるためには関係者を説得するための戦略をキチントもっていなければいけない。自分だけ自社だけが有利になるというような姿勢はすぐに見透かされる。何の機能をどのベンダーが担当すべきかを決めるにはチャントした必然性のある合理的な理由が必要になる。判断基準で一番分かりやすいのがお金だ。多くの金額を受注するベンダーが中心的な役割を担うことは必然的で合理的なことだ。

顧客のキーマンに上手に扇のかなめになって動いていただくように促し、みんなが納得できるような妥当な結論に早く誘導すべきである。

【リスク回避のアクション・チェックリスト】

◎複数ベンダー間で分担が不明確な場合のさばき方

□ ① 中心ベンダーによるプロジェクトの統合管理をすること。
□ ② 準備段階における各社の責任および役割の明確化を行うこと。
□ ③ 自分だけの力では不足なら上司の力も借りること。
□ ④ 関係各社の本当のキーマンは誰なのかを早い時点で見抜いておくこと。

事例6.　不適切な事前着手

◎体制の事前確保が仇に！

【事象概要】

契約に向けて顧客側と折衝を続けていたが、金額面がなかなか折り合わなかった。顧客側から要求されている納期が非常に短く、サービスインスケジュールを確保するためにベンダー側で要員を確保した。要員がそろったため作業を進めていたが、いざ契約の段階で開発対象機能を削減されてしまい、それまでに進めていた基本設計作業は契約範囲外ということでなかったことにされ、契約未締結期間の要員コストはベンダー側の持ち出しとなり、赤字プロジェクトになってしまった。[SEC]

【判断の誤り】

プロジェクトを進める上で必要な要員をすぐに準備できるわけではない。契約を締結してから体制作りや要員調達をしていては、短納期プロジェクトは難しい。そのため、契約は未締結であったが、金額交渉に問題はないと判断し、基本設計に着手してしまった。[SEC]

【発生問題】 損益悪化

【リスク要因】

キーリスクは、④組織能力不足（マネジメントリスク）にある。特に非科学的な情緒的思考や戦略の欠如（問題に対する取り組みの姿勢）がプロジェクトを失敗に導いている。

関連リスクは次の通り。

⑦ リーダーのプロジェクトマネジメント能力（マネジメントリスク）

⑨ プロセス管理の有無（マネジメントリスク）

⑰ あいまいなスコープ。目的・開発範囲の不明確さ・複雑さ。

【リスク回避策】

短納期に対応すべく契約前に事前着手を行ったが無駄になってしまった。

現実には納期が間に合いそうにもない場合は事前着手と称して開発指示書（契約書）がなくても開発行為を開始することが多い。しかしながらこのようなプロジェクトにおいてはQCDのいずれかまたは全てに目標未達の結果が多い。

契約書が成立していないと言うことは、開発費が決まっていない、開発期間が決まっていない、仕様が決まっていない、等々ビジネスの基本要件が成立していない状態なのだ。このような状態では開発体制も未整備なままでお金や時間の管理も不十分なことが多い。このようなプロジェクトにおいては、もう少し時間がたてば何とか改善されるだろうとの甘い認識のまま、結局要件定義工程はずるずると遅延し仕様は肥大化し開発費は不足したままでプロジェクトは混乱状態に陥ってしまうことが多い。納期を守るためとか仕事を取るためとかの理由だけで開発着手を急ぐことは誤りだ。

近年，ユーザー側からRFP（要件定義書）等がベンダー側に提示されるケースも増加しており日本における契約に対する認識はかなり改善して来てはいるものの，背に腹は替えられず何らのリスク回避策も打たないまま事前着手に突入している例がしばしば見受けられる。プロマネとしては条件・契約が早く成立するように，それらの阻害要因の解決に向けて利害関係者間との調整などのタフなネゴシエーションをする必要がある。

【リスク回避のアクション・チェックリスト】

◎事前着手のリスクは自分の負える範囲内で負うこと。

□ ① 基本的には事前着手は不可。

□ ② 事前着手する条件は二つある。

- 自分が損失をかぶるリスクを負える場合。
- 顧客側が損失を共同負担してくれる約束を明文化してくれる場合。

事例7.　不適切な新旧システムの同時並行開発

【事象概要】

新法制度への対応のため、現行システムを、そのシステムを保守しているベンダーがリプレースするというプロジェクト。基本は現行システム機能を踏襲することになっており、新しい法制度の要件を加味したシステムにする必要があった。

ところが現行システムも機能追加・変更が多く、現行システムへの機能追加・変更が、現行踏襲の新システムにも影響し、開発作業が重複するという事態になった。[SEC]

【判断の誤り】

新システムの開発で現行システム踏襲分の仕様を凍結し、新システムの結合テスト後に、それまでの間に現行システムに発生した機能追加分を一括開発することにした。しかしそのため、総合試験環境と開発環境の構築が重複することになり、環境の構築、管理する要員の負荷が増え、環境ミスによる進捗遅れが生じてしまった。[SEC]

【発生問題】 進捗遅延

【リスク要因】

キーリスクは、④組織能力不足（マネジメントリスク）、特に戦略の欠如（問題に対する取り組みの姿勢）にある。

関連リスクは次の通り。

⑦ リーダーのプロジェクトマネジメント能力（マネジメントリスク）
⑭ 開発環境の不備（技術リスク）
⑲ 情報の不備・不足

【リスク回避策】

そもそも企画ミスの疑いが濃厚な事例だ。何で新システムの開発が決まっているのにいつまでも現行システムの開発を続けているのか理解に苦しむ。この顧客もベンダーも何を考えているのだろうか。もう捨てることに決まっている現行システムに次から次へと機能を加えようとしている。金と労力の無駄使いそのものだろう。もしベンダー側がこのことを知っていた上で単に売り上げが増えればこれ幸いなどと考えているのだったら、これは無知な顧客に対する不信行為そのものだ。この世界の一般的な常識からすれば、新システム開発が決定する前の時点ですでに現行システムに対する開発は抑制されるものだ。

この事例に対してのリスク回避策は何かと聞かれたら、ベンダーは新システムの受注が決定した時点で顧客に現行システムの開発を抑制し開発のパワーを新システムに集中することの合理性を進言することだろう。

何らかの他の合理的な理由で新旧同時開発を実行しなければならなかったのなら、旧システムについての開発を最小限にする提案を行うべきであっただろうし、開発量および時期の重複リスクを考慮した開発体制の分離および開発環境の分離などの事前準備が必要だった。

格言にいわく、[二兎を追うものは一兎をも得ず」。

【リスク回避のアクション・チェックリスト】

◎リプレース開発における新旧システム同時開発の危険性

□ ① 旧システムの開発を抑制すること。
□ ② 新旧開発体制を分離すること。
□ ③ 新旧開発環境を分離すること。

事例8.　ブラックボックスの事前検証不足

◎未知のパッケージを協力会社から提案してもらったが、そのパッケージに品質問題が発生

【事象概要】

顧客に対する提案内容として、自社内では、ある中核機能に関する実現手段を持っていなかったため、協力会社に提案してもらったパッケージを利用する方法を採用した。しかし、いざその方法で構築作業に入ったところ、提案されたパッケージに重大な品質不良が発覚した。提案してきた協力会社配下にて、さらにパッケージベンダーとハードウェアベンダーの間で、原因の切り分けができず、対策に時間と費用を要した。[SEC]

【判断の誤り】

中核機能の実装にあたって、協力会社がパッケージを提案してきたが、プロジェクト・マネージャにはパッケージの内容については理解不足で判断しようもなく、協力会社を信じてパッケージの品質には問題がないだろうと判断した。[SEC]

【発生問題】品質不良、進捗遅延

【リスク要因】

キーリスクは、①他者依存的姿勢（マネジメントリスク）、ベンダーにおける協力会社への依存的姿勢にある。

関連リスクは次の通り。

④　組織能力不足（マネジメントリスク）
⑦　リーダーのプロジェクトマネジメント能力（マネジメントリスク）
⑪　関連部署との連携不足（マネジメントリスク）
⑲　情報の不備・不足

【リスク回避策】

これも一種の丸投げ的他者依存的姿勢が生み出した問題なのだ。協力会社のパッケージだから協力会社がなんとかするものだなんて思っているわけだ。中核機能のパッケージについて自分たちは知らないでもいいとでも思っているのだろうか。協力会社を信じてなんて、何と情緒的なのだろうか。ロジックの設計をやっているヒト開発者の発言とは思えない。

中核機能としてのパッケージなのに、それまでの導入実績や障害事例などについて何も調査してなかったようだ。もし殆んど実績がないようだったらそんなもの中核業務用として採用できるわけもない。それでも採用して使ってみたらバグだらけだった。こんなこといつまでもやっていると会社やヒトは確実につぶれてしまう。

【リスク回避のアクション・チェックリスト】

◎未検証のパッケージを中核業務用に採用した結果、重大な品質不良

□　①　下請けのリスクは自分のリスクだという認識を持つこと。
□　②　他社パッケージの採用に当たって、その導入実績・障害履歴の調査を行うこと。
□　③　本格的な採用の前に現物を入手し動作確認のシュミレーションを実施すること。

事例9.　顧客とベンダーによる共同研究開発の落とし穴

◎顧客との共同開発という名目で、イニシアチブが取れずに要件がなかなか確定できない

【事象概要】

医療法人である顧客と、開発受託会社とで費用折半でシステムを共同研究・開発することになった。開発したシステムは、開発受託会社で他の医療法人へのパッケージ展開する計画で顧客とも合意していたが、その顧客にしか適用できないような個別要件が噴出し、歯止めが利かず、設計手直しなどの手戻りが多発。共同開発ということで、開発側もイニシアチブを発揮することができなかった。結果、ローカル仕様ばかりの、他社展開できないようなシステムができあがった。[SEC]

【判断の誤り】

顧客側の仕様確定が遅延していたが、共同研究のメリットを重視し、仕様未確定であるにもかかわらず開発を開始した。[SEC]

【発生問題】 損益悪化

【リスク要因】

キーリスクは、④組織能力不足（マネジメントリスク）、特に未熟な組織文化（非科学的な情緒的思考、認識の不足）と戦略の欠如（問題に対する取り組みの姿勢の弱さ）にある。

関連リスクは次の通り。

① 他者依存的姿勢（マネジメントリスク）
② 上位マネジメントの関与不足（マネジメントリスク）
⑰ あいまいなスコープ。目的・開発範囲の不明確さ・複雑さ。
⑱ あいまいな要件。要件定義の不明確さ。

【リスク回避策】

パッケージ展開する計画だったというわりにはパッケージ仕様の骨子を顧客と十分に討議・研究していないようだ。顧客に対する企画段階からの戦略ミスだと言える。

ベンダー側は顧客側のノウハウを安く手に入れてパッケージ化による甘い汁を吸うというもくろみだったのだろうが、顧客の方が一枚上手で、ほとんど顧客要件のカスタマイズ開発となり、少ない開発費で新システムを手に入れた結果となっただけの話なのだ。ベンダー側は名目だけの共同開発という名前に踊らされただけの間抜けた話だ。

本当の意味で両者が共に成果を享受するためには、顧客およびベンダーの両者はこのプロジェクトの企画段階においてパッケージ化にふさわしい機能要件は何かということについて、いいかえると医療業界に広く受け入れられるデファクト・スタンダードになりえる機能要件は何かということについて真剣に研究・討議すべきだった。その結果顧客の希望する要件が全く顧客依存型の要件しか受け入れられないというのであれば、企画段階でこの共同研究開発はキャンセルすべきものだった。

【リスク回避のアクション・チェックリスト】

◎顧客とベンダーによる共同研究開発の落とし穴

□ ① 顧客との共同研究開発によるパッケージ開発のポイントは企画段階において業界に広く受け入れられるデファクト・スタンダードになりえる機能要件は何かということについて真剣に研究・討議した結果に基づいて開発を行うことにある。

事例10.　プロマネ人材の選任ミス

◎プロジェクト・マネージャに大規模プロジェクトの経験がなく、結合テストの頃から調整不足が露呈した

【事象概要】

大規模プロジェクトを受注したが、プロジェクト・マネージャを任命する責任者に大規模プロジェクトの経験がなく、そのプロジェクト・マネージャが適任かどうかを判断する能力がなかった。要件定義から設計、プログラミング、サブシステム内結合テストまでは、計画通りに順調に進んでいるように見えていた。しかし、サブシステム間テストの段階で、結合テストの進め方やテストデータの準備を誰がするのかといった調整不足が露呈し、テスト作業が停滞してしまった。[SEC]

【判断の誤り】

責任者が、大規模プロジェクトをマネジメントできるスキルを確認せずに、プロジェクト・マネージャを任命した。[SEC]

【発生問題】 進捗遅延、品質不良

【リスク要因】

キーリスクは、④組織能力不足（マネジメントリスク）および②上位マネジメントの関与不足（マネジメントリスク）にある。

関連リスクは次の通り。

① 他者依存的姿勢（マネジメントリスク）
② 上位マネジメントの関与不足（マネジメントリスク）
⑦ リーダーのプロジェクトマネジメント能力（マネジメントリスク）
⑨ プロセス管理の有無（マネジメントリスク）
⑩ コミュニケーション能力（阻害・組織間ないしは人間間のギャップ）（マネジメントリスク）

【リスク回避策】

未熟なプロマネをアサインした上司が問題の原因のように書いてあるが、みんなそれを横で見ていただけなのだろうか。これもまた無責任なことなのだ。この開発組織は自律性のかけらもないいいかげんな人間の集団なのか。

また任命されたプロマネにおいても自分の手に余るような問題に直面したときに傷が深くならないうちに支援のSOSを適時発信すべきだった。だれしも未熟な間は失敗もするが、傷を深くしない方法は、できること／できないことを自分ではっきりと自覚し、できないことはできる人に頼むか教わるかをすることだ。一番悪いのは問題を放置することだ。

【リスク回避のアクション・チェックリスト】

◎未熟なプロマネの失敗

□ ① 適時のSOSは恥ではないこと。
□ ② 任命した上司は未熟なプロマネに対するコミュニケーションを絶やさないこと。SOSに対しては適切なレスキューを行うこと。
□ ③ 他のプロジェクトの先輩プロマネは他人事と看過せず適時のアドバイスを心がけること。

事例11．　システムの劣化あるいはスパゲッティ化について

◎仕様変更による「システム老化」

【事象概要】

さみだれ式に仕様・変更を受け、即応し続けた。その後、機能拡充段階でも同様の開発を続けてゆくうちに、当初の簡明なモジュール構造がスパゲッティ状態となっていった。高いソフト改修コスト、改造に伴う品質劣化、開発期間増大が顕著になり、システムの老化を早めた。顧客は「作りが悪い」とベンダーを攻めた。[SEC]

【判断の誤り】

仕様変更によるシステム構造の複雑化は、受注側の事業面ではマイナスにならず、顧客だけの責任だと見切った。[SEC]

【発生問題】 システム構造の複雑化による、ソフトウェア・メンテナンス性の低下、品質不良

【リスク要因】

キーリスクは、④組織能力不足（マネジメントリスク）にある。

関連リスクは次の通り。

⑯　資源投入戦略の誤り・開発費投入時期ミス（マネジメントリスク）

⑲　情報の不備・不足

【リスク回避策】

経年変化により，システムはまずハードウエアから劣化が始まり、ソフトウェアの動作環境劣化を招く。次にソフトウェアについては継続的な機能改修・追加によりその健全性は初期導入時に比べて必ず劣化していき，所定の性能および品質は確実に悪化していく。劣化の大きさおよび早さはその改修規模の大きさ，複雑さおよび回数に比例するものだ。だから長期稼動が予定されているソフトウェアシステムにおいては適時にリファクタリング等の再生・再構成・改修等の実施が必須である。 ソフトウェアが劣化しないと言うのは迷信なのだ。

【リスク回避のアクション・チェックリスト】

◎システムの老化・劣化あるいはスパゲッティ化の防止について

□　①　更新回数・規模が大きいソフトウェアシステムは一定期間ごとにリファクタリングを行う必要がある。ソフトウェア構造の複雑化の解消、矛盾したロジックの解消、レスポンス・パフォーマンス性能の改善を定期的に実行する必要がある。

事例12.　開発ガイドラインの不備

◎開発ルール未確定のため保守性、流用性が悪い

【事象概要】

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にくく、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。[SEC]

【判断の誤り】

納期を優先したため、とりあえず現状維持。後日、修正、変更時に徐々にプログラムを手直し、追加変更、移植性の良いものに改善させた。[SEC]

【発生問題】 開発効率の悪化

【リスク要因】

キーリスクは、⑧メンバーの技術能力（技術リスク）、⑫ドキュメント（手順書など）の不備（技術リスク）、⑲情報の不備・不足（技術情報の貧困さ）にある。

【リスク回避策】

設計手順書、コーディング規約、結合テスト手順書、総合テスト手順書などは設計・製造・評価などにおけるノウハウ集でありチームにおける設計・製造・評価の思想および手法のガイドラインです。これらのガイドラインは設計・製造・評価における業務品質の均一化および業務の効率化の実現には必須のものです。チームの技術者が各自勝手なやり方で設計・製造・評価した物の整合性をとることは殆んど不可能です。これらの手順書や規約などのガイドラインの整備は開発チームの必携のドキュメントだと言えます。もしこれらのガイドラインが無いようならば整備を急ぐ必要があります。

【リスク回避のアクション・チェックリスト】

◎開発必携のガイドライン

☐ ①　開発プロセスガイドラインを整備すること。
☐ ②　設計手順書を整備すること。
☐ ③　コーディング規約を整備すること。
☐ ④　結合・総合テスト手順書を整備すること。

事例13.　新言語採用にあたっての準備不足

◎初物プログラム言語での開発で十分な性能が出せない

【事象概要】

ユーザ側から、まだ市場に出て日が浅いプログラム言語を使っての開発を指定された。その顧客はそのベンダー側にとっては重要な顧客であったため、どうしても受注したかった。そのプログラム言語での実装を行ったが、十分な性能要件を満たすことができなかった。[SEC]

【判断の誤り】

どうしても受注するという営業判断もあったため、自分たちで経験していない新しいプログラム言語についての見極めができていないまま、プロジェクトをスタートさせた。[SEC]

【発生問題】 パフォーマンス性能不良

【リスク要因】

キーリスクは、新言語の経験が浅い技術者ではなく、⑦リーダーのプロジェクトマネジメント能力（マネジメントリスク）にある。有効な事前準備も行わず開発を開始するようなプロマネはまちがいなくプロジェクトを失敗させる。

関連リスクは次の通り。

④ 組織能力不足（マネジメントリスク）
⑧ メンバーの技術能力（技術リスク）、うっかりミスなどのヒューマンエラー（マネジメントリスク）
⑬ 開発のベースの有無（技術リスク）
⑭ 開発環境の不備（技術リスク）
⑲ 情報の不備・不足

【リスク回避策】

経験の浅い言語の採用にあたっては、新言語の経験者を他社からでも採用するとか、事前の技術情報の収集やプロトタイプによる学習などの事前準備が必要となる。なぜそのようにしなかったのか。このリーダーの目線は顧客から得られる利益しか見ていないようだ。顧客価値の実現とか良い製品をどのように実現するとかの目線が全く感じられない。それでは仕事がうまくいくはずもない。常識力の欠如としか言いようがない。

【リスク回避のアクション・チェックリスト】

◎新しい言語の使用にあたって

新言語の具体的な特徴についての事前情報の収集および試用は必須となる。

☐ ① 自分がやろうとしていることは常識の延長線上にあるか。
☐ ②新言語経験者の採用。
☐ ③ 新言語の特長・欠点の把握。
☐ ④ 新言語の制約・条件の把握。
☐ ⑤ 新言語由来の不具合情報の収集。
☐ ⑥ 小規模のものから試用する。

事例14.　未経験分野の開発

◎経験の少ないオープン系システムで、プロジェクト・マネージャがコントロール不能に陥る

【事象概要】

未経験のオープン・システム基盤を用いたシステム開発案件で、顧客にとって非常に重要であり、セキュリティ要件や性能要件などコントロールの難しい要素が多数あるプロジェクトであった。

ベンダーの経験不足・知識不足もあり、安易な体制でスタートしたため、プロジェクト・マネージャは、あがって来る懸案事項や日々のタスクを処理しきれない状態になり、統括できなくなってしまった。[SEC]

【判断の誤り】

受注前に十分調査をせず、マネジメントの問題が発生するだろうと想定することもなく、安易なプロジェクト体制で受注した。[SEC]

【発生問題】 プロジェクトマネジメント制御不能状態

【リスク要因】

キーリスクは、②上位マネジメントの関与不足（マネジメントリスク）および④組織能力不足（マネジメントリスク）にある。

関連リスクは次の通り。

⑦ リーダーのプロジェクトマネジメント能力（マネジメントリスク）

⑲ 情報の不備・不足

【リスク回避策】

この問題は担当プロマネだけの責任ではない。若干の責任があるとすれば、上司に対しオープン・システムの知識を持つ人材をサブリーダーとしてつけてくれと言わなかったことだろう。この担当プロマネに何のサポートもつけなかった上司が、マネジメントとして無能だったということだ。この開発でプロマネを一人つぶした罪は重い。たぶんその後、この会社では上司は元気で担当プロマネは病気になったか辞めたかだろう。不条理なことだ。

【リスク回避のアクション・チェックリスト】

◎未経験分野の開発

□ ① 自分がやろうとしていることは常識の延長線上にあるか。

□ ② 未経験分野の開発には技術調査やトレーニングなどの十分な事前準備と経験人材の確保が必須。

事例15.　開発体制の不備

◎DBMS の技術的な問題で対応できない

【事象概要】

DBMS に不慣れな開発要員だけでアプリケーションを開発。DB 操作のレスポンスが遅すぎて、システムとして使い物にならなかった。[SEC]

【判断の誤り】

IT アーキテクト要員の不足。本来なら利用技術が確定したところで必要な技術者をアサインし、技術グループを構成するところだが、それをしないで進めた場合に陥りやすい問題。[SEC]

【発生問題】 レスポンス性能不良

【リスク要因】

キーリスクは、⑦ リーダーのプロジェクトマネジメント能力（マネジメントリスク）、特にヒト・資金の資源投入時期・質量のミスにある。

関連リスクは次の通り。

⑧ メンバーの技術能力（技術リスク）、うっかりミスなどのヒューマンエラー（マネジメントリスク）

⑲ 情報の不備・不足

【リスク回避策】

設計ミスというよりかスキルのない人材による体制を決めたプロマネのマネジメントミスだ。DBMSのアプリケーションは通常の業務アプリケーションに加えてDBMS技術に精通した開発者が必須の領域だ。DBを知らないアプリケーション開発者をいくら投入しても性能達成はできない。蒸気機関車の運転手に電気機関車の運転ができるわけもないのと同じで、この問題もプロマネにおける一般常識力の欠如が問題の真因だったと言える。

【リスク回避のアクション・チェックリスト】

◎常識力の獲得と開発体制の不備

□ ① 自分がやろうとしていることは常識の延長線上にあるか。

□ ② 開発にあたっての基本的な技術要件を明確に認識し、適合する開発者で体制を構築すること。

事例16.　新技術採用における準備不足

◎最新技術てんこ盛りで、納期も短いシステム構築で手戻り多発

【事象概要】

重要顧客の記念事業として、最新技術をふんだんに盛り込んだ社内システムを計画していたが、技術的に困難が予想され、しかも短納期で開発する必要があった。プロトタイピングなどあらかじめ検証する工数や工期も取れなかった。設計～製造フェーズに入り、実現できない機能が多発し、一部は要件定義に手戻りが生じた。結果、工数増大し不採算プロジェクトとなった。[SEC]

【判断の誤り】

実現性、生産性、性能など未知の領域を残したままだったが、顧客側の押しも強く、なんとかなるだろうと思い開発に着手した。[SEC]

【発生問題】 損益悪化、納期遅延、品質不良

【リスク要因】

キーリスクは、④組織能力不足（マネジメントリスク）、特に未熟な組織文化（非科学的な情緒的思考、情報軽視、データ軽視、認識力の弱さ）と⑦リーダーのプロジェクトマネジメント能力（マネジメントリスク）、特に見える化能力（見えない問題の顕在化能力）の不足にある。

関連リスクは下記次の通り。

⑤　見積り能力（マネジメントリスク）
⑩　コミュニケーション能力（阻害・組織間ないしは人間間のギャップ）（マネジメントリスク）
⑬　開発のベースの有無（技術リスク）
⑭　開発環境の不備（技術リスク）
⑮　開発費不足（マネジメントリスク）
⑲　情報の不備・不足

【リスク回避策】

新技術による開発だと分かっていたのだから、それらの新技術についての技術情報や失敗事例を集めて実際に動かしてみて検証しておくのが常識でしょう。プロトタイピングなどの工数や工期も取れなかったなどの言い訳は何の理由にもならない。絶対に必要な費用は顧客からいただかなければいけないし、何らかの理由でいただけないのなら自社の費用を投入しなければならない。このプロマネは顧客に、”私たちはこの新技術について何も知りませんが、とりあえずお客様の本番開発で試してみます”なんていえるのだろうか。

新技術による開発だと分かっていたのに何も準備せず、必要な開発費獲得の努力もせず、何らの根拠もない何とかなるだろうという情緒的な思考の典型的な事例です。無責任・無能・非常識とはこう言うことを指すのだろう。

【リスク回避のアクション・チェックリスト】

◎新技術採用にあたって

□　①自分がやろうとしていることは常識の延長線上にあるか。
□　②　事前調査・事前検証は必須。
□　③　調査・検証費用は顧客から獲得する努力をすること。自社の開発のベースとなるものなら自社の資金の投入も考えること。

事例17.　新製品開発における準備不足

◎テストエ程で設計仕様変更が多発

【事象概要】

総合テスト工程に入り、画面設計、業務仕様、出力帳票などで設計変更する個所が多発し、納期遅れにつながった。[SEC]

【判断の誤り】

・Web ベースの新製品の開発で、新しいプロダクトの技術的な特徴、性能、制約条件などを正確・的確に把握するのが困難だった。

・業務仕様においては、ユーザー要件の検討項目を膨らませ過ぎて仕様凍結が遅れ、仕様変更がテスト工程まで発生した。[SEC]

【発生問題】 納期遅延

【リスク要因】

キーリスクは、⑧メンバーの技術能力（技術リスク）と⑥要件定義能力（技術リスク）にある。

関連リスクは次の通り。

⑦ リーダーのプロジェクトマネジメント能力（マネジメントリスク）

⑨ プロセス管理の有無（マネジメントリスク）

⑲ 情報の不備・不足

【リスク回避策】

新製品の開発において新しいソフトウェアモジュールなどの採用が必要な場合、これらの新技術についての事前調査やプロトタイプの作成による実地検証が成功のポイントになる。事例のようにとりあえず始めてから考えてみようという情緒的思考では必ず失敗する。

【リスク回避のアクション・チェックリスト】

◎新製品開発における事前準備

□ ①自分がやろうとしていることは常識の延長線上にあるか。

□ ② 技術調査やトレーニングなどの十分な事前準備を行うこと。

□ ③ プロトタイプによる実地検証の実行。

□ ④ 経験人材を確保すること。

事例18.　不適切な基幹技術選定対応

◎DBMSが詳細設計後に決まって手戻り

【事象概要】

実績、機能、価格の異なる2つのDBMSの何れを使うか顧客が迷っていた。また、マルチベンダーでの開発案件だったが、開発ベンダー間でもそれぞれのDBMSを押す勢力ができてしまい、お互いに譲らない形になった。そのため顧客も更に迷い、DBMS決定に手間取った。詳細設計終盤で顧客が、当プロジェクトの想定DBMSとは別のDBMSに決定。一部詳細設計が無駄になった。[SEC]

【判断の誤り】

プロジェクトの特性や世の中のトレンドを考え、一方の DBMS を使うことになるだろうと見切り、詳細設計に着手した。[SEC]

【発生問題】 損益悪化

【リスク要因】

キーリスクは、①他者依存的姿勢（マネジメントリスク）にある。顧客における自律性の欠如に端を発した問題だが、複数のベンダー間においても自分の主張のみにとらわれており本来の自律性の発揮がなかった。

関連リスクは次の通り。

④　組織能力不足（マネジメントリスク）

⑦　リーダーのプロジェクトマネジメント能力（マネジメントリスク）

【リスク回避策】

DBMS方式の決定は要件定義工程の中では業務要件の決定以上に重要な要件だ。このような基本要件が詳細設計工程の終盤まで決まらなかったのは複数のベンダー各社の怠慢としかいいようがない。基幹技術要件やシステムの基本性能要件は要件定義工程の初期の段階で決定すべきものだ。顧客は一般的にこのような基幹技術について知識がない場合が多く、複数のベンダーが異なった方式を推薦してくると決定を先延ばしするしかなくなってくる。このようなデッドロックに陥らせないためには顧客に対して最も適したDBMSを見つけ出してくれる中立的な技術コンサル会社を紹介することも一つの方法と言える。

一方ベンダー側においてはどの方式であっても影響を最小限に食い止められるような設計を前もって検討しておく必要がある。この事例は顧客が決めるまで待つしかないという自律性のない他者依存の姿勢が招いた当然の結果だ。

【リスク回避のアクション・チェックリスト】

◎基幹技術方式決定遅延が設計手戻りの原因に

□　①　DBMS方式などの決定が遅れそうな状況があれば業務設計部はどの方式になっても対応可能なような考慮をしておくこと。

□　②　基幹技術方式の顧客における決定を促進させる方策をプロジェクトの準備段階から実行しておくこと。例えば中立系の技術コンサル会社の紹介、顧客と同業他社における採用情報や市場シェア情報の提供など。

第3章　要件定義工程におけるリスクと回避策

事例19.　プロマネを見殺しにした上司

◎仕様変更による「品質問題」

【事象概要】

全国オンラインの基幹システム開発で顧客との仕様凍結合意後、大型の機能追加要求が発生した。顧客は競合企業が出した新サービスへの対抗上欠かせない緊急のものであり、対応できないなら、継続発注の保証が無いことを暗に伝えてきた。プロジェクト・マネージャ（PM）は上司に相談したが、効果あるプロジェクト体制強化もできず、顧客との継続取引が上司の事業部門の生命線となっているため、「受入れざるをえない」と言われた。PM は孤軍奮闘したが、結果的にサービス開始後、システム停止、誤課金など重大バグが多発した。[SEC]

【判断の誤り】

顧客の強い要望に対して、誠意あるクイックレスポンスが最も大切と判断。現場 PM は緊急機能追加によるプロジェクト計画ベースライン（QCD）への影響の査定を甘くせざるをえないと考えた。[SEC]

【発生問題】 品質不良

【リスク要因】

キーリスクは、②上位マネジメントの関与不足（マネジメントリスク）にある。ここでいう関与とは効果的な支援のことだ。プロマネの足を引っ張ることや、まして奈落の底に突き落とすことではない。

関連リスクは次の通り。

③　ユーザーの参加・協力度不足（マネジメントリスク）
④　組織能力不足（マネジメントリスク）
⑦　リーダーのプロジェクトマネジメント能力（マネジメントリスク）
⑩　コミュニケーション能力（阻害・組織間ないしは人間間のギャップ）（マネジメントリスク）

【リスク回避策】

顧客との仕様凍結後に出された大型の機能追加要求を何らの条件もなしに受け入れた上位マネジメントの対応ミスが最大の原因と言える。丸投げされたプロマネは沈没するしかないだろう。このような顧客要求丸呑みのやり方が顧客の真の満足を得られる唯一の方法とはいえない。

大型の追加機能が出された時点でベンダー側の上位マネジメントはこの要求を実現するための追加コスト・追加期間などの条件を顧客に提示しタフな交渉を行うべきだった。それ以前の問題としてそれほど重要な顧客ならばベンダーの上位マネジメントは顧客との間のパートナーシップを強力なものにしておく必要があった。何の支援もなく丸投げされたプロマネは最後に品質を犠牲にするしかなかったのだろう。結果として上司が最も恐れていた継続発注は失ったのかも知れない。

【リスク回避のアクション・チェックリスト】

◎プロマネを見殺しにした上司

上司がやるべきことは次の通りです。

□　①自分がやろうとしていることは常識の延長線上にあるか。
□　②　顧客との強力なパートナーシップを確立しておくこと。
□　③　インパクトのある追加要求に対しては条件交渉を行うこと。
□　④　プロマネには体制面での支援を行うこと。

事例20.　上司に対する支援依頼をあきらめたプロマネ

◎仕様変更による「納期遅延」

【事象概要】

仕様変更には普段ガードが固過ぎる PM が、顧客から法制度上欠かせない緊急の機能追加を半強制的に指示された。PM は上司に QCD を守るために待機中であるキーパーソンの参画が必須と考えたが、大型案件の受注拡大を進め、キーパーソンをあてようとしている上司にキーパーソンを要請しても勝算はないとあきらめた。顧客の強制的姿勢に押され、機能追加要求に伴うリスクを伝えることができなかった。PM は現プロジェクト体制に外注要員を加えた機能追加の対策を提案し、上司は PM 提案を了承した。しかし、結局プロジェクトの納期は遅延した。[SEC]

【判断の誤り】

PM は上司に新規受注以外の問題には聞く耳がないと見切った。従って、顧客の仕様変更要求は断り続けることがベストだという信念を持つようになった。上司は、PM が何でも正直にプロジェクトの問題を話してくれるものと信じていた。[SEC]

【発生問題】 納期遅延

【リスク要因】

キーリスクは、⑦リーダーのプロジェクトマネジメント能力（マネジメントリスク）にある。

関連リスクは次の通り。

④ 組織能力不足（マネジメントリスク）

⑩ コミュニケーション能力（阻害・組織間ないしは人間間のギャップ）（マネジメントリスク）

【リスク回避策】

プロマネのミスである。上司が聞く耳があろうがあるまいがプロジェクトが危機に瀕するような要求を受けた時点で上司に支援を要請すべきだろう。キーパーソンの支援が無理かどうか頼んでみなけりゃ分からない。もしだめだったとしても他の有能なメンバーの支援が得られたかも知れない。

上司への支援依頼をあきらめたということは自分のプロジェクトを見限ったということと同じだろう。極端をいうと他のプロジェクトがストップしたとしても自分のプロジェクトは絶対に成功させるというくらいの強い気持ちがプロマネには必要なのだ。

何もしないうちにあきらめるなどという情緒的な思考では到底プロマネとしての責任は果たせないし、結果として顧客にも会社にも迷惑をかけてしまうことになる。

【リスク回避のアクション・チェックリスト】

◎上司に対する支援依頼をあきらめたPM

□ ① 普段から上司とのコミュニケーションを密にし、風通しのよい関係を保っておくこと。

□ ② プロジェクトが危機に瀕しそうになった場合、自分自身の勝手な判断であきらめる前に依頼すべきことは依頼し、主張すべきことは主張すること。

事例21．　問題から逃げたプロマネ

◎開発中のリスクについてプロマネが上司への報告をあきらめた

【事象概要】

プロマネが仕様変更折衝に難航した。上司は官僚的な機能型組織の業務しか経験がなく、良い管理職はルール通りの定期報告を行うプロマネを是とし、問題提起報告で同時に解決策が書けないプロマネを能力不足と評価する傾向があった。プロマネは上司に、問題をエスカレーション（報告）できず、問題が潜在したまま、下流工程でより大きな QCD の問題となって現れた。[SEC]

【判断の誤り】

問題について相談しても解決に向けて前進できると期待できない上司だから,自己解決しようと考えた。[SEC]

【発生問題】 QCD未達成

【リスク要因】

キーリスクは、①他者依存的姿勢（マネジメントリスク）および⑩コミュニケーション能力（阻害・組織間ないしは人間間のギャップ）（マネジメントリスク）にある。

関連リスクは次の通り。

② 上位マネジメントの関与不足（マネジメントリスク）
④ 組織能力不足（マネジメントリスク）
⑦ リーダーのプロジェクトマネジメント能力（マネジメントリスク）

【リスク回避策】

この事例はあきらかにプロマネ自身の問題だ。仕様折衝が難航している状況をなぜ報告できないのか。何の解決案もなしに、「こうなっちゃいました。どうしましょう」なんて上司に相談をもちかけても怒られるにきまっている。自分に解決案がないから上司に相談をするのをあきらめたなんて子供のすることと同じだ。このプロマネの姿勢は、他者依存的姿勢の裏返しなのだ。本当は依存心が強いのに怒られるのを恐れて問題を放置しただけなのだ。

このプロマネは、まず直面している問題点について必死に解決策を考えなければいけない。仕様折衝不調の原因は何なのか、顧客に問題があるのかあるいは自分の能力に問題があるのか、問題はどんな問題なのかなど、真剣に考えれば何らかの案は出てくる。最初から何も考えたくないような依存的姿勢のままならプロマネの資格はない。そこまで真剣に考えた後なら、良い知恵が出なかったとしてもそこまでを含めて上司に相談すればいいだけだ。

事象概要に、問題提起報告で同時に解決策が書けないプロマネを能力不足と評価する傾向があった、との記述は上司に問題があったかのような印象を与えるが、この上司の評価傾向に何にも問題はない。プロマネもマネジャーであるかぎり問題提起に対応した解決策の一つや二つがあって当然だ。それでも怒るようなバカ上司ならじっと耐えて自分のやれる対策を最大限にタンタンと実行していくだけなのだ。

【リスク回避のアクション・チェックリスト】

◎問題から逃げたプロマネ

□ ① 問題から逃げないで問題の現実を直視すること。
□ ② 問題は何かを書き出すこと。
□ ③ 問題の原因はどこにあるのか書き出すこと。
□ ④ 自分でできることは何、自分でできないことは何かについて列挙すること。
□ ⑤ 自分でできるアクションと支援が必要なアクションに分けて必要な行動をとること。

事例22.　大幅な開発費削減の要請

◎当初価格の大幅削減要求を受け付け QCD 問題発生

【事象概要】

仕様が決まり仮発注も受けとっていたが、顧客にて予算削減が必要となったことから、契約額の大幅削減を要請してきた。開発スコープ縮小の余地が無かった。重要な取引先であり、断れずに、要請通りの減額を行った。その後、納期遅れ、品質問題も発生した。[SEC]

【判断の誤り】

顧客が発注額を削減したので、外注要員を減らして、収支を合わせるしかないと見切った。[SEC]

【発生問題】 QCD未達成

【リスク要因】

キーリスクは、②上位マネジメントの関与不足（マネジメントリスク）にある。

関連リスクは次の通り。

④　組織能力不足（マネジメントリスク）

⑦　リーダーのプロジェクトマネジメント能力（マネジメントリスク）

⑩　コミュニケーション能力（阻害・組織間ないしは人間間のギャップ）（マネジメントリスク）

【リスク回避策】

これは問題ではないだろう。開発側として、開発内容はそのままで開発費減額を受け入れた結果、不足開発費の補充がなければ当然のことながら納期・品質に問題が出たというだけのことなのだ。この顧客が本当に自分らの生命線の最重要顧客だったのなら経営戦略上やむなく開発減額を受け入れたとしても製品の品質・納期を守るべく不足開発費に自社の資金を投入することが常識的かつ良心的な対応だろう。

この事例において上位マネジメントやプロマネは何を考えていたのか理解不能だ。開発費の大幅減額受入れが絶対条件だったのなら開発要件の削減要求をするのは当然のことだ。それもいや、あれもいやでは素人の仕事としか思えない。

この案件の最大リスクはこんな対処しかできないこの会社の上司やプロマネ自身がリスクなのだ。

【リスク回避のアクション・チェックリスト】

◎大幅な開発費削減要請に応えるには

□　①自分がやろうとしていることは常識の延長線上にあるか。

□　②　可能な削減金額の提示を行うこと。

□　③　金額削減を実現するための条件を提示すること。

□　　・　開発要件の削減提案

□　　・　1 次開発、2次開発などによる分割リリースの提案。2次開発費は別途見積り。

□　　・　削減金額相当分を別途の案件に上乗せすること。

◎困難な要求への対処法

□・　Yes（受入れの表明）＆But（条件の提示）で対応すること。

事例23.　大幅な納期短縮の要請

◎当初納期の大幅前倒し要求を受け付け QCD 問題発生

【事象概要】

基幹システムで仕様は決まっていたが、顧客より予定納期の大幅な前倒しが要請された。要請通りの納期前倒しを行った後、大きな QCD 問題が発生。顧客のためによかれと思って対応したのに、結果が出せなかった。[SEC]

【判断の誤り】

ITのことは全く分からない顧客の経営トップからの要請であり、受けざるを得ないと判断した。[SEC]

【発生問題】 QCD未達成

【リスク要因】

キーリスクは、④組織能力不足（マネジメントリスク）、特に未熟な組織文化（非科学的な情緒的思考）にある。

関連リスクは次の通り。

② 上位マネジメントの関与不足（マネジメントリスク）

③ ユーザーの参加・協力度不足（マネジメントリスク）

⑦ リーダーのプロジェクトマネジメント能力（マネジメントリスク）

【リスク回避策】

問題は、開発側の責任者が達成できる根拠も見込みもないのに何も条件を付けずに顧客からの大幅な納期短縮を受け入れたことにつきる。要求が顧客のトップから出されようが、ITの分からない相手であろうが、そんなことは関係のないことだ。

大幅な納期短縮の要請はときどきある話で特別なことではない。何らの条件変更もなく開発期間の大幅短縮などできるわけもない。最初の見積り条件の体制と費用のままで開発期間は半分でできますと答えたら、最初の見積りはウソだったということになる。

顧客トップからの納期短縮要請は顧客のビジネス上必須の要件だったのかも知れない。この要請を重く受け止めることは重要なことだが、何の条件変更もなく実現性の検討もなく、ただガンバリマスだけでは顧客に迷惑をかけ開発会社も信用をふくめ利益も失う結果になる場合が多い。

納期の大幅短縮の要請を受けたときにやるべきことは次のようだろう。

① 可能な納期短縮期間を提示すること。

② 納期短縮を実現するための条件を提示すること。

- 開発費も含めた再見積りの実行
- 開発要件のトレードオフの提案
- 1 次開発、2次開発などによる分割リリースの提案

困難な要求や不条理な要求への対処法は次の通りです。

まずYesと応えること。これは相手の要求を受け入れる姿勢の表明です。次にButで困難な受け入れを実現するための条件を提示すること。非常に難しい要求を実現するための黄金律はこの”Yes＆But”による対応しかない。ただし困難ではない事象にこのやり方を使えば自分の評価は大幅に落ちることを知っておいた方がいい。もちろん、どうあがいても自分や自分の会社の実力を総動員しても実現できない要求に対してはNOと答えるか、逆に相手が実現できない条件を提示するしかない。

【リスク回避のアクション・チェックリスト】

◎大幅な納期短縮要請に応えるには

□ ①自分がやろうとしていることは常識の延長線上にあるか。
□ ② 可能な納期短縮期間を提示すること。
□ ③ 納期短縮を実現するための条件を提示すること。
□ ・ 開発費も含めた再見積りの実行
□ ・ 開発要件のトレードオフの提案
□ ・1 次開発、2次開発などによる分割リリースの提案

◎困難な要求への対処法

□ ・ Yes（受入れの表明）＆But（条件の提示）で対応すること。

事例24.　開発者の急遽大量投入問題

◎結合テストで品質不良が発生

【事象概要】

Web アプリケーション・サーバー上の結合テストに着手したが、プログラム設計の誤りによるバグが多発した。[SEC]

【判断の誤り】

詳細設計工程で業務仕様が増加したため、プログラム設計工程から要員を多量に追加した。これらの要員は、対象業務も、そのプロジェクトの開発環境も初めての経験だったうえ、進捗の遅れも重なり、オブジェクト指向によるプログラム設計や新処理方式に関する事前研修が不十分だった。研修不足を補うために有識者のレビューを実施することもなく、途中参加の要員にプログラム設計を任せていた。[SEC]

【発生問題】 品質不良

【リスク要因】

キーリスクは、⑰あいまいなスコープ、⑱あいまいな要件および⑦リーダーのプロジェクトマネジメント能力（マネジメントリスク）、特にヒト・資金の資源投入時期・質量のミスにある。

関連リスクは次の通り。

⑧ メンバーの技術能力（技術リスク）、うっかりミスなどのヒューマンエラー（マネジメントリスク）

【リスク回避策】

この事例の原因は設計ミスではなく要件定義能力の低さに起因した開発体制の準備不足にある。 設計工程で業務仕様が大幅に増加したということは要件定義があいまいなまま設計工程に突入したということだろう。いきなり不慣れな設計者を大量に投入した場合、開発品質を保つことは非常に難しい。

基本的な対処方法は、要件定義工程を適正に制御しながら必要な技術者の準備を早めに行うことだ。万一、準備不足のまま設計工程に突入してしまった場合は未経験の技術者を大量に投入するのではなく、スキルのある技術者を一定の割合で加えることだ。

【リスク回避のアクション・チェックリスト】

◎開発者の急遽大量投入問題

□ ① 緊急対応時であっても未経験の技術者を大量に投入してはいけない。一定の割合で経験者を加えて開発品質の統制を図ること。

事例25.　仕事に対する基本的な認識の欠如

◎変化即応型の開発をスコープ変更ルールが無く請負った

【事象概要】

仕様変更ルール／変更管理会議（PMBOK の CCB）設置は顧客が杓子定規で変化即応の時代に合わないと嫌うので、中流工程を始める前に取り決めなかった。後になって、品質問題が発生した上で、費用増加の責任について、発注側・受注側で「言った／言わない」の応酬となった。[SEC]

【判断の誤り】

追加作業に対して、いきなり金の話しを持ち出す気まずさと、費用の手当てが伴うとの甘い判断があった。[SEC]

【発生問題】 品質不良、損益悪化

【リスク要因】

キーリスクは、⑦リーダーのプロジェクトマネジメント能力不足（マネジメントリスク）にある。

関連リスクは次の通り。

④ 組織能力不足（マネジメントリスク）

⑩ コミュニケーション能力（阻害・組織間ないしは人間間のギャップ）（マネジメントリスク）

⑫ ドキュメント（要件定義書・設計書・チェックリスト・手順書など）の不備（技術リスク）

⑰ あいまいなスコープ。目的・開発範囲の不明確さ・複雑さ。

⑱ あいまいな要件。要件定義の不明確さ。

【リスク回避策】

このプロマネはいくつかの点で基本的な失敗をおかしている。

① 変更管理の重要性に対する認識不足

変更管理は”何をどのように”作るのかを最終的に確定する作業、つまりスコープや要件を明確させるためのプロジェクト活動の基本中の基本の仕事なのだ。顧客がどう思うかによってやったりやらなかったりするようなものではない。変更管理会議の設置とか大上段に構えるから顧客側も警戒する。普通に要件開発の一環として実施すればいいのだ。

② ビジネスの基本が分かっていない

追加仕様に費用が伴うのは当たり前のことだ。気まずいとか何とかいう話ではない。ビジネスとは顧客の求めるものを提供し、それに対する正当な対価をいただくことなのだ。一方、仕様が削減された項目については費用の削減についてチャント顧客に伝えることも忘れてはいけない。開発の仕事はボランティアではない。

③ 顧客との協調関係不足

プロジェクトの準備段階からしっかりとした顧客との間のパートナーシップを作り上げていないようだ。顧客との協調関係を良好にするには密なコミュニケーションが必須だ。顧客が本当に何を欲しがっているのか真剣にヒアリングすることから始めればいい。

顧客とのパートナー意識がしっかりしていれば、気まずいとか言い出しにくいとかの情緒的な思考は出てこないはずだ。

【リスク回避のアクション・チェックリスト】

◎変更管理の阻害要因の排除

□ ① 変更管理の重要性を認識すること（変更管理はQCDに大きな影響を及ぼす）。

□ ② ビジネスの基本の再認識（正当な仕事で正当な対価を得ることは当り前のこと）。

□ ③ 顧客との協調関係を事前準備段階から作り上げておくこと。

顧客との良好な協調関係なくしてはプロジェクトの成功はない。

事例26.　故意による事故

◎仕様通りに開発したが運用障害で事業が長時間停止

【事象概要】

顧客にはシステム／端末の対人間インターフェースの操作性要件や通常日の運用仕様は具体的に決めて貰えた。しかし、システムのライフサイクルを通じた運用についてはあいまいであったが、PM は気づいていても、予算オーバーとなるので指摘しなかった。

カットオーバー後、年跨り時に、本来行うべきファイル切替・再設定などの自動処理機能がシステムとして設計されておらず、また運用者の手動手順も用意されていなかったことから、ファイルの上書きなど重大障害が発生。運用手順マニュアルにはないデータ・リカバリが必要となり、サービス再会まで長時間がかかった。顧客は事業機会を喪失する事態となったので、受託側ベンダーが責められた。

[SEC]

【判断の誤り】

顧客が決めた対人間インターフェースの操作性要件などを忠実に設計に反映したのだから、それで十分だと見切った。[SEC]

【発生問題】 品質不良

【リスク要因】

キーリスクは、④組織能力不足（マネジメントリスク）＊未熟な組織文化（非科学的な情緒的思考）に尽きる。

関連リスクは次の通り。

⑦ リーダーのプロジェクトマネジメント能力（マネジメントリスク）
⑩ コミュニケーション能力（阻害・組織間ないしは人間間のギャップ）（マネジメントリスク）
⑰ あいまいなスコープ。目的・開発範囲の不明確さ・複雑さ。

【リスク回避策】

このベンダーには一言いわせてもらうと”よく言うよ！”なのだ。あきれてあいた口がふさがらないとはこのことだ。事実を知っていてだまっていたプロマネは重罪だ。

”年跨り処理は予算オーバーとなるので指摘しなかった””仕様通り開発した”などと良くも抜けぬけと言えたものだ。”バカ言うんじゃないよ”なのだ。車に例えると予算がなかったのでブレーキは付けませんでしたと言っているのと同じことだ。この顧客はシステムともども年末まで走らされて年が明けたら断崖絶壁から突き落とされたようなものだ。この事例は開発リスク回避のレベルを超えていて犯罪リスクの世界に入るものだろう。損害賠償請求されても仕方のない事例だ。

【リスク回避のアクション・チェックリスト】

◎故意による事故防止

□　故意による事故の防止策はない。普段の仕事ぶりをみて虚言の多い人間をプロジェクト・マネジャーなどの重責に選ばないことくらいしかない。

事例27．　他者依存的姿勢がもたらした要件定義遅延

◎新サービスと非合理的な納期

【事象概要】

金融機関で創立〇〇年の記念日に新サービスを開始することが至上課題だった。新サービスについてはマスコミへの発表もされていたため納期順守が絶対であった。ところが、新サービスということで実現方式も二転三転し、手戻りが発生。カットオーバー前に突貫作業でとりあえず動くレベルの物を納入。しかし、稼働後は突貫作業の影響で品質の問題が多発した。[SEC]

【判断の誤り】

目新しい新サービスということで、実現方式がなかなか決まらなかった。そのため、非機能要件やセキュリティポリシーについてもなかなか決まらず、方式設計が曖昧なままであった。納期まで時間がないこともあって、開発に進めそうなところから着手していかざるを得なかった。[SEC]

【発生問題】 要件定義遅延、品質不良

【リスク要因】

キーリスクは、①他者依存的姿勢（マネジメントリスク）にある。自律性・積極性のなさが期限管理のあいまいさやコミュニケーションの不全を生みプロジェクトを失敗に導いている。

関連リスクは次の通り。

② 上位マネジメントの関与不足（マネジメントリスク）
③ ユーザーの参加・協力度不足（マネジメントリスク）
④ 組織能力不足（マネジメントリスク）
⑥ 要件定義能力（技術リスク）
⑦ リーダーのプロジェクトマネジメント能力（マネジメントリスク）
⑨ プロセス管理の有無（マネジメントリスク）
⑩ コミュニケーション能力（阻害・組織間ないしは人間間のギャップ）（マネジメントリスク）
⑰ あいまいなスコープ。目的・開発範囲の不明確さ・複雑さ。
⑱ あいまいな要件。要件定義の不明確さ。

【リスク回避策】

要件定義・方式設計の遅れが結局開発・評価時間不足を招き稼動時の品質問題の多発となった。 いつまでも要件定義に時間を費やすべきではなかった。要件定義工程に許された時間に関してあらかじめタイムリミットを設定しておき、限度が近づいた時点で開発側から仕様の逆提案を行うか、機能の分割リリースを提案し顧客の承認を得るべきだった。このような提案を通りやすくするためにも顧客とのコミュニケーションをプロジェクトの準備段階から密にし、顧客を早い時点からパートナー化するようにしたいものだ。

とりあえず動けばよいというような考え方は止めたほうがいい。納期を守るために品質を犠牲にすることは止めたがいい。品質を犠牲にすると顧客の信頼を失い、今後のビジネスも失ってしまう可能性が高い。品質かコストかの選択を迫られたらコストを犠牲にするしかない。一見顧客がモタモタしていたように見えるが、本当はプロである自分がモタモタしていたせいでこんなことになったのだから自分が責任を引き受ける方の選択をするのが正しい仕事のやり方だ。こんなことにならないように最初にいったようにタイムリミットの前にサッサとリスクを排除しておくことだ。

【リスク回避のアクション・チェックリスト】

◎要件定義遅延対策

□ ① 顧客との密なコミュニケーションを維持することで顧客を早い段階からパートナー化しておくこと。

□ ② 要件未凍結のリスク排除のタイムリミットをあらかじめ設定しておくこと。

□ ③ リスク排除のタイムリミットがきたら仕様提案、機能分割リリースなどの提案アクションを直ちにとること。

事例28.　プロジェクトマネジメント能力不足による要件定義遅延

◎仕様凍結確認せず中流工程に着手

【事象概要】

大規模基幹系システムで要件定義、機能設計が遅れ、予定納期までの工期がその分、短縮された。仕様凍結確認のないまま、中流工程に着手。その後、大きな手戻りが発生し、大きなQCD 問題が発生。顧客のためによかれと思って先行着手したのに、結果が出せなかったことで、悪者扱いにされた。[SEC]

【判断の誤り】

顧客は対外的にサービス開始時期を公表しているため、仕様凍結確認のないまま、中流工程に着手せざるを得ないと判断。[SEC]

【発生問題】 要件定義遅延、QCD未達成

【リスク要因】

キーリスクは、⑦リーダーのプロジェクトマネジメント能力（マネジメントリスク）、特にタイムマネジメント能力および⑨プロセス管理の有無（マネジメントリスク）にある。

関連リスクは次の通り。

④ 組織能力不足（マネジメントリスク）

⑰ あいまいなスコープ。目的・開発範囲の不明確さ・複雑さ。

⑱ あいまいな要件。要件定義の不明確さ。

【リスク回避策】

この事例における問題点を時系列順にならべると次のようになるだろう。

① 要件定義が遅延した。

② 時間がなくなったので仕様凍結を待たずに要件定義の途中から開発・製造に着手した。

③ その後顧客から出された変更・追加の要件定義の内容とすでに着手した設計・製造の内容が大きく違ってしまった。

④ 設計・製造の大幅なやり直しを行うことになった。

⑤ 結果としてQCD問題が多発した。

仕様凍結遅れが設計遅れを招き、仕様のあいまいさが更に設計の手戻りを招き、開発に必要な時間が確保できなくなり各工程の成果物の品質は著しく低下し、その結果評価工程で品質問題が多発し、QCD全て未達成の大失敗プロジェクトの典型的なパターンなのだ。

この事例の大きなミスは次の二つだ。

① 要件定義の遅延を防げなかったこと。

② 顧客に仕様凍結の確認を取らないまま設計・製造を開始してしまったこと。

他の同類の事例でも指摘したが下記の対策を実施すれば問題は防げるだろう。

【リスク回避のアクション・チェックリスト】

◎要件定義遅延対策

□ ① 顧客との密なコミュニケーションを維持することで顧客を早い段階からパートナー化しておくこと。

□ ② 要件未凍結のリスク排除のタイムリミットをあらかじめ設定しておくこと。

□ ③ リスク排除のタイムリミットがきたら仕様提案、機能分割リリースなどの提案アクションを直ちにとること。

事例29.　要件定義能力不足による要件定義遅延

◎基本設計の承認なしで詳細設計に着手

【事象概要】

顧客がビルを移転するとともに古いシステムを廃棄することになっていたため、新ビルでの新システム稼働が絶対条件であり、サービスインスケジュールになんとしてでも間に合わせる必要があった。要件定義工程が大幅に遅れたため基本設計を顧客に提示後、承認を得ないまま、詳細設計まで進行。途中、顧客から基本設計の変更を要請され、詳細設計をやり直すという二度手間になり、遅れたスケジュールを取り戻すため、要員を追加することとなってしまった。[SEC]

【判断の誤り】

要件定義に時間がかかったが、その分、システムの実装の姿が見えていたつもりだった。顧客側もだいぶ焦っており、基本設計書を顧客に提出したが、もともと承認されるまでに時間のかかる顧客だったので、承認未完了のままで詳細設計に着手した。[SEC]

【発生問題】 要件定義遅延、損益悪化、進捗遅延

【リスク要因】

キーリスクは、⑥要件定義能力の不足（技術リスク）にある。これにリーダーの外部交渉力の弱さもあいまって要件定義の承認も基本設計の承認も得ないまま詳細設計に突入しプロジェクトを失敗させたのだ。

関連リスクは次の通り。

⑦ リーダーのプロジェクトマネジメント能力（マネジメントリスク）　＊外部交渉能力
⑨ プロセス管理の有無（マネジメントリスク）
⑩ コミュニケーション能力（阻害・組織間ないしは人間間のギャップ）（マネジメントリスク）
⑰ あいまいなスコープ。目的・開発範囲の不明確さ・複雑さ。

【リスク回避策】

表面的には、基本設計について顧客未承認のまま詳細設計にすすみ、結局基本設計も詳細設計もやり直しとなってしまったのだが、これは設計工程のプロセス無視に原因があるというわけではなく、要件定義工程の時間管理の失敗が設計工程のプロセス無視を誘発したのだ。顧客から基本設計の変更を要求されたということは要件定義の精度も悪かったということだ。

要件定義工程においては公式のタイムリミットのはるか前にその進捗度や内容精度レベルについての見極めが必須だ。要件定義工程の１／３から１／２工程あたりで見極めレビューを必ず実施する必要がある。

これではダメだと判断したらすぐに対策アクションを実施しなければならない。対策は次のようにいろいろある。

- 要件定義メンバーを増強すること。
- 顧客と一緒に要件の骨子をレビューし顧客価値の順にプライオリティ付けを行い。優先度の低い要件はドロップするか二次開発項目とすること。
- 顧客側の要件有識者を増強すること。
- 開発内容を分割リリースにすること。
- リリース時期を延ばすこと。

要件工程における失敗、要件工程の大幅遅延・あいまいなスコープやあいまいな要求内容はこの後の工程を全て破壊してしまうという共通認識が必要だ。

【リスク回避のアクション・チェックリスト】

◎要件定義工程では適時のレビューが必須

- □ ① 要件工程の1／3から1／2工程あたりで見極めレビューを必ず実施すること。
- □ ② ダメだと判断したらすぐに対策アクションを実施すること。
- □ ・ 要件定義メンバーを増強すること。
- □ ・ 顧客と一緒に要件の骨子をレビューし顧客価値の順にプライオリティ付けを行い。優先度の低い要件はドロップするか二次開発項目とすること。
- □ ・ 顧客側の要件有識者を増強すること。
- □ ・ 開発内容を分割リリースにすること。
- □ ・ リリース時期を延ばすこと。

事例30.　人選ミスによる要件定義遅延

◎業務経験・業務知識の乏しい要員で要件定義工程を実施

【事象概要】

要件定義に必要な要員の人数は確保できたので、プロジェクトをスタートした。担当者に業務知識がなく、要件定義は日を追うごとに遅れていき、成果物の品質もきわめて低いものとなってしまった。[SEC]

【判断の誤り】

業務経験・業務知識があるかどうかを特に確認しないまま、確保した要員ばかりで、要件定義作業に着手した。[SEC]

【発生問題】 要件定義遅延、品質不良

【リスク要因】

キーリスクは、⑦リーダーのプロジェクトマネジメント能力不足（マネジメントリスク）にある。開発の仕事は基本的に人材によって品質が決まるということが分かっていないのだ。要件定義に能力のない人間を何人投入してもカネと時間の無駄使いなのだ。こんなリーダを選んだ上長が最も悪いのかも知れない。

関連リスクは次の通り。

④　組織能力不足（マネジメントリスク）
⑥　要件定義能力（技術リスク）
⑧　メンバーの技術能力（技術リスク）、うっかりミスなどのヒューマンエラー（マネジメントリスク）

【リスク回避策】

スキルのない人材で要件定義を実施したが結局低品質の結果となった。当たり前のことだ。このプロマネは何を考えていたのだろうか。例えば飛行機を作る仕様をまとめるのを自転車の技術者をあてたようなものだろう。このリーダーはモノもヒトも見る能力がなかったのだ。よく今までリーダーが務まったものだ。

【リスク回避のアクション・チェックリスト】

◎スキル不足のメンバーによる要件定義

□　①　適材適所のできない人間にプロマネをやらせてはいけない。このプロマネを配置した上位マネジメントの責任は重大だ。

□　②　開発の仕事はいわば精密機械を組み立てるような繊細な仕事であると認識すること。開発は単なる肉体労働ではない。人数合わせでやれるほど簡単な仕事ではない。

事例31．　要件定義の優先順位の誤り

◎仕様検討に時間を要し、十分なテストができないまま顧客受け入れテストに突入

【事象概要】

短工期を求められていた上に、仕様検討に時間を要した。仕様の確定した機能から順次開発に着手したが、工期内でテストが十分できていない機能があった。 工期を守ることを優先し、その状態でリリースしたため、顧客の受け入れテストで仕様認識違いを含めて不具合が多発した。[SEC]

【判断の誤り】

多少の品質の悪さは、リリース後の受け入れテスト時点での修正でなんとか対応できると考えて、工期を優先してリリースしてしまった。[SEC]

【発生問題】 要件定義遅延、品質不良

【リスク要因】

キーリスクは、⑦リーダーのプロジェクトマネジメント能力だ。仕事の優先順位についての認識不足なのだ。

関連リスクは次の通り。

④　組織能力不足（マネジメントリスク）
⑥　要件定義能力（技術リスク）
⑨　プロセス管理の有無（マネジメントリスク）
⑫　ドキュメント（要件定義書・設計書・チェックリスト・手順書など）の不備（技術リスク）
⑰　あいまいなスコープ。目的・開発範囲の不明確さ・複雑さ。
⑱　あいまいな要件。要件定義の不明確さ。

【リスク回避策】

そもそも仕様の確定した機能から順次開発に着手するとの発想が間違っている。どうせ仕様が確定しやすい簡単な機能から着手したのだろう。開発の優先順位について何も認識がないようだ。最初の要件定義でまず何をやるべきかというと、顧客が提示した複数の開発テーマを顧客の価値の優先度順に並べることから始めることだ。つまり顧客が一番欲しがっている機能から順に詳細の要件定義を始めるということだ。要件開発の難易度とは無関係だ。開発着手はこの優先度が高く先に詳細要件が固まったものから実行することだ。

このような開発アプローチをとれば主要機能においての顧客との仕様ずれや不具合の発生する頻度は格段に低いものになる。この事例のように開発時間がなくなったから適当にテストをはしょってほとんど検証もしていないような傷だらけの半製品を客先に出すハメになり、問題を多発させてしまうのだ。

大体このプロマネの”多少の悪さは客先の受入れテストの時に直せばいい”という認識はとんでもないことだ。顧客は開発の実験場ではない。顧客の受入れ検査の目的は発注した機能が希望とおりに作られているかどうかの検証なのであって、バグ出しの仕事を手伝っているわけではない。

【リスク回避のアクション・チェックリスト】

◎要件定義の優先順位の誤り

☐　①　要件定義は顧客価値の高い順に実行すること。
☐　②　開発の着手も、先に要件が決定した顧客価値の高い要件から実行すること。
☐　③　客先検証はバグ出しの場ではない。

事例32.　待ちの姿勢による要件定義遅延

◎顧客側メンバーの体制が弱く、仕様が詰められない

【事象概要】

顧客側メンバーは、システム化の経験が浅く、現場部門に対する仕様決定権も弱いので、仕様確定には計画より期間がかかる可能性があったが、ベンダー側の戦略上受注する必要があった。プロジェクトが始まると、顧客側メンバーが多忙との理由により、仕様検討の稼働や時間が十分に取れず仕様確定が遅れ始めた。顧客側の要件の提出期限をベンダー側から提示。顧客は、遅れをとりもどすためメンバーの増強を行なった。 また、スケジュールを延期することとなり、当初予定していた納期は守れなかった。[SEC]

【判断の誤り】

当初ベンダー側としては、顧客側メンバーのスキル不足を補う形で、ベンダー側で作成した仕様案をもとに打ち合わせすることにより、仕様確定をスムーズに行い計画通りに納めることができると考えて受注した。[SEC]

【発生問題】 要件定義遅延、納期遅延

【リスク要因】

キーリスクは、①他者依存的姿勢（マネジメントリスク）にある。

関連リスクは次の通り。

③ ユーザーの参加・協力度不足（マネジメントリスク）
④ 組織能力不足（マネジメントリスク）
⑥ 要件定義能力（技術リスク）
⑦ リーダーのプロジェクトマネジメント能力（マネジメントリスク）
⑨ プロセス管理の有無（マネジメントリスク）
⑩ コミュニケーション能力（阻害・組織間ないしは人間間のギャップ）（マネジメントリスク）
⑰ あいまいなスコープ。目的・開発範囲の不明確さ・複雑さ。
⑱ あいまいな要件。要件定義の不明確さ。

【リスク回避策】

この事例もまた、要件定義工程における開発側の待ちの姿勢にある。開発側で作成した仕様案で仕様確定をスムーズに行おうと考えていたとのことだが結局そのようにはしなかったのだろう。 顧客側の経験が浅いので時間がかかることが分かっていたのに何もしなかった。一体これって何だろう。要件定義は顧客の仕事だからわざわざめんどうくさいことを引き受けたくないとでもいいたいのだろうか。それで何も不都合なことがおきなければそれでも良いだろうが、このパターンの結果はいつもQCDすべてが失敗で顧客側もベンダー側も大きな被害をこうむる結果になる。 要件定義の早めの段階でこれらの問題を認識したら、開発側主導で強く顧客にアプローチするしかない。これも開発の仕事の内なのだ。開発側は他者依存的な姿勢ではなく自律的な行動をする必要がある。それが開発という名に値する仕事だと言える。

【リスク回避のアクション・チェックリスト】

◎要件定義工程での待ちの姿勢は最悪の状況を招く

□ ① 顧客の要件定義力が弱い場合、開発側からの積極的アプローチで要件の掘り起こしおよび提案を行うこと。
□ ② 顧客との密なコミュニケーションを維持することで顧客を早い段階からパートナー化しておくこと。
□ ③ 要件未凍結のリスク排除のタイムリミットをあらかじめ設定しておくこと。
□ ④ リスク排除のタイムリミットがきたら仕様提案、機能分割リリースなどの提案アクションを直ちにとること。

事例33.　アジャイルの理解不足による要件定義遅延

◎顧客の体制が弱いときは、仕様追加、変更が多発する可能性が大

【事象概要】

業務的にシステム化範囲が広いシステムの開発を受注。発注側の仕切り能力が弱く、業務内容の定義がきちんとできないことが懸念された。それぞれのイテレーションごとに追加要求や仕様変更が頻発し、規模がどんどん膨れた。進捗も遅れだし、品質にも問題が出始めた。規模の増大に伴って試験規模も大幅に必要となり、結果的に工数が大幅超過となってしまった。[SEC]

【判断の誤り】

顧客要件をきちんと抽出することで構築後に大幅な手戻りをなくすことができるだろうと考え、4回のイテレーションを回すスパイラル型で開発を進めることにした。[SEC]

【発生問題】 要件定義遅延、損益悪化、進捗遅延

【リスク要因】

キーリスクは、④組織能力不足（マネジメントリスク）と⑦リーダーのプロジェクトマネジメント能力（マネジメントリスク）にある。

関連リスクは次の通り。

⑥　要件定義能力（技術リスク）

⑩　コミュニケーション能力（阻害・組織間ないしは人間間のギャップ）（マネジメントリスク）

⑰　あいまいなスコープ。目的・開発範囲の不明確さ・複雑さ。

⑱　あいまいな要件。要件定義の不明確さ。

【リスク回避策】

アジャイルであろうとウォーターフォールであろうと要件定義および開発着手の順番は顧客価値の高さ、つまり顧客の要求度の高い順に実行していくのが鉄則だ。この事例ではアジャイルをどういうつもりで採用しているのか分からないが、形式だけで全くアジャイルの根本思想が分かっていないように見える。イテレーションとかスプリントとかスパイラルだとかの形だけまねても何も良くならない。 アジャイルは顧客価値の高い要件から順に仕様を決め開発を行う方式なのだ。発注側の仕切り能力が低ければベンダー側で積極的に顧客の要件を掘り起こし共に協調しながら開発をすすめなければならない。

【リスク回避のアクション・チェックリスト】

◎ツボは顧客価値優先度順の開発

□　①　要件定義も含め、開発の実行順は顧客価値の高い要件から実行すること。
　　単に要件が決まったものからとかやさしい要件から着手する考え方はうまくいかない。

事例34.　要件定義書未作成による要件定義遅延

◎正式要件定義書無く作業。仕様あいまいにより変更多発。

【事象概要】

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件定義書がなく、また、ベンダーも強く要求していなかった。その結果、些細な点で要件定義のズレが発生。完成度の低いシステムとなった。[SEC]

【判断の誤り】

後付けで要件定義書を作成し、再度、要件定義との整合をとった。[SEC]

【発生問題】 要件定義遅延、品質不良

【リスク要因】

キーリスクは、⑫ドキュメント（要件定義書）の不備（技術リスク）、⑰あいまいなスコープおよび⑱あいまいな要件にある。

関連リスクは次の通り。

④ 組織能力不足（マネジメントリスク）
⑥ 要件定義能力（技術リスク）
⑦ リーダーのプロジェクトマネジメント能力（マネジメントリスク）

【リスク回避策】

この事例はあいまいな要件定義の典型的な例で、要件定義書そのものがなかったというものだ。これでまともなものが作れるわけもないだろう。

これまでに示された要件定義遅延問題に対しての対策を整理すると次のようになる。

◎要件定義遅延対策

□ ① 顧客との密なコミュニケーションを維持することで顧客を早い段階からパートナー化しておくこと。
□ ② 要件未凍結のリスク排除のタイムリミットをあらかじめ設定しておくこと。
□ ③ リスク排除のタイムリミットがきたら仕様提案、機能分割リリースなどの提案アクションを直ちにとること。

◎要件定義工程では適時のレビューが必須

□ ① 要件工程の1／3から1／2工程あたりで見極めレビューを必ず実施すること。
□ ② ダメだと判断したらすぐに対策アクションを実施すること。
□ ・ 要件定義メンバーを増強すること。
□ ・ 顧客と一緒に要件の骨子をレビューし顧客価値の順にプライオリティ付けを行い。優先度の低い要件はドロップするか二次開発項目とすること。
□ ・ 顧客側の要件有識者を増強すること。
□ ・ 開発内容を分割リリースにすること。
□ ・ リリース時期を延ばすこと。

◎要件定義工程での待ちの姿勢は最悪だ

□ ① 顧客の要件定義力が弱い場合、開発側からの積極的アプローチで要件の掘り起こしおよび提案を行うこと。
□ ② 顧客との密なコミュニケーションを維持することで顧客を早い段階からパートナー化しておくこと。
□ ③ 要件未凍結のリスク排除のタイムリミットをあらかじめ設定しておくこと。
□ ④ リスク排除のタイムリミットがきたら仕様提案、機能分割リリースなどの提案アクションを直ちにとること。

◎要件定義の優先順位の誤り

□ ① 要件定義は顧客価値の高い順に実行すること。
□ ② 開発の着手も、先に要件が決定した顧客価値の高い要件から実行すること。

【リスク回避のアクション・チェックリスト】

◎要件定義遅延対策のまとめ

□ ① 早い段階から顧客をパートナー化しておくこと。
□ ② 仕様凍結の事前タイムリミットを設定しておくこと。
□ ③ 事前タイムリミットがきたら速やかに対策アクションを実行すること。
□ ④ 顧客から要件が出されるのを待つ姿勢をやめ積極的な提案姿勢をとること。
□ ⑤ 要件には顧客価値の大きさに従って優先順位を付け、優先順位の高いものから仕様凍結を図り開発を実行すること。

第4章　事前準備・見積り工程／要件定義工程のリスク検出チェックリスト

すでに前章までにおいてそれぞれのリスクに対する対策アクションのチェックリストを記載してきたが、本章においてはこれらのリスクを事前に検出するためのチェックリストをまとめた形で記載する。

1）　事前準備・見積り工程のリスク検出チェックリスト

◎　他者依存的姿勢（マネジメントリスク）
- □　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

◎　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

◎　ユーザーの参加・協力度不足
- □　ユーザーの参加・協力度不足

◎　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）
- □　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・スパゲッティ化
- □　見積りプロセスの手続きルール違反

◎　見積り能力
- □　安易な現行機能の保証
- □　フィット＆ギャップ調査能力
- □　既存資産流用の可否判断
- □　顧客要件の精査

◎ 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

◎ メンバーの技術能力、ヒューマンエラー（うっかりミス）
- □ 開発言語能力不足
- □ 新技術知識不足
- □ 顧客業務知識不足

◎ 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

◎ 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不足

◎ ドキュメント（要件定義書・設計書・チェックリスト・手順書など）の不備
- □ 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）

◎ 開発費不足
- □ 赤字受注

◎ あいまいな開発範囲（スコープ）
- □ あいまいなスコープ

◎ あいまいな要件定義
- □ あいまいな要件

◎ 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）
- □ 未経験言語の技術情報不足
- □ 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

2） 要件定義工程のリスク検出チェックリスト

◎ 他者依存的姿勢
- □ 顧客の動き待ちの姿勢の結果による納期遅延

◎ 上位マネジメントの関与不足
- □ 顧客交渉戦略･能力不足（18、21、22、24，26）
- □ 大幅な仕様追加要求の丸呑み（18、19，20）
- □ 大幅な開発費削減要求の丸呑み（21）
- □ 大幅な納期短縮要求の丸呑み（22）

◎ ユーザーの参加･協力度不足
- □ ユーザーの参加･協力度不足

◎ 組織能力
- □ 組織能力不足（未熟な組織文化、戦略の欠如）

◎ 要件定義能力
- □ 要件定義能力不足

◎ 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）
- □ 適時の支援依頼発信能力不足
- □ 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
- □ 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）
- □ タイムリミット管理能力不足
- □ 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

◎ プロセス管理の有無
- □ 仕様凍結未確認のまま設計着手

◎ コミュニケーション能力（阻害･ギャップ）
- □ プロマネと上司間のコミュニケーション
- □ 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
- □ 営業と開発部門間のコミュニケーション能力

◎ ドキュメント（要件定義書･設計書･チェックリスト･手順書など）の不備
- □ 要件定義書なし

◎ あいまいな開発範囲（スコープ）
- □ あいまいなスコープ

◎ あいまいな要件定義
- □ あいまいな要件定義

3） 要件定義書のリスク検出チェックリスト

過不足のない要件定義書の条件として、IEEE830は下記の品質特性を定義しています。

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

また自然言語で記述される要求を仕様化する方法「ＵＳＤＭ（Universal Specification Describing Manner）」における具体的な記述方法をIEEE830と対比させたものが次に示した「要件定義書のチェックリスト」です。 要件定義書の作成者においてはこれらの要件定義書における基本的な品質要件を満足させる必要があり、一方要件定義書の受領者においては受け取った要件定義書の品質レベルを確認することに役立つでしょう。またこのチェックリストは要件定義書のみならず設計書を初めとしたあらゆる開発ドキュメントの品質レベルの指針としても活用できるでしょう。

【要件定義書のリスク検出チェックリスト】

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。
- ☐ 上位のシステム要件定義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

- ☐ 要件定義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。
- ☐ 要件定義書の“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。
- ☐ 要件定義書では、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。
- ☐ 要件定義書の構成や内容は、後工程の読者に分かるように書くこと。
- ☐ 「等」「etc」の文言は使用しないこと。

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件定義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、要件定義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。
- ☐ 要件定義書には、「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
- ☐ モレなく書くこと。
- ☐ 要件定義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、thenとelseの両方を明らかにすること。

4. 矛盾がないこと

- □ 要件定義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、要件定義書に記述された要求を満たしているかどうかを確認可能であること。
- □ 検査部門の人に、「検査可能」という側面から要件定義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

- □ 要件定義書に対する変更が、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、同じ要求が要件定義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。つまり、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

- □ 要件定義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、要件定義書に書き戻すこと。
- □ 「要求」と「理由」をセットで表現すること。
- □ 要件定義には固有の記番号を付けること。

（参考資料： IEEE830品質特性、USDM）

4） 要件定義書の記述内容

要件定義書に記述されるべき項目の一例をサンプルとして示します。

1 表紙・変更履歴・目次
2 目的
 2-1 システム化の理由
 2-2 システム化前後の運用
 2-3 システム化のメリット
3 システム概要
 3-1 システム構成図
 3-2 ハード構成
 3-3 ソフト構成
 3-4 システム運用一覧
4 運用
 4-1 運用スケジュール
 4-2 概要ビジネスフロー（運用フロー）
5 業務概要
 5-1 業務一覧
 5-2 ビジネスルール定義
6 機能
 6-1 詳細ビジネスフロー
 6-2 画面一覧
 6-3 画面設計
 6-4 画面項目説明
 6-5 画面遷移図
 6-6 帳票一覧
 6-7 帳票設計
 6-8 帳票項目説明
 6-9 要求機能定義
 6-10 要求データ一覧
 6-11 要求データ仕様定義
 6-12 オンライン仕様一覧
 6-13 要求性能
 6-14 その他

要件定義書の受領時に、これらの要件定義の記述の有無および記述内容によって要件定義書の内容レベルの評価を行う必要があります。

各項目内容に対する評価は、例えば「要件として十分：3、多少の不明点あり：2、不十分：1、記述なし：0」などの採点法もあります。

合格点未達のものは直ちに要求元に、その理由を添付・返却し修正を要求することが望まれます。これらの継続的な活動は要件定義書の確実なレベルアップに貢献するでしょう。

さいごに

多くの失敗事例を見て改めて感じることは、切羽詰まった挙句常識から程遠い選択をすることが失敗の真因だということがよく分かります。

冷静に状況を判断すれば、自分が選択しようとしている方針では失敗の確率が非常に高いことが分かるはずですが、当座の苦しい状況から一刻も早く脱出したいという焦りが判断を狂わせ、何らの合理的根拠もないまま、やれば何とかなるだろうと無理に自分を納得させた結果本当は選ぶべきではない選択肢を選んでいるように思えます。

「われわれ人間は重要な選択を迫られた場合、考えに考え抜いた挙句に最悪の選択をする動物である」と言うことわざが思い出されます。

またノーベル賞を受賞した行動経済学者のダニエル・カーネマンはプロスペクト理論において次のように語っています。

【リスク下における意思決定の分析】

「八方塞がりになった人々が絶望的な賭けに出て、大損を免れるいちるの望みと引き換えに、高い確率で事態を一層悪化させる選択肢を受け入れる。対処可能だった失敗を往々にして大惨事に変えるのは、この種のリスクテークだ。確実な大損を受け入れるのはあまりに苦痛が大きく、それを完全に避けられるかもしれないという望みはあまりに魅力的である。その結果、起死回生の一手を打つしかないという決断に立ち至る。たとえば、すぐれた技術の登場で次第に追いつめられた企業が一発逆転を狙って残った資産を無駄に注ぎ込む、といったケースはこれに当たる。戦争でもこうしたことがよく起きる。敵の勝利はもう時間の問題だという事態になってからも、負けている側は敗戦を受入れがたいため、絶望的な戦いを続けることになりやすい。」

最良のリスクヘッジ法は、自分やプロジェクトをそのような切羽詰まった状況に置かないようにすることです。ひっ迫した状況を回避する唯一の方法は事前準備と改善活動の継続的な実施にあると確信されます。

ピーター・ドラッカーの次の言葉はプロジェクトを成功に導くことの神髄を良く表しています。

【成果をあげる者は時間からスタートする】

「私の観察によれば、成果をあげる者は仕事からスタートしない。時間からスタートする。計画からもスタートしない。何に時間がとられているかを明らかにすることからスタートする。成果をあげる者は時間こそが、真に普遍的な制約条件であることを知っている。」

（ピーター・ドラッカー）

最初から全てのリスク要因に対して完璧なリスクの排除や回避はできません。最初は現在もっている組織能力やプロマネの能力に相応した成果しか出せないのかも知れません。しかしこれらのリスク回避・排除活動がもし以前はほとんど実行されてこなかったとしたならば、最初の成果は大きな成果がきっと出るに違いありません。また更に次から次へと繰り返されるプロジェクト活動に中で、これらのリスク回避および排除活動を繰り返していけばプロマネやチームの能力はその都度増強され、そのような人口が増えていけば遠からず組織全体の能力もレベルアップされていくことでしょう。

添付資料

CMMIにおける「要件管理」（レベル2）に関する規定の抜粋

下記は、CMMI レベル2の「要件管理」に関するガイドラインの抜粋です。ここに記されているゴールおよびプラクティスをチェックリストに見立てて自組織での実行の有無をチェックすることでレベル2程度の実力があるのか否かが評価できます。

『要件管理』（成熟度レベル 2 のプロジェクト管理のプロセス領域）

目的

『要件管理』（REQM）の目的は、プロジェクトの成果物の要件および成果物構成要素の要件を管理すること、およびこれらの要件と、プロジェクトの計画および作業成果物との間の整合性を確保することである。

導入説明

「要件管理」プロセスは、プロジェクトが受け取るかまたは生成するすべての要件を管理する。これらの要件には、技術面の要件および技術面以外の要件の両方が含まれ、さらに組織がプロジェクトに課す要件も含まれる。

特に、『要件開発』プロセス領域が実装されている場合、そのプロセスから成果物要件および成果物構成要素の要件が生成される。これらの要件は、「要件管理」プロセスにより管理される。

プロセス領域全般にわたって、『成果物』および『成果物構成要素』という用語を使用する場合、それらが意図する意味には、サービス、サービスシステム、およびそれらの構成要素も含まれる。

『要件管理』プロセス領域、『要件開発』プロセス領域、および『技術解』プロセス領域がすべて実装されている場合、関連するプロセスが密接に結びつけられ、同時並行的に実施される場合がある。

プロジェクトは、プロジェクトの計画策定および実行のニーズに応えるために、承認された一連の要件が管理されるようにするための適切なステップをとる。承認された要件提供者からプロジェクトが要件を受け取ると、これらの要件がプロジェクトの計画に取り入れられる前に課題を解決し誤解を予防するために、要件は要件提供者と共にレビューされる。要件提供者および要件受領者が合意に達すると、要件に対するコミットメントがプロジェクト参加者から獲得される。プロジェクトでは、要件の進化に伴って要件に対する変更を管理し、計画、作業成果物、および要件の間に発生する不整合を特定する。

要件を管理することの一部として、要件変更およびそれらの論理的根拠を文書化する。また、源となる要件、成果物と成果物構成要素のすべての要件、および他の指定された作業成果物との間で双方向の追跡可能性を維持する。（用語集にある『双方向の追跡可能性』の定義を参照のこと。）

すべてのプロジェクトには要件がある。保守活動の場合、変更は、既存の要件、設計、または実装に対する変更に基づく。また、製品の機能について増分毎に納入するプロジェクトでは、変更は、進化する顧客ニーズ、技術の成熟や陳腐化、および標準の進化に基づく場合がある。両方

の場合で、要件変更が、もしあれば、顧客または最終利用者からの変更要求として文書化されることもあるし、要件開発プロセスから受け取った新たな要件という形式をとることもある。その出所や形式にかかわらず、要件への変更により引き起こされる活動は、相応に管理される。

固有ゴール

SG1　要件を管理する

☐　要件が管理され、プロジェクト計画および作業成果物との不整合が特定されている。

プロジェクトでは、以下の事項を行うことによって、プロジェクトの全期間にわたり最新で承認された一連の要件を保守する:

☐　要件に対するすべての変更を管理すること
☐　要件、プロジェクトの計画、および作業成果物の間の関係を維持すること
☐　要件、プロジェクトの計画、および作業成果物の間の整合性を確保すること
☐　是正処置をとること

固有プラクティス

SP1．1　要件を理解する

☐　要件の意味に関して要件提供者と共に理解する。

プロジェクトが成熟し要件が導出されるにしたがって、すべての活動グループまたは専門分野グループが要件を受け取る。要件が徐々に増大することを避けるために、要件を受け取るための適切な経路または公式の出所を指定する基準を確立する。要件の意味に関して矛盾がない共有された理解に到達するようにするために、要件を受領する人員は、提供者と共に要件の分析を実施する。これらの分析および意見交換の結果が、承認された一連の要件である。

作業成果物の例

☐　1. 適切な要件提供者を識別するための基準の一覧
☐　2. 要件の評価および受け入れの基準
☐　3. 基準に照らした分析の結果
☐　4. 承認された一連の要件

サブプラクティス

☐　1. 適切な要件提供者を識別するための基準を確立する。
☐　2. 要件の評価および受け入れのための客観的な基準を確立する。

評価基準および受け入れ基準の欠如は、多くの場合、不十分な検証、高価な手戻り、または顧客による却下へと帰着する。

評価基準および受け入れ基準の例を以下に示す:

☐　明確かつ適切に述べられている
☐　完全である
☐　それぞれ互いに首尾一貫している
☐　一意に特定されている
☐　アーキテクチャ面の取り組み方法および品質属性の優先度と首尾一貫している
☐　適切に実装できる
☐　検証可能（テスト可能）である

☐ 追跡可能である
☐ 達成可能である
☐ 事業価値に結びついている
☐ 顧客にとっての優先事項であると特定されている

☐ 3. 確立された基準が満たされているようにするために要件を分析する。
☐ 4. プロジェクト参加者が要件にコミットメントを形成できるように、要件について、要件提供者と一つの理解に到達する。

SP1．2 要件に対するコミットメントを獲得する
☐ プロジェクト参加者から要件に対するコミットメントを獲得する。

前述の固有プラクティスは、要件提供者と一つの理解に到達することを扱った。この固有プラクティスは、要件を実装するのに必要な活動を実行する人々の間での合意およびコミットメントを扱う。要件はプロジェクト全般にわたって進化する。要件の進化に伴い、この固有プラクティスでは、プロジェクト参加者が、最新かつ承認された要件に対して、そして、その結果として生じる、プロジェクト計画、活動、および作業成果物での変更に対して、コミットするようにする。

作業成果物の例
☐ 1. 要件の影響評価
☐ 2. 要件および要件の変更に対する文書化されたコミットメント

サブプラクティス
☐ 1. 要件が既存のコミットメントに与える影響を評価する。
要件が変更されるとき、または新しい要件に着手するときには、プロジェクト参加者に対する影響を評価する。
☐ 2. コミットメントを協議し記録する。
既存のコミットメントに対する変更については、プロジェクト参加者が新しい要件または要件の変更に対するコミットメントを形成する前に、協議する。

SP1．3 要件変更を管理する
プロジェクト実行中の要件の進化に伴い、要件への変更を管理する。

要件はさまざまな理由で変更される。ニーズの変化および作業の進行に伴い、既存の要件に対して変更を行わなければならない場合がある。このような変更を、効率的かつ効果的に管理することが必須となる。変更の影響を効果的に分析するには、各要件の出所が把握されており、その変更の論理的根拠が文書化されている必要がある。プロジェクトは、変更制御に対する新しい取り組みまたは取り組みの改訂が必要かどうかを判断するために、要件変更率の適切な尺度を追跡したい場合がある。

作業成果物の例
☐ 1. 要件の変更要求
☐ 2. 要件変更の影響報告書
☐ 3. 要件の状況
☐ 4. 要件データベース

サブプラクティス

- ☐ 1. プロジェクトに与えられたか、またはプロジェクトによって生成されたすべての要件および要件変更を文書化する。
- ☐ 2. 変更に関する論理的根拠を含め、要件変更履歴を保守する。
  変更履歴の保守は、要件変更率の追跡に役立つ。
- ☐ 3. 直接の利害関係者の立場から要件変更の影響を評価する。
  成果物アーキテクチャに影響を与える要件変更は、多くの利害関係者に影響を与える場合がある。
- ☐ 4. 要件および変更データをプロジェクトで利用可能にする。

SP1．4 要件の双方向の追跡可能性を維持する

☐ 要件と作業成果物との間に双方向の追跡可能性を維持する。

この固有プラクティスの意図は、要件の双方向の追跡可能性を維持することである。(用語集にある『双方向の追跡可能性』の定義を参照のこと。) 要件がうまく管理されていれば、源となる要件から下位レベルの要件へ、そしてそれらの下位レベルの要件から原要件へ戻っての追跡可能性を確立できる。このような双方向の追跡可能性は、源となる要件のすべてが完全に取り上げられているかどうか、および下位レベルの要件のすべてを有効な出所に対して追跡できるかどうかを判断することに役立つ。

要件の追跡可能性は、中間作業成果物や最終作業成果物、設計文書の変更、およびテスト計画など、他の実体との関係も扱う。追跡可能性は、垂直方向の関係ばかりでなく、インタフェース横断的な関係など水平方向の関係を扱う場合もある。追跡可能性は、プロジェクトの活動および作業成果物への要件変更の影響を評価する場合に特に必要となる。

考慮する追跡可能性の側面の例を以下に示す:

- 追跡可能性の範囲:追跡可能性が必要な範囲を示す境界
- 追跡可能性の定義:論理的な関係を必要とする要素
- 追跡可能性の種類:垂直方向の追跡可能性や水平方向の追跡可能性が必要な場合

このような双方向の追跡可能性は、常に自動化されるわけではない。表計算プログラム、データベースおよび他のよくあるツールを用いて手作業で行われる場合もある。

作業成果物の例

- ☐ 1. 要件の追跡可能性マトリクス
- ☐ 2. 要件追跡システム

サブプラクティス

- ☐ 1. 下位レベル（導出された）要件の出所が文書化されているようにするために、要件の追跡可能性を維持する。
- ☐ 2. 要件から、その導出要件への追跡可能性を維持する。また、要件から作業成果物への割り当てに対する追跡可能性を維持する。
  追跡可能性を維持する対象となる場合がある作業成果物には以下が含まれる: アーキテクチャ、成果物構成要素、開発の反復（または増分)、機能、インタフェース、オブジェクト、人員、プロセス、および他の作業成果物。
- ☐ 3. 要件の追跡可能性マトリクスを生成する。

SP1．５ プロジェクト作業と要件の間の整合性を確保する

☐ **プロジェクトの計画および作業成果物が要件と整合している状態が保たれているようにする。**

この固有プラクティスでは、要件とプロジェクト計画および作業成果物との間の不整合を発見し、不整合を解決するための是正処置を開始する。

作業成果物の例

- ☐ 1. 出所および状況を含む、要件とプロジェクト計画および作業成果物との間の不整合について文書化したもの
- ☐ 2. 是正処置

サブプラクティス

- ☐ 1. 要件および要件に対する変更との首尾一貫性について、プロジェクトの計画、活動、および作業成果物をレビューする。
- ☐ 2.（もしあれば） 不整合の出所を特定する。
- ☐ 3. 要件ベースラインへの変更の結果生じる、計画、および作業成果物に対して行うあらゆる変更を特定する。
- ☐ 4. 必要なあらゆる是正処置を開始する。

＊『要件開発』（成熟度レベル 3 のエンジニアリングのプロセス領域）に関しては、下記の引用資料　p325以降を参照のこと。）

引用:『開発のための CMMI® 1.3 版 CMMI-DEV, V1.3 CMMI』成果物チーム
より良い製品とサービスを開発するためのプロセス改善 2010 年 11 月 技術報告書 CMU
http://www.sei.cmu.edu/library/assets/whitepapers/CMMI-DEV-V1.3-Japanese.pdf

参考文献・出典

[プロジェクト実態調査800社　2009. 2. 2]　p1
　プロジェクトの成功率　プロジェクト実態調査800社、日経　2009/02/02

[SEC]　p7～p42
　第2章および第3章における事例紹介の「事例タイトル」、「事象概要」および「判断の誤り」についてはIPA／SEC発行『ITプロジェクトの「見える化」上流工程編（2007 年 4 月）・中流工程編（2008 年 8 月）・下流工程編（2006 年 5 月）』Copyright (c) 2010 IPA を引用した。

CMMIにおける「要件管理」（レベル2）に関する規定の抜粋
　『開発のための CMMI® 1.3 版　CMMI-DEV, V1.3 CMMI』成果物チーム
より良い製品とサービスを開発するためのプロセス改善　2010 年 11 月　技術報告書　CMU
http://www.sei.cmu.edu/library/assets/whitepapers/CMMI-DEV-V1.3-Japanese.pdf)