はじめに

プロジェクト見える化部会は、2005年の発足から3年が過ぎた。これまでに、ソフトウェア開発のライフサイクルを上流工程、中流工程および下流工程と分けて捉え、各工程の特質に根ざしたITプロジェクトの「見える化」手法やツールを開発してきた。これらの活動を通じ、一貫して現場での課題認識や実践経験に基づくプロジェクト見える化を提言してきた。

ここでいう上流工程、中流工程、下流工程とは、次のようなものである。

①上流工程：経営者も含めた要件定義とシステム設計を行うグラウンド・デザインの工程
②中流工程：ソフトウェア設計→プログラミング→単体テストという技術者による開発工程
③下流工程：プログラムの結合・総合テストを通じて、システムの構築・稼働に至る最終工程

プロジェクトは上流、中流、下流の各工程で、置かれている状況が異なるため、それぞれの特徴に応じた見える化を検討し、これまで上流工程編、下流工程編の2編を発行した。

これに対して、「見えないプロジェクトが見えるようになった」「実践的な事例からプロジェクトの実態を説明できるようになり問題解決の糸口を見いだせるようになった」「中流工程編も欲しい」との、プロジェクト・マネージャを中心として、広くソフトウェア業界の多くの方々から手応えのある反響や、力強いご支持をいただいた。

本書では中流工程におけるプロジェクトの見える化に関する手法について述べる。見える化のために中流工程を、「上流工程のアウトプットである要件とプロジェクト計画を受け、要件と整合したシステムを実装する工程」と捉えた。これはすなわち、

①上流工程では情報が不足しているため、どうしても「見えなかった」問題
②中流工程で新たに発生する大幅なプロジェクト計画変更などの問題
③下流工程で「見えた」時には手が付けられず、もはや手遅れとなってしまう問題

といった問題に対して、下流工程に持ち越さないための「品質作り込みの砦」として、中流工程を捉えることである。

そして、中流工程が「砦」としての役割を果たすために、問題を速やかに見える化し、問題の是正結果も見える化するための手法として、中流工程に必要となる俯瞰図、チェックシート、失敗プロジェクト事例集、定量的見える化ツール、および統合的見える化ツールを提示し、「要

求の実装検証」レベルの粒度・詳細度で具体化した。また「砦」を守るための、中流工程におけるプロジェクト・マネジメント論についても総括する。

本書をプロジェクト・マネージャやプロジェクト・リーダーなど直接のプロジェクト関係者に加えて、PMOなどプロジェクト管理部門の方々にも、プロジェクト見える化手法の全体を理解するのに活用していただきたい。プロジェクトのシステム開発現場で、見えにくいプロジェクトを見える化し、プロジェクトのトラブルを防止する一助になることを期待している。

2008年8月

プロジェクト見える化部会
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

ITプロジェクトの「見える化」 中流工程編

はじめに …… 1

第1章 中流工程の見える化の目標 …… 6
1.1 ITプロジェクトの実情 …… 6
1.2 見える化の目標 …… 7
1.2.1 システム開発工程 …… 7
1.2.2 見える化の目標 …… 8
1.2.3 中流工程を航海に例えて理解する …… 9

第2章 中流工程における見える化の全体像 …… 12
2.1 3つのアプローチ（定性的、定量的、統合的） …… 12
2.2 定性的な見える化アプローチ …… 14
2.3 定量的な見える化アプローチ …… 14
2.4 統合的アプローチ …… 14

第3章 定性的見える化アプローチ …… 16
3.1 概要 …… 16
3.2 俯瞰図の意義 …… 16
3.2.1 俯瞰図とは …… 16
3.2.2 ドミナント・アイテムと俯瞰図作成のねらい …… 17
3.3 俯瞰図を使った見える化 …… 17
3.3.1 ステークホルダー俯瞰図 …… 19
3.3.2 プロジェクト推進体制俯瞰図 …… 20
3.3.3 周辺システム構成俯瞰図 …… 20
3.3.4 システム構成俯瞰図 …… 24
3.3.5 スケジュール俯瞰図 …… 24
3.3.6 要員遷移俯瞰図 …… 24
3.3.7 役割分担表 …… 26
3.3.8 プログラム関連図－作業関連図 …… 27
3.4 チェックシートを使った見える化 …… 28
3.4.1 チェックシートを用いた見える化の流れ …… 30
3.4.2 自己評価シート …… 30
3.4.3 自己評価の実施方法 …… 31
3.4.4 ヒアリングシート …… 34
3.4.5 ヒアリングの実施方法 …… 34
3.4.6 チェックシートの改良 …… 39
3.5 中流工程事例集 …… 41
3.5.1 事例集の公開 …… 41
3.5.2 問題の発生傾向と対策 …… 42
3.5.3 要求仕様変更 …… 46
3.5.4 中流工程におけるSI技術の問題 …… 47

第４章　定量的見える化アプローチ……48
4.1 概要…… 48
4.2 測定項目リスト…… 48
4.3 導出尺度の見方と分かることの例…… 49
4.3.1 タイム（時間）に関する例…… 49
4.3.2 品質に関する例…… 52
4.3.3 人的資源に関する例…… 53

第５章　統合的見える化アプローチ……54
5.1 概要…… 54
5.2 中流工程の統合的見える化アプローチ…… 54
5.3 中流工程における統合的見える化アプローチの観点…… 56
5.3.1 中流工程の序盤（ソフトウェア設計）…… 56
5.3.2 中流工程の中盤（プログラミング）…… 57
5.3.3 中流工程の終盤（ソフトウェア・テスト）…… 58
5.4 実装検証分類表の使い方…… 59
5.4.1 定性的情報の裏付けを取るための定量的情報を探る…… 59
5.4.2 定量的情報から問題の根幹を特定する…… 61
5.4.3 想定される事例から監視すべき定性的、定量的情報を知る…… 62
5.5 実装検証分類表の改良…… 64

第６章　中流工程の見える化によるプロジェクト・マネジメント……66
6.1 中流工程におけるプロジェクト・マネジメントの特徴…… 66
6.1.1 上流工程での完成度…… 66
6.1.2 発注者と受注者の意図の違い…… 67
6.1.3 マネジメント手法が異なる…… 67
6.1.4 文化・言語の違いからくるあいまい要件…… 68
6.2 中流工程における見える化の全体像とプロジェクト・マネジメント…… 68

おわりに…… 76

付　録
1. 自己評価シート…… 78
2. ヒアリングシート…… 90
3. プロジェクトにおける問題事象と対策の事例集……110
4. 中流工程分析ツール…… 136
5. 測定分析データ一覧表…… 142

参考文献…… 166

第1章

中流工程の見える化の目標

1.1 IT プロジェクトの実情

最近のプロジェクトのトラブル原因を調査すると、上流工程での要件定義があいまいなまま、中流工程でのプログラム(*)への変換作業が行われていることが多い。

これは、上流工程の作業の遅れを引きずり、十分に訓練された要員が整わない状態で中流工程のプロジェクトが進行してしまうからである。また、プロジェクト管理の方針を決定しない状態で、大量の要員を抱えながら何をすべきかをはっきりさせないまま開発をしているからである。

プロジェクト・マネージャは、定例の進捗会議で委託先から根拠も不明な「予定通り」という報告を受け、それを集約して顧客へ報告することがプロジェクト管理と思っている例も見受けられる。

中流工程では、専門家による分業体制に移行し、個人への作業依存が前後の工程に比べて高くなる特徴がある。

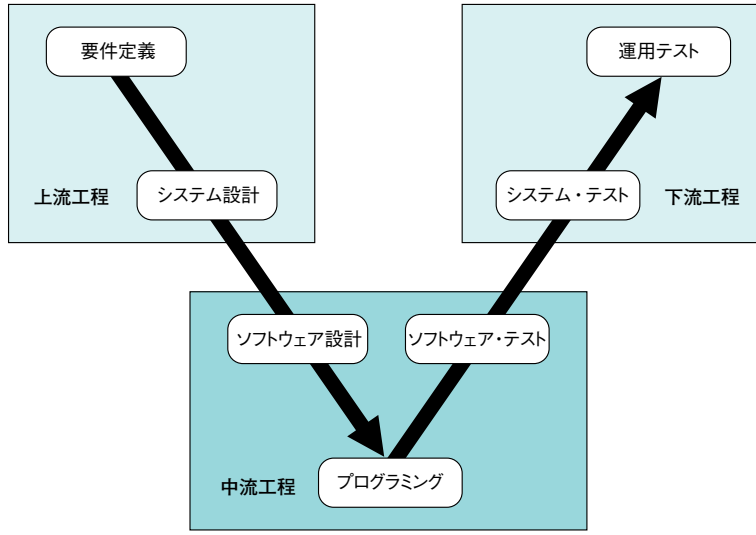
具体的には、詳細設計では、ソフトウェア・コンポーネントを、コーディングの単位となるソフトウェア・ユニットのレベルへと分解し詳細化して、全ソフトウェア要件をソフトウェア・ユニットに割り振る。その後、1つまたは複数のソフトウェア・ユニット(プログラム)を各人員に割り当て、並行作業を行う。

中流工程では、作業者数さえ増やせば並行作業の多重度も高められるので、複数の会社への委託を含め作業者数も多くなる。しかも個人作業に依存する割合が高いので、目標品質などの発注条件および詳細な機能仕様を明確にしないと、品質のばらつきも大きくなる。従って、品質はばらつくというリスクを前提で管理を行わなければ、不良の作り込み防止はできない。

また、作業場所の分散、多層構造の組織になり、プロジェクト管理で重要な「円滑なコミュニケーション」が図り難くなる。

(*)本書では、共通フレーム2007における「分割してコンパイル可能なコードのひとまとまり」としての「ソフトウェアユニット」と同様な意味で「プログラム」と呼んでいる。

図表1-1●ITプロジェクト中流工程の位置



中流工程は、**図表1-1**に示す「システム開発V字プロセス」の折り返し地点で、実世界の仕組みをコンピュータの仕組みに翻訳する工程である。図の下側ほど個人作業の様相が濃くなり、上側ほど共同作業的になる。中流工程の特徴は、作業効率が作業者の能力で数倍の差が生じることである。利用するソフトウェア製品の特性の理解によっても、品質に大きな差が出る。

このように、中流工程では、実質的な品質を決定してしまう重要な工程であると認識すべきであるが、多くの場合、油断し下流工程で大混乱を引き起こすケースが多いのである。

1.2 見える化の目標

1.2.1 システム開発工程

本書で対象とする見える化は、**図表1-2**の通りで、ソフトウェア設計、プログラミング、ソフトウェア・テストを中流工程とした。

本書の中流工程は、「ソフトウェア設計」「プログラミング」「ソフトウェア・テスト」に分けているが、SLCP(*)のプロセス／アクティビティに対応付けると、「ソフトウェア方式設計」「ソフトウェア詳細設計」「ソフトウェアコード作成およびテスト」「ソフトウ

(*)SLCP：Software Life Cycle Processes　国際規格ISO/IEC12207(JIS X0160)

ェア結合」「ソフトウェア適格性確認テスト」になる。

「ソフトウェア設計」は、SLCPの「ソフトウェア方式設計」に、「プログラミング」は「ソフトウェア詳細設計」と「ソフトウェアコード作成」に、「ソフトウェア・テスト」は「ソフトウェアコード作成およびテスト」「ソフトウェア結合」「ソフトウェア適格性確認テスト」に対応付けられる。

1.2.2 見える化の目標

中流工程では、顧客が要求する要件を、漏れなくコードに置き換えていく工程である。ソフトウェア開発の有識者により、インプットとなるソフトウェア要件のレビューを十分に行い、明解になっている機能要件と非機能要件(*)に対してあいまいさを極力少なくすることである。

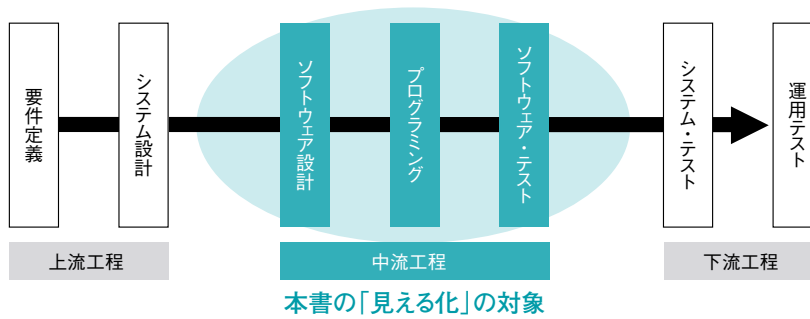
次に個人差によるばらつきを減らすために、ソフトウェア要件をコンポーネントからユニットへ詳細化する作業プロセスにおいて、作業の標準化、手順のマニュアル化を行い、作業要員へのトレーニングを行う。

中流工程の責任者は、個々に生じるプロジェクトの進捗、ソフトウェア品質のばらつきを是正し、プロジェクトの関係者に実情を見える化することが求められる。

具体的には、①作業品質の見える化(ばらつき、ルールの順守)、②非機能要件の見える化、③仕様変更要望への対応の見える化、などである。

開発環境から管理データをできる限り自動的に収集できる機能を組み込んだ管理環境を整備することで、作業場所が複数個所に分散しても、現状の状況をタイムリーに定量的な情報として

図表1-2●本書の見える化の対象



(*)非機能要件とは、利用者の要求を満足するためにソフトウェアが実現しなければならない要件の一種で、利用者の業務および手順を表す機能要件以外の要件をまとめていう。品質要件、技術要件、運用・操作要件、移行要件、付帯作業などを指す。

収集して分析し活用できる。

1.2.3 中流工程を航海に例えて理解する

中流工程を「上流工程のアウトプットである“要件”と“プロジェクト計画”を受けて、下流工程の要件に合ったシステムであるかどうかの検証へとつなぐ工程」と考えると、中流工程は以下に示すように、船が荷積みし、出航した状況に例えると分かりやすい(**図表1-3**)。

船は出航すると(システム開発に着手すると)、荷主(ユーザー)の責任より、船会社(ベンダー)の責任が大きくなる。当然、荷主(顧客)は船会社(ベンダー)を使って、相手に荷物を届ける(システム化により恩恵を受ける部門へのサービス提供する)本来の責任を持っている。

船は航海の途中(プロジェクト期間中)、レーダー、海図など様々な装置、情報を使って(ツール、チェックリスト、EPMなどの測定ツール)、現在の状況を把握し(見える化)、針路を確定(プロジェクト計画)する。そして、無事目的港に着岸し、荷物を相手に引き渡さなければならない(サービスイン)。

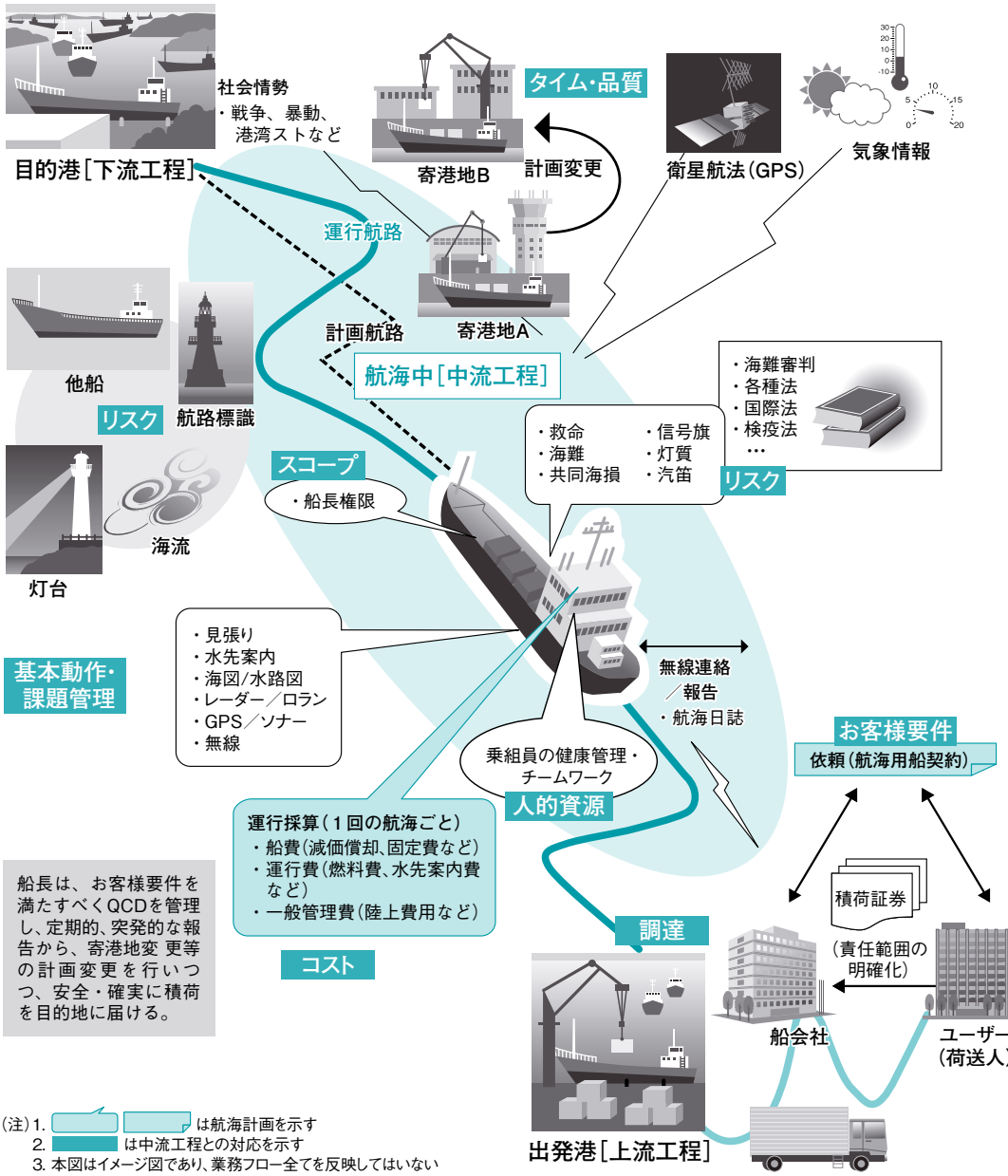
さらに、プロジェクト管理の視点で見ると、船を出航させる前に(上流工程で)、荷主(顧客)と船会社(ベンダー)間で「何を」「どれだけ」「いつまで」「いくらで」「どのようにして」などを明確(契約)にしなければならない。その際、運ぶものは、人、動物、原油、ガス、鉱材、危険物、食料品、冷凍の必要性などが明確(要件定義)になっていなければならない。船会社(ベンダー)は、それ(要件定義)に基づいて、船(プロジェクト体制の確立)を手配する。

航海の途中、自然災害など不測の事態が発生(問題、課題)した場合、船長(プロジェクト・マネージャ)は一刻も早く船会社(上司)、荷主(ユーザー)に報告し、対策を講じなければならない(エスカレーション)。

本来、ITプロジェクトもこのようなものであるはずだが、往々にして荷主(顧客)と船会社(ベンダー)間で「何を」「どれだけ」「いつまで」「どのようにして」運ぶかが、あいまいなまま出航(開発着手)するようなことが多い。これは、船が出航(開発着手)した後に、再び船を戻したり、別の船で荷物を運び、航海の途中(プロジェクトの途中)で荷物を積み替えたりするようなものである。

そして、ようやく荷物が積み込まれた(要件が確定した)としても、予定日までに到着できない(納期が間に

図表1-3●航海における船長の役割とプロジェクト・マネジメント



（注）1. ▭ ▭ は航海計画を示す
2. ▭ は中流工程との対応を示す
3. 本図はイメージ図であり、業務フロー全てを反映してはいない

合わない）と急に船の数を増やしたり（要員増強）、困難と思われる程のスピードアップ（残業、休日出勤など）を図ったりすることは、到底、船の世界ではありえないことだ。

本来の姿に戻すためには、出航前の荷積み（中流工程着手）の際、荷主（ユーザー）と約束した荷物（要件）が正しいか検証（要件機能検証、確認）する。検証結果「良」と判断した場合、荷積の作業手順を明確にし、クレーンなどの手配と荷積を実行する（ソフトウェア設計）。さらに、着岸後の荷揚げを見据えて（結合テスト、総合テスト）確認項目や手順などを明確にしておく（結合テストと総合テストのシナリオ作成）。同時に、出航に備えて、船員の健康状態、設備、機器、燃料の確認、外部との連絡などを実施する（プロジェクト計画、各種ルール）。準備が完了したら、出航（プログラミング、単体テスト）し、安全確実に運行しながら（プロジェクト・マネジメント）目的港（下流工程）に向かう――という細かい業務の積み重ねが必要になるのだ。

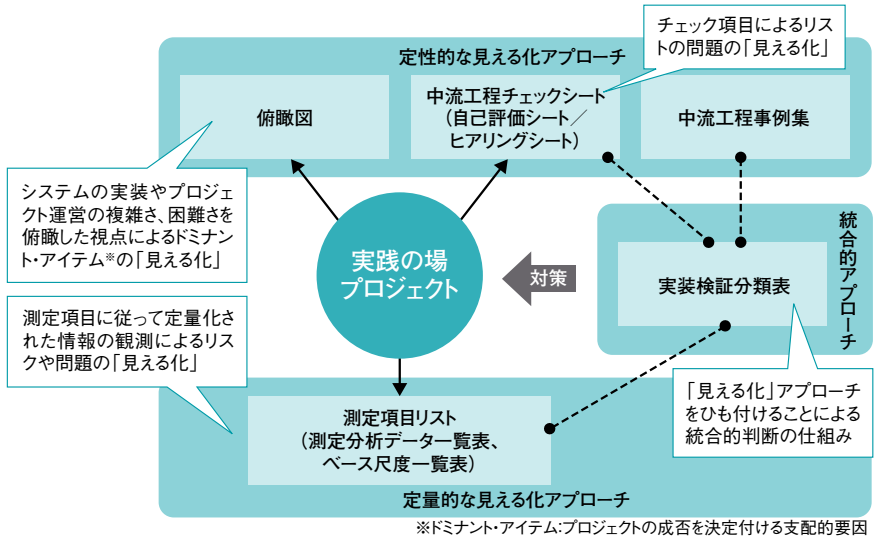
第2章

中流工程における見える化の全体像

2.1 3つのアプローチ(定性的、定量的、統合的)

前述したように、ITプロジェクトの中流工程においては、上流工程のアウトプットである要件の完成度を確認し、要件からプログラムに翻訳するまでの作業プロセスを決め、分業で並行して作業が進められるように段取り、各作業の成果物の品質、進捗、作業量などの計画と実績の差異を把握できるようにするため、①定性的な見える化アプローチ、②定量的な見える化アプローチ、③両者を統合的に見る統合的アプローチ、という3つのアプローチを適用する。

図表2-1●見える化の3つのアプローチ



※ドミナント・アイテム:プロジェクトの成否を決定付ける支配的要因

見える化の3つのアプローチを**図表　2-2**に、それぞれ示す。
2-1に、見える化ツールの一覧を**図表**

図表2-2●見える化ツールの一覧

アプローチ	ツール・資料		説明
定性的な 見える化 アプローチ	俯瞰図		ITプロジェクトの持つ本質的な問題（変化／変革による潜在的問題の多さ、ステークホルダーの多さ、加速度的な進化・変容を続けるIT技術・利用技術などの影響）の下で、ITプロジェクト・マネジメントにかかわる諸要素を俯瞰するための図。 これを用いることにより、プロジェクト・マネジメント・プロセスで起こりうる問題を予測し、「リスクの見える化」を図り、リスクを加味したマネジメントを展開できる。
	チェックシート		プロジェクトのリスクや問題の洗い出しに用いる。自己評価シートとヒアリングシートの2種類があり、プロジェクト・マネージャ自身の意識と、プロジェクト・マネージャを第三者がヒアリングした結果との差異に基づいて、プロジェクト・マネージャにリスクや問題への気付きを与える。
		自己評価 シート	プロジェクトの状況について、プロジェクト・マネージャが自らチェックするためのチェックシート。 各項目に3段階で評価を記入すると、レーダーチャートで結果を表示し、特に注意するべき項目と結果を対策案とともに表示する。
		ヒアリング シート	プロジェクトの状況について、外部の専門家がプロジェクト・マネージャにヒアリングをするときに活用するチェックシート。各項目に5段階で評価を記入すると、レーダーチャートで結果を表示し、特に注意するべき項目と対策案を表示する。自己評価シートの結果と合わせることで、プロジェクト・マネージャの自己評価との乖離などをレーダーチャートなどで見ることができる。
	中流工程事例集		問題が起因する工程に分け、プロジェクト失敗事例をまとめたもの。 この事例を参考にすることによって、同じミスを繰り返さないようにでき、同様なリスクや問題への対応策を見つけるのに役立つ。
定量的な 見える化 アプローチ	測定項目リスト		プロジェクトの状況を定量的に測定するための測定項目とその測定方法、分析方法をまとめた。
		測定分析 データー覧表	
統合的 アプローチ	実装検証分類表		実装検証分類表は、定性的、定量的な情報と、事例を組み合わせて分析する事により、統合的な見地で判断を下せるようになっている。 実装検証分類表では、各工程間の関連を縦軸に、定量的・定性的情報と事例を横軸に構成し、この分類表を用いる事により、各情報を関連付けた統合的な分析が可能となる。

2.2 定性的な見える化アプローチ

プロジェクトを全体的に捉えるための俯瞰図、問題が潜んでいる可能性のある個所を特定するためのチェックシート、過去のITプロジェクトで発生した中流工程の問題に関する事例集を用意した。

中流工程の俯瞰図では、プロジェクトに従事する人数も関係する会社も上流工程に比べて増えるため、ステークホルダー俯瞰図およびプロジェクト推進体制俯瞰図に加え、役割分担を図にした「役割分担図」、作成するプログラムの複雑な構成を俯瞰する「プログラム関連図」を準備するとよい。

詳しくは、第3章において解説する。

2.3 定量的な見える化アプローチ

中流工程におけるプロジェクトの状態を数値的に見える化するための項目をまとめた測定分析データ一覧表を用意した。

それらの数値を見てどのような状況として判断すべきなのかが理解できる。プロジェクト状況を定量的に測定する際に利用でき、計画と実績との差、時系列での変異、指標値との違いなどを明確に捉えることができ、実装時の誤りを簡易に捉えられる。

詳しくは、第4章において解説する。

2.4 統合的アプローチ

プロジェクトの経験豊かな有識者がどのような視点でプロジェクトの問題点を洗い出すかを分類した項目に従って、おのおののツール（チェックシート、測定分析データ一覧表、中流工程事例集）をひも付けた「実装検証分類表」を用意した。

各ツールを統合的に利用することで、中流工程での問題の早期発見、下流工程でのリスクの的確な洗い出しが行えるようになっている。

中流工程では、複数社の多数の要員が分散した場所で並行して作業を行うため、各種ツールを活用し、効率的、即時的にプロジェクトをマネジメントすることが求められ、統合的アプローチを活用することで、効果的な対策を講じられる。

詳しくは、第5章で解説する。

第3章

定性的見える化アプローチ

3.1 概要

中流工程では、成果物（プログラムなど）の内部に、不良（品質の悪さ）が作り込まれてしまうことが往々にしてある。

原因には、①上流工程では顕在化しなかった問題（要件定義、基本設計の甘さ）、②中流工程で発生した問題（プログラム仕様書の記述レベルがあいまいなのにプログラムを作成してしまうなど）、の2種類がある。

中流工程でこれらの問題を放置した状態でプロジェクトを進めると、下流工程で、急に問題が顕在化することになる。

品質が悪いまま下流工程に移行した場合は、品質改善のための作業手戻りが多く発生し、進捗が遅延し、結果として納期が守れないといった問題が発生しやすくなる。上流工程で顕在化しなかった問題と中流工程で生じた問題に対する見える化を図り、品質不良を未然に防ぐことが重要である。

本書では見える化の具体的なツールとして、俯瞰図、チェックシート（自己評価シートとヒアリングシート）、事例集を用いた定性的見える化アプローチについて説明する。

3.2 俯瞰図の意義

3.2.1 俯瞰図とは

俯瞰とは、地面にいては把握し難い事柄を、高い所から見下ろすことによって、地上をより的確に把握することである。すなわち、システム開発プロジェクトにおける俯瞰とは、それぞれの個別事情に埋没して見えにくい全体的な開発状況を把握するという意味で使っている。

プロジェクト開発の現場にいると「木を見て、森を見ず」の例えのごとく、目先の状況やマイナーな問題に目を奪われ、プロジェクト全体に多大な影響を及ぼす本質的な問題を見失って

しまうことが多い。この点を解消するためのツールが「俯瞰図」である。

ただし、同じものを見ていても、見る人の関心がどこにあるかで、見えるものが異なり、「見方」が変わるということに注意が必要である。

3.2.2 ドミナント・アイテムと俯瞰図作成のねらい

システム開発プロジェクトでは、必ずプロジェクトの成功、失敗を決めるような支配的な要因が存在する。これを「ドミナント・アイテム」と呼ぶ。

しかし、システム開発プロジェクトでは、プロジェクトごとに特色があり、何が要因となって、結果的に何が起こるのか、予測がつき難い。

このため俯瞰図の作成では、「何がドミナント・アイテムになりそうか」「リスクを事前に阻止するためには、何がどこまで見えていなければならないか」に関心を絞って考察する必要がある。それを繰り返し、自分で納得できるまで俯瞰図を練り上げていく。

俯瞰図を練り上げていくと、俯瞰図を作ろうとしなければ闇の中に埋没したままとなっていたであろうプロジェクトの重要な要因が浮き彫りになり、プロジェクトを成功に導くマネジメントを行うため、どこに注力すべきかの「見方」が定まってくる。

自分が納得し、他人も納得させられる俯瞰図ができた段階で、プロジェクト・マネージャは、「ドミナント・アイテム」を掌握でき、より確実にプロジェクトを成功へと導けるようになるはずだ。

3.3 俯瞰図を使った見える化

中流工程で、利用する俯瞰図には8種類ある（**図表3-1**）。

中流工程の特徴として、上流工程および下流工程に比べて関係者の数も多く、かつ、関与する関連会社の数も多くなり、各組織の役割が明確になっていないことによる問題が発生しやすいことが挙げられる。そのため、ステークホルダー俯瞰図およびプロジェクト推進体制俯瞰図を用いてプロジェクトの全体像を把握するだけでなく、各組織の役割分担表を俯瞰図として取り入れる必要がある。

また、昨今のシステム開発は、プログラム類が多大な数になるに加え、複雑な構成になっている場合が多く、システム構成図に加えて「プログラム関連図」を俯瞰図として取り入れることが望ましい。

図表3-1●俯瞰図の種類

種類	使用方法
ステークホルダー俯瞰図	多くのステークホルダーが関係し複雑に利害関係が絡むプロジェクトの全体像を把握する。各組織でこの人を抑えておけばよいというキーパーソンを探し出し、俯瞰図にキーパーソンとなる人の名前を記入する。 中流工程では、顧客要件を実装するため、実現可能な詳細仕様の作成とシステムへの変換を専門性を持った複数の協力会社に委託し、並行して開発作業を進めるため、顧客要件の有識者、システム化技術の有識者などもキーパーソンに加え、協力体制と情報共有の仕組みを築き、プロジェクトを推進する。
プロジェクト推進体制俯瞰図	プロジェクトは、複数の組織および企業が集まって構成されている。各組織の果たすべきミッションを記入したプロジェクト推進体制の全体像を把握する。プロジェクトの発足時のキックオフミーティングで各組織のキーパーソンにミッションを認識し、合意してもらうことが原則である。 しかし、この合意プロセスが不完全のまま中流工程を始めてしまうケースや人事異動でステークホルダーが変わるケースもあり、中流工程で改めて俯瞰図を作成し直し、中流工程におけるドミナント・アイテムを浮き彫りにし、問題の早期是正を図る。
周辺システム構成俯瞰図	開発システムと連動する他システムとの関連を把握する。連動するシステム同士の関係、位置、インターフェース、システム性能がボトルネックにならないようにするため、総合テストおよび移行計画を策定するために俯瞰図を作成する。
システム構成俯瞰図	上流工程では方式検討のため、システム全体を把握可能なシステム構成俯瞰図を作成したことに対し、中流工程では、データベース・サーバーへの接続やアプリケーション・ソフト内のファイル構成などを検討するため、開発対象のアプリケーション・ソフトをインストールする機器を中心としたハードウェア間の結合を表現する俯瞰図を作成する。
スケジュール俯瞰図	多くの詳細スケジュールがあるプロジェクトでは、プロジェクト・マネージャが把握できるレベルのスケジュールが必要となる。このスケジュールは、クリティカル・パス関連に絞ったものとする。 結合テスト、総合テストがクリティカル・パスである場合は、この工程をさらに詳細に落とし込んだスケジュールを作成し、問題になりそうな部分をクローズアップさせる。
要員遷移俯瞰図	各工程において、作業要員の確保状況が一目で分かり、作業要員の中にキーパーソンがアサインされているかについても記入したものを作成する。 システム設計を行ったキーパーソンが中流工程での統合テストを準備(計画作成)し、下流工程で結合テストの実施、管理を行うことで品質向上、生産性向上が狙える。また、詳細設計やコーディングのキーパーソンが投入された作業成果物の品質と生産性は高い。そのためキーパーソンはどのプロジェクトでも引っ張りだこになってしまう。 そこで、この俯瞰図を活用して、キーパーソンを重要工程に配置できるような、スケジューリングを行う必要がある。
役割分担表	各組織の作業項目(役割および責任の範囲)を明確にしたものであり、特に組織間に跨った作業は作業分担があいまいになりがちである。そこで、役割分担表を作成し、各組織の役割を明確にする必要がある(中流工程では組織数が多くなるため、不明確な作業が多くなる)。
プログラム関連図	バッチ処理を例にあげると、1つの処理は複数のジョブやプログラムから成り立っており、複雑な構成をとっている。 ジョブ間、プログラム間の関係を関連図として作成し、プログラム作成およびテストなどのスケジュール作成の指針として活用する。

3.3.1　ステークホルダー俯瞰図

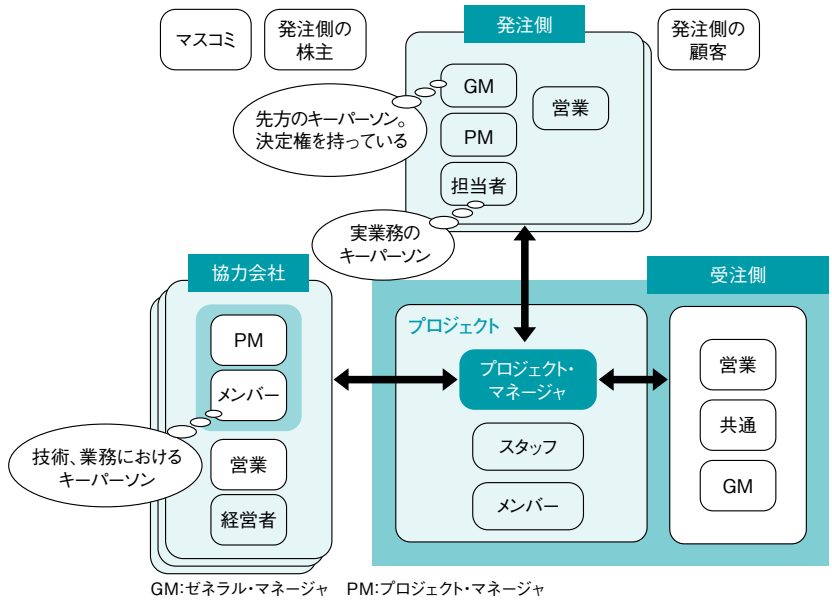
近年、マルチベンダーでのプロジェクト体制を取るシステム開発が増えている。こうした場合には、ステークホルダーの数が多いだけでなく互いに複雑に絡み合った構成になっている。

ステークホルダー俯瞰図とは、このように絡み合った状態を図にすることで、発注側、受注側およびその協力会社というステークホルダーの役割を分かりやすくしたものである（**図表3-2**）。プロジェクトを成功に導くためにも、俯瞰図によりキーパーソンを探し出し、キーパーソンとの人間関係を構築していくことが望ましい。

中流工程のステークホルダー俯瞰図は、発注側の関係者、受注側のプロジェクト・メンバー、さらには協力会社も加えて、上流工程から増えており、情報共用やスピーディな意思決定、タイムリーな対応が行えるように拡張し改訂する必要がある。

中流工程の未完作業や不具合が下流工程に入り込まないように、キーパーソンとの連携を効率よく図る仕組み

図表3-2●ステークホルダー俯瞰図の例



を、このステークホルダー俯瞰図を利用して構築するとよい。

通常、発注側、受注側、協力会社間での情報交換は契約などで取り決めた責任者を介して行うことになるが、各種情報がステークホルダーにも伝わるようにし、伝わったことが確認できることも、要員が最も多くなる中流工程では必要になる。

3.3.2 プロジェクト推進体制俯瞰図

一般的なプロジェクトは複数の組織および企業が集まって構成されているため、各組織の果たすべきミッションや組織間のインターフェースや役割分担があるべきである。これが不適切または不明確であると、失敗プロジェクトの原因となることが多い。このため上流工程において、プロジェクト推進体制の全体像を誰にでも分かるような俯瞰図（プロジェクト推進体制俯瞰図）で明らかにした上で、キックオフ・ミーティングでプロジェクト・オーナーや各組織のキーパーソンなどにミッションや組織間連携の関係を認識し、合意形成することが重要となる。

しかし、この合意形成プロセスが不完全のまま中流工程に突入してしまうケース、合意内容が形だけで実際には機能していないケース、人事異動によりステークホルダーが途中で変わるケースもよくみられる。

中流工程において、改めて俯瞰図を作成し直し、中流工程におけるドミナント・アイテムを浮き彫りにした上で、問題があれば早期の是正措置を行うことが、プロジェクトをより確実に成功へと導くことになる。

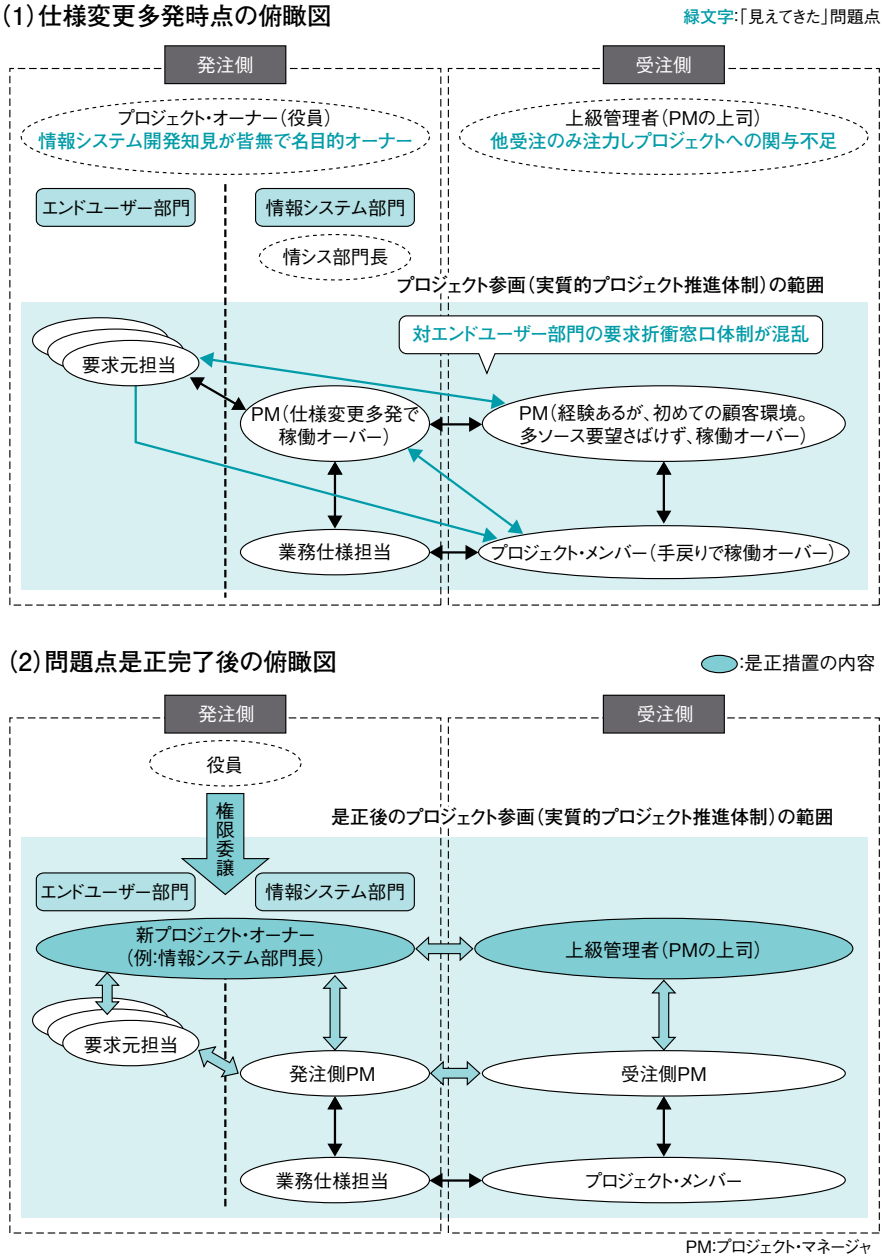
図表3-3は中流工程におけるプロジェクト推進体制俯瞰図の例である。ここでは、中流工程における問題の抽出から問題是正までの流れが分かるように、「仕様変更が多発した場合」と「問題点を是正した場合」という2つに分けて示す。

仕様変更が多発する原因は、実質的プロジェクト・オーナーの不在やエンドユーザー部門からの要求仕様の変更など発注側体制に問題があることもあれば、プロジェクト・マネージャの上司によるプロジェクトへの関与が不足するといった受注側体制に問題があることもあるなど、いろいろなケースが考えられる。

3.3.3 周辺システム構成俯瞰図

システムを開発する上で、周辺システムとのインターフェースの正当性、正確性は、プロジェクトの完成度に大きな影響があるため、周辺システムとの関連を明確にすることは、極めて重要である。

図表3-3●中流工程におけるプロジェクト推進体制俯瞰図の例



通常、プロジェクトは、単独のシステムで実現することは稀であり、大半は複数のシステム（サブシステム）から構成される。そして、各サブシステム間は、直接、間接を問わず、何らかの形で関連性がある。複数のサブシステムで構成される場合は、単一システムで実現される場合に比べ、プロジェクト・マネジメントが格段に難しくなる。関連性のある項目とは、例えば次のようなものがある。

・各サブシステムの機能配置の考え方、確定方法
・開発プログラム規模への影響考慮
・システム性能への影響考慮
・障害対策、障害時運用への影響考慮
・システム、業務運用などシステム全般的視点での検討、調整、合意
・移行の考え方、手順、検証の仕方
・保守、拡張性に対する考え方、指針
・テスト項目の設定、調整、手順、検証、再テスト時の計画
・報告、問題発生時の提起の方法、調整体制、方法

関連性の結果によっては、技術面への影響のみならず、プロジェクト管理の次のようなことに影響を与える。

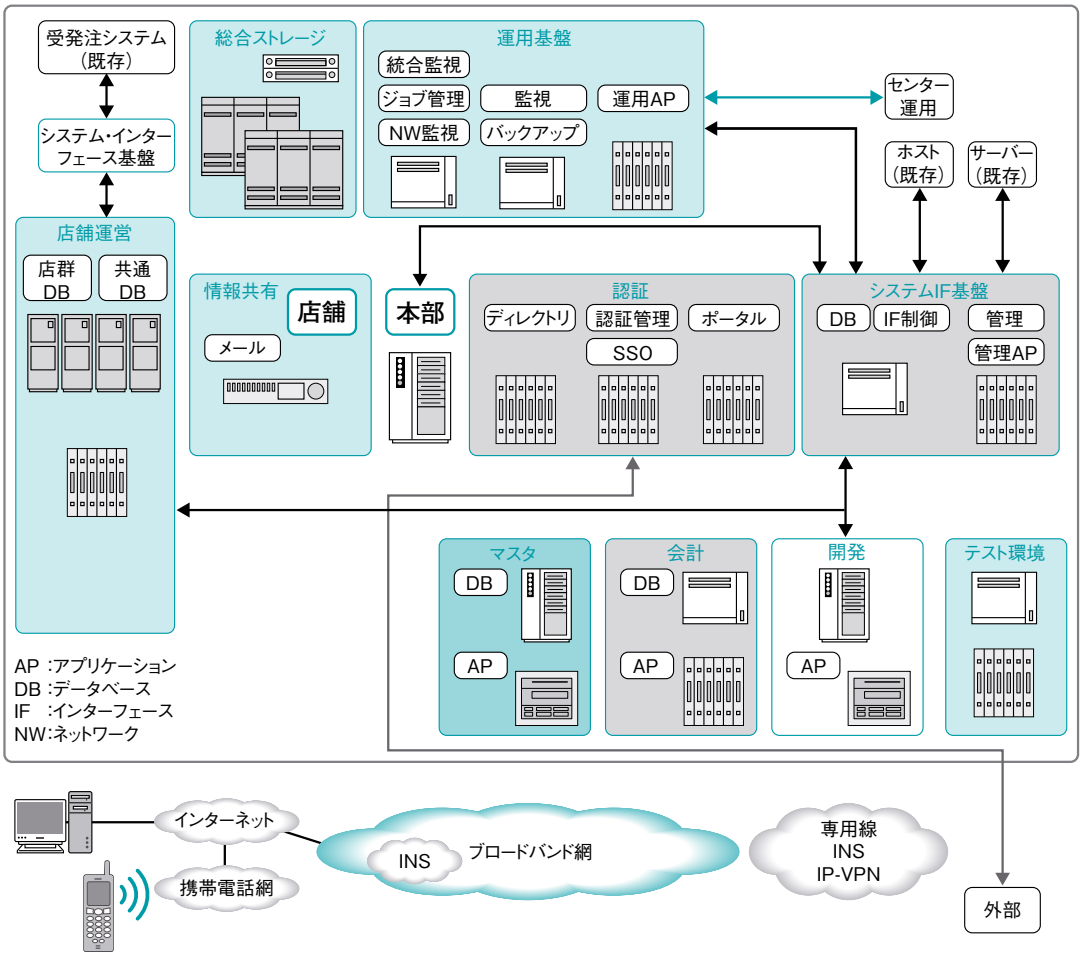
・顧客のかかわり度合い（特にマルチベンダーの場合）
・体制の作り方
・作業分担の仕方
・工数、費用、作業項目の見積り方
・進捗管理の考え方、仕方
・報告、問題提起の方法、対処方法
・各種レビューの方法、各サブシステム担当ベンダー間の調整

そのため、周辺システム構成俯瞰図は、要件定義とともに実現手段の基本となるものであり、大変重要なものである。周辺システム構成俯瞰図は、当然、各工程（上流、中流、下流）の特性に沿ったものでなければならない（**図表3-4**）。各工程により周辺システム構成俯瞰図に求めるものが異なるためであり、単純にインターフェースを示すものであってはならない。

中流工程における周辺システム構成俯瞰図は、実現手段の確実性の観点で作成しなければならない。そのため、

・検証（有用性）
・性能（応答、負荷）
・信頼性確保（品質、正確性）
・障害時対応
・運用性
・互換性
・接続性
・保守性（障害発生時の切り分け容易性）

図表3-4●周辺システム構成俯瞰図の例



・セキュリティ（機密性）
・移行性
・新規性
・経済性
・検証容易性

などの観点から総合的に分類し、その中から該当プロジェクトで必要な項目の見える化を達成できるものでなければならない。これらのうち、新規性、経済性、検証容易性などは、上流工程

でも考慮しておく必要のある項目である。そういう意味で、中流工程における周辺システム構成俯瞰図は、システムを完成させるためには最も重要と言える。

上流工程および下流工程においても、周辺システム構成俯瞰図を作成する必要があったが、中流工程とは観点が異なっている。

上流工程における周辺システム構成俯瞰図は、「機能性（機能配置）」「独立性」「拡張性」などに留意し作成しなければならない。一方、下流工程における周辺システム構成俯瞰図は、テストを実施し、システム品質を確保するために不可欠なものである。中流工程における周辺システム構成俯瞰図の目的は、各サブシステム間のインターフェース（種類、内容、タイミング、エラーの有無など）を検証し、最終的にシステム品質を確保することである。

一般的にシステム間インターフェースの数は、システムの数をNとすると、N（N－1）／2で表され、Nが増えると等比級数的に増加する。そのため、インターフェースが複雑になり、設計、確認検証、そして不具合発生時の切り分けが非常に困難となる。そこで、最も留意しなければならないことは、インターフェースの数を減らし、単純化することである。

3.3.4 システム構成俯瞰図

中流工程におけるシステム構成俯瞰図は、ハードウェアを中心とした機器（開発するソフトウェアをインストールする先）と、その周辺機器との連動情報を記述したものである（**図表3-5**）。中流工程では、データベース・サーバーへのアクセスや、アプリケーションにおけるファイル構成などを検討するため、このようなハードウェア中心で記述する。

3.3.5 スケジュール俯瞰図

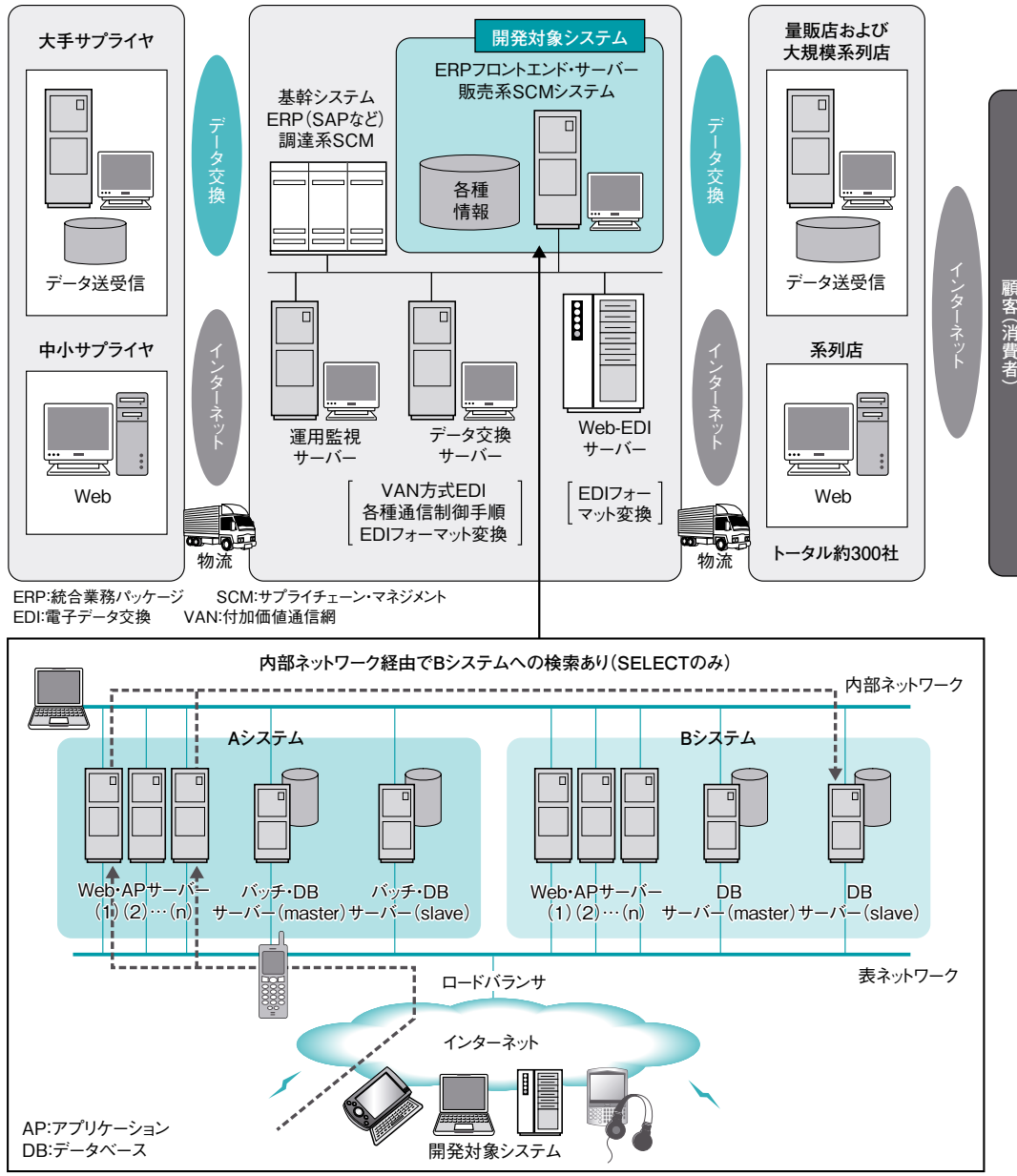
中流工程ではステークホルダーの人数も多くなり、かつ作成する成果物の個数も多くなっている。中流工程では特にスケジュールを詳細レベルにブレークダウンして、実績の進捗および是正処置を取ることが必要である。

そのためにも、大日程からある程度詳細に落とし込んだスケジュールを一目で見ることができる俯瞰図の作成が有効である（**図表3-6**）。

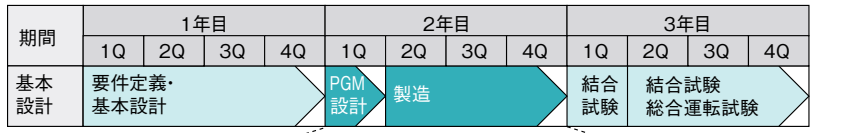
3.3.6 要員遷移俯瞰図

システム設計者のキーパーソンが統合テストを担当すると、テストの品質が高くなるだけではなく、テスト効率もよくなる。さらに詳細設計の実施、管理を担当すると、詳細設計の品質、

図表3-5●システム構成俯瞰図の例



図表3-6●スケジュール俯瞰図の例



現工程のスケジュールは詳細に記述し
後工程のスケジュールは、大枠が決まっている。

項目	工程	担当者	4月	5月	6月	7月	8月	9月	10月	11月	12月	1月以降
オンラインA機能	PGM設計	山田	―									
	製造	加藤		―								
オンラインB機能	PGM設計	山田		―	―							
	製造	加藤			―	―						
バッチC処理	PGM設計	鈴木	―	―								
	製造	山下		―	―							
バッチD処理	PGM設計	田中	―	―								
	製造	山下				―	―					
⋮												

PGM:プログラム

詳細設計の効率も高くなる。また、詳細設計・コーディングのキーパーソンが投入されると、作業の成果物の品質や生産性は高くなる。

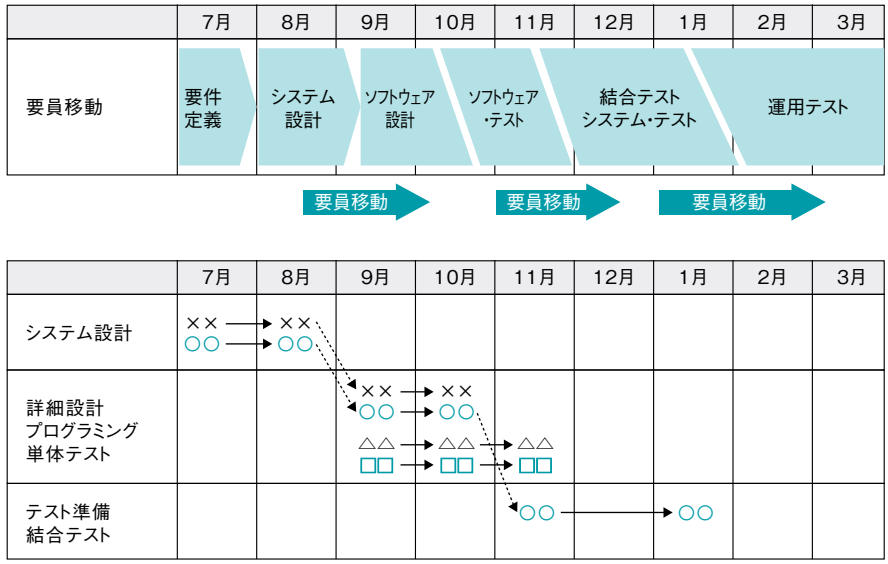
例えば、プロジェクトの現場では、コーディングの進捗に遅れがある場合、本来なら並行して進んでいる別の作業を担当しているキーパーソンを、コーディングの遅れているところに投入する。そのため、キーパーソンの作業のアサイン、作業期間、次の作業、その移行時期を管理することは重要である。特に、納期に間に合わせるために、並行作業で実施するときには、キーパーソンがネックにならないようなスケジューリングが必須となる。

その際、**図表3-7**のように、上流工程でシステム設計を担当したキーパーソンと、詳細設計・コーディングのキーパーソンが、担当すべき作業項目を遷移図の形で整理し、全体を俯瞰できる要員遷移俯瞰図を作成する。これにより、要員の適正配置、過不足の点検に利用できる。

3.3.7 役割分担表

中流工程は、ソフトウェア設計、プログラミング、ソフトウェア・テスト

図表3-7●要員遷移俯瞰図の例



のプロセスからなり、非常に工数、人数が必要とされる工程でもある。各組織に対して果たすべき役割（ミッション）を明確にして、作業を割り当てなければ、組織責任の境界線があいまいとなる。このようなあいまいなところで、問題が発生することが多い。

組織に跨る共通作業は特に重要（優先度高）であるため、対応者、対応作業・範囲などを明確にし、対応者のいない作業がないようにすることが必要となる。中流工程では、多くの組織がプロジェクト内で役割分担を行い、作業を遂行していくことになるため、プロジェクト推進体制俯瞰図より細かい粒度の役割分担を図表にしたものが必要である（**図表3-8**）。

3.3.8 プログラム関連図―作業関連図

上流工程、下流工程と比較して、中流工程では、非常に多くの成果物を作成する。バッチ処理を例に挙げると、1つのバッチ処理は、複数のジョブから成り立っている。また、ジョブおよびプログラムの関連が複雑に絡み合うこととなる。

複雑に絡み合う状態を俯瞰図にする

図表3-8●役割分担表の例

項番	作業項目	発注側	受注SIベンダーA社					協力会社	備考
			PM	ハード・ネットワークG	共通基盤G	設計部	品質保証部		
1	要件定義	○				○			
2	基本設計	○				○	○		(注5)文章表現／体裁チェック
3	詳細設計－オンライン							○	
4	詳細設計－バッチ					○		○	
5	製造					○		○	
6	単体テスト							○	品質保証部での受け入れ検査
7	結合テスト					○	○	○	品質保証部での受け入れ検査
8	システムテスト	○		○	○	○	○		発注側は、別の観点で行う。
9	総合テスト	○	○	○	○	○			
10	性能テスト					○			
11	移行設計	○				○			
12	移行データの検証仕様書	○				○	○		品質保証部レビュー
13	データ移行					○			
14	移行データの補正	○				○			顧客の指示で作業する。
15	移行データ検証	○				○			データ責任は顧客である。
16	本番切替	○	○	○	○	○			作業分担する。
17	運用設計書作成								
18	ハード・ネットワーク設計・構築	○		○					
19	共通基盤設計・構築				○				
20	操作手順書作成					○			
21	業務操作手順書作成	○							
22	操作研修	○				○			
23	プロジェクト計画書作成		○						

ことで、ジョブおよびプログラムの詳細設計、プログラミング、単体テストの優先順位が付けやすくなるとともに、作業漏れをなくすなどに役立てられる。中流工程では、俯瞰図の一部としてプログラム関連図－作業関連図を作成し、利用することが必要である(**図表3-9**)。

3.4 チェックシートを使った見える化

詳細設計・プログラミング・単体テスト段階は、システム自体を作り込む工程であるためプログラム1本ごとの品質を作り込む段階であるといっていい。この工程で品質を高められれば、下流工程でプロジェクトを円滑に推進できる。

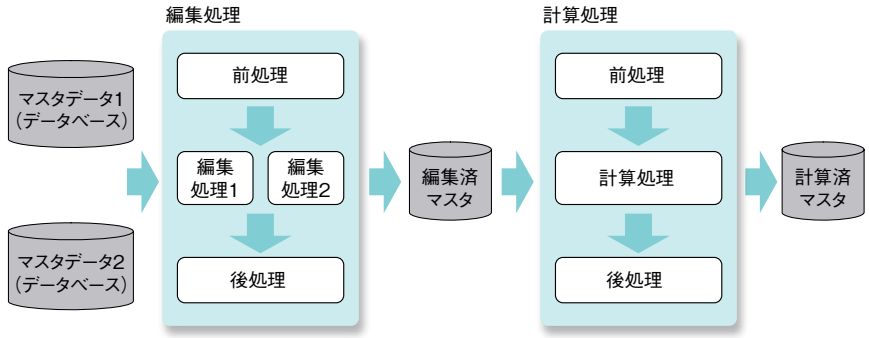
詳細設計書からプログラミング作業を行う際には、コーディングする上で必要かつ十分な内容が詳細設計書に記載されていない限り、プログラムの作成を実施できない。しかし、設計書に、あいまいな記述や必要な記述の欠如、記述内容の矛盾があるにもかかわらず、プログラム作成を実施してしまうことが多いのも事実である。

プログラム作成後、単体テストを行うが、単体テスト項目(Program Check List)がプログラムのコードに対応していなかったり、明らかに少なすぎた

図表3-9●プログラム関連図ー作業関連図の例

プログラム関連図	XXXXX システム	XXXXX サブシステム	XXXXX 業務	作成日 更新日	ページ
				作成者 更新者	

マスタ計算処理（バッチ処理）



作業関連図	XXXXX システム	XXXXX サブシステム	XXXXX 業務	作成日 更新日	ページ
				作成者 更新者	

計算処理（バッチ処理）

機能大分類	機能小分類	処理作成				
		作業1	作業2	作業3	作業4	作業5
マスタ 計算処理	編集処理	前処理	編集処理1	後処理		
			編集処理2			
	計算処理	前処理	計算処理	後処理		
データ 移行処理	マスタデータ1					
	マスタデータ2					

りすると、十分なテストが行われない場合があるのも事実である。このような事態に陥らないために、プロジェクト計画書では、各工程でのレビューの実施などを規定する必要がある。

プロジェクト見える化部会では、プロジェクト・マネージャやプロジェクト・リーダーが自らプロジェクトの問題点やリスクを把握するために、定性的考え方に基づいた項目を集めた「自己評価シート」を用意した。

また、専門家によるヒアリングによりプロジェクト外から客観的に診断を行うことでプロジェクト・マネージャの認識を確認できる「ヒアリングシート」も用意している。

この2つのチェックシートを使用することにより、プロジェクト・マネージャが認識できない問題を明らかにできる。2種類のシートは、付録資料として掲載した。実際のプロジェクトで使用する場合は扱いやすいように、電子ファイル（Excel）をSECのWebサイトからダウンロードできるようにしているので活用して頂きたい。

3.4.1 チェックシートを用いた見える化の流れ

このチェックシートで想定していることは、①詳細設計の終了後に利用、②利用者はシステム開発を行っている受注側のプロジェクト・マネージャ、である。チェックシートを用いるきっかけは、次の2つである。

①プロジェクトの主要メンバー（プロジェクト・マネージャ、プロジェクト・リーダー）が問題点やリスクを明らかにしたい場合
②PMOやプロジェクトを監視する立場の者（専門家チーム）がプロジェクトの問題点やリスクを明らかにしたい場合

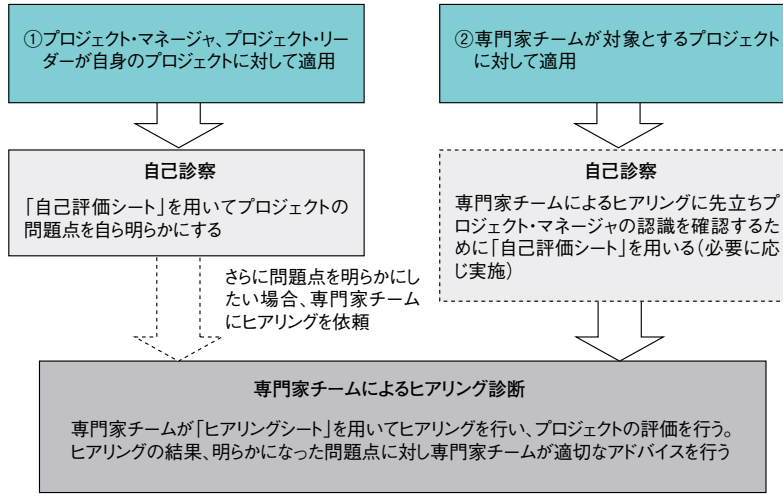
①の場合は自己評価シートを使用し、②の場合は自己評価シートならびにヒアリングシートを使用する。チェックシートを利用して自己診察およびヒアリングを行う流れを**図表3-10**に示す。

3.4.2 自己評価シート

自己診察を行う際に利用する「自己評価シート」のイメージを**図表3-11**に示す。自己評価シートには、38のチェック項目を用意している。

チェック項目に対応して評価入力欄を用意しており、3段階で診断結果を入力する仕組みとなっている。すべてのチェック項目に対する入力が終われば、**図表3-11**の下方に示すようなレーダーチャートが表示される。

図表3-10●チェックシートを用いた見える化の流れ



このレーダーチャートは知識エリア別に表示しているので、プロジェクトでの弱点となる知識エリアがひと目で分かるようにしている。

自己評価シートには、チェック項目や評価記入欄の他にプロジェクト・マネージャへのヒントや問題に対する対策案を載せている。これらの項目に対する説明を**図表3-12**に示す。

3.4.3　自己診察の実施方法

自己診察は、プロジェクト・マネージャがチェック項目に対応する「評価基準」と「マネジメントにおけるヒント」を理解した上で「評価記入欄」に**図表3-13**の要領で入力を行う。

チェック項目の記述内容は汎用的な言葉を使用しているので、ほとんどのITプロジェクトで利用できるが、評価するプロジェクトによっては言葉の定義が異なる場合があるため、言葉の読み替えが必要になる場合もある。

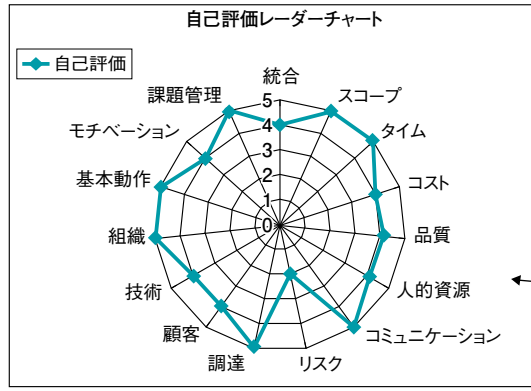
読み替えもできない場合は、「対象外」の「-1」を「評価記入欄」へ入力する。

「評価記入欄」への入力を行うと、判定欄に**図表3-14**の内容で判定結果が表示される。すべての「評価記入欄」への入力が完了すると、**図表3-11**の下方に示したようなレーダーチャートが表示される。各チェック項目に対する判定結果より、空欄（問題なし）以

図表3-11●自己評価シート

NO.	知識エリア	チェック項目	評価基準	マネジメントにおけるヒント	評価記入欄	影響度	評点	最大評点	判定
S1	統合	プロジェクト計画書が作成され、レビューされているか さらに、これに基づき実行されているか	●プロジェクト計画書は、プロジェクト開始から規定期限以内に作成され、レビューされていること ●プロジェクト計画書は、ひな型になるような過去事例や社内標準などをもとに作成していること	●計画の無い所には、マネジメント無し ●レビューはステークホルダー間の合意形成の場である。プロジェクト計画書のレビューを通じた合意は最初の、最も重要な合意事項である ●工期、品質（重点機能）などの目標値をどの範疇にいれるのかなど、ターゲットを明確化する		1	0	3	A
S2	統合	プロジェクトの全体像を俯瞰できているか キーマンや重要機能が規定されているか	●今回の開発対象となるシステムを含めた、顧客側の関連システムを表すドキュメント（システム構成俯瞰図など）を作成していること ●顧客や開発側の関係会社（協力会社など）に関係する体制図があること ●ドミナントアイテム（プロジェクトの成否を左右する支配的要因）を把握できていること	●プロジェクトの全体像を俯瞰できないと、プロジェクト計画がきちんとできていない可能性がある ●完全なものが作成できなくても、トライすることが重要である。トライの積み重ねとその後のプロジェクト・マネジメント経験の反省を通じて、俯瞰図作成レベルは着実に向上する		1	0	3	A
S3	統合	納入物件および必要な作業成果物が明確となっているか 契約書との比較は済んでいるか	顧客との間でどういう記述方法で、どういうドキュメントを納品するかが明確になっていること	●顧客との間で成果物の質と量が明確でないとその後の作業の割り振りや規模の見積もりなどで大きな判断ミスをする可能性がある ●事前に成果物サンプルを提示することで、納入物件に関する双方の共通認識を確立することが望ましい ●各種計画書や仕様書、運用テストの項目一覧など、具体的な成果物のサンプルを示すことで、どのような作業が必要になってくるかを事前に検討したり、要員のアサインを考えたりしやすい ●各成果物のボリューム観の顧客との認識あわせ		1	0	3	A
S4	統合	前段の段取り作業を含めてすべての作業項目がスケジュール化されているか	●WBS が書き出されており、その内容について十分検討されていること ●すべての WBS がスケジュールに落とされていること	●必要な作業項目を洗い出すために、例えば各社で標準化している WBS のひな型を利用したり、過去の他のプロジェクトを参考にするなどの工夫をすること ●システム構築のための設計書作成やプログラム開発作業だけではなく、準備作業、顧客との調整事項など、想定される作業をなるべく網羅的に洗い出すこと ●WBSをもとにしてスケジュール中の各作業の前後関係の整合が取れていること		1	0	3	A

自己評価の結果を1～3の数字で記入、
判定結果がレーダーチャート･グラフで出力される



プロジェクトの問題が
ひと目で分かる

図表3-12●自己評価シートの項目説明

項目	項目説明
No	“S” で始まる連番が付けられている。チェック項目を識別するのに使用する
知識エリア	チェック項目が属する15種類の知識エリア名。PMBOKの9つの知識エリア（統合、スコープ、タイム、コスト、品質、人的資源、コミュニケーション、リスク、調達）に加え、PMBOKにヒューマンウェアの部分を追加している。追加しているのは、「顧客」「技術」「組織」「基本動作」「モチベーション」の5つとプロジェクトの監視コントロールに注目した「課題管理」である
チェック項目	質問の内容が記述されている。この質問に従って、あとの評価記入欄に結果を入力する
評価基準	チェック項目の質問を補足する。具体的な評価基準の例が記述されている
マネジメントにおけるヒント	チェック項目を評価する際のポイントとともにマネジメントを行う際に気付きを与えるヒントが記述されている。特に自己診察のみ実施する場合にはこの項目をよく読んでチェック項目の趣旨と何を気づかせようとしているのか理解する必要がある
評価記入欄	この欄に評価結果を1～3の数字で記入する。このチェック項目が評価対象のプロジェクトに該当しない場合には－1を記入する。評価結果入力方法は【3.4.3項】で説明する
判定	判定結果がA、Bまたは空欄にて表示される。判定の見方は【3.4.3項】で説明する
対策案	判定がAまたはBの場合の対策案が記述されている。対策案は、一般的なプロジェクトを基に考えられているので、評価対象プロジェクトでは読み替えが必要な場合もある

図表3-13●自己評価シートの評価基準

記入欄数字	評価基準
1	チェック項目のマネジメント方法を知らない
2	チェック項目のマネジメントとして何をしなければならないかは分かっているが、種々の事情によりうまくできていない
3	チェック項目のマネジメントとして何をしなければならないかが分かっており、うまくできている
－1	対象外。評価対象のプロジェクトには、チェック項目に該当するものがない

図表3-14●自己評価シートの判定

判定	判定の意味
A	対策が必要な項目
B	注意が必要な項目
空欄	問題なし

外の項目においては対策や注意が必要となるため、対策案欄の内容を理解し、対象プロジェクトの現状に照らし合わせて対応策を考える必要がある。

レーダーチャートの結果より、対象プロジェクトの弱点分野が導き出されているので、弱点分野の強化策も対象プロジェクトの現状に照らし合わせて考える必要がある。

プロジェクト・マネージャ側で自己評価が不十分である場合は、専門家チームにヒアリングを依頼して評価してもらうことも望ましい。

3.4.4 ヒアリングシート

プロジェクトの問題を明らかにするには、自己評価シートを使った「自己診察」の他に、PMOやプロジェクトを監視する立場の者（専門家チーム）がプロジェクト・マネージャを対象にプロジェクトの現状を「ヒアリング」することが望ましい。ヒアリングの流れは**図表3-15**のようになる。

ヒアリングを行う際に利用する「ヒアリングシート」のイメージを**図表3-16**に示す。

ヒアリングシートには、78のチェック項目を用意している。チェック項目に対応して評価入力欄を用意しており、5段階で評価を入力する仕組みとなっている。

すべてのチェック項目に対する入力が終われば、**図表3-16**の下方に示すようなレーダーチャートが表示される。このレーダーチャートは分野（知識エリア）別に表示しているので、プロジェクトでの弱点知識エリアがひと目で分かるようにしている。

ヒアリングシートは専門化チームがヒアリングを実施する上で扱いやすいように作成されているため、自己評価シートにあるようなプロジェクト・マネージャに気付きを与えるチェック項目はないが、ヒアリングの要領やエビデンス（プロジェクト資料）などによる確認方法を記載している。これらの項目に対する説明を**図表3-17**に示す。

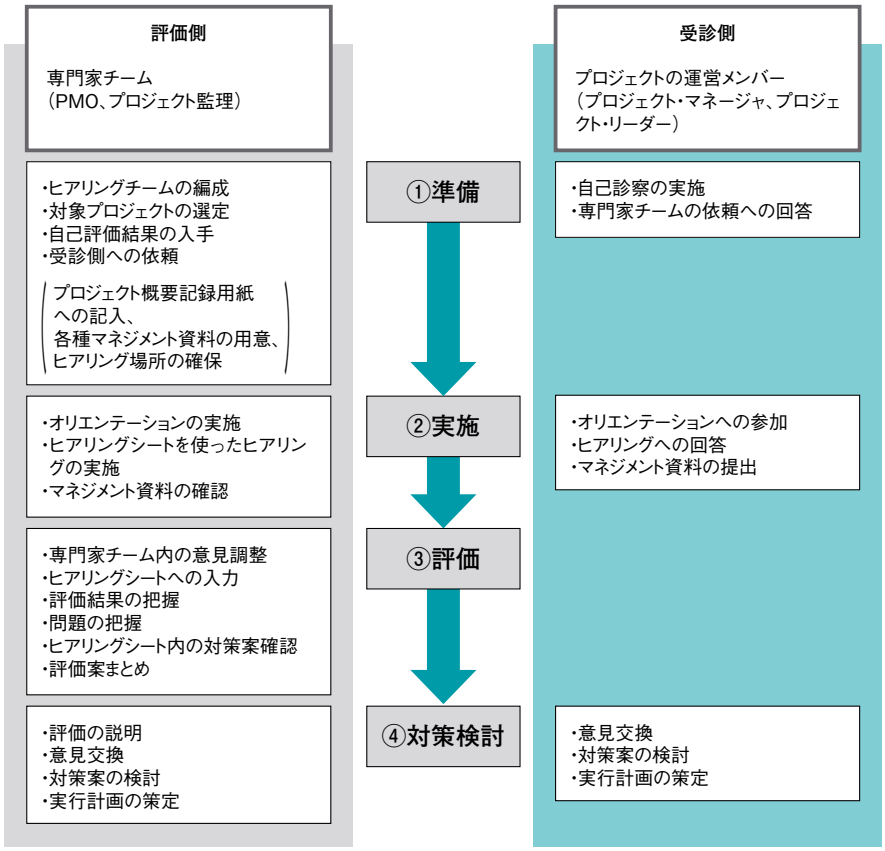
3.4.5 ヒアリングの実施方法

ヒアリングの流れは**図表3-15**に示した通り、評価側（専門家チーム）と受診側が一連の行動（①準備→②実施→③評価→④対策検討）を協力して行う。

①準備

ヒアリングの実施する専門家チームは、**図表3-18**の編成を取り、受診側のプロジェクト・マネージャに、「自己評価シートの提出」「プロジェクト概要書の作成と提出」「各種マネジメント資料の提出」「ヒアリング場所の

図表3-15●専門家チームによるヒアリングの流れ



確保」といった作業を依頼する。

②実施

評価側は、実施前にプロジェクト・マネージャが準備した各種資料を理解する必要がある。

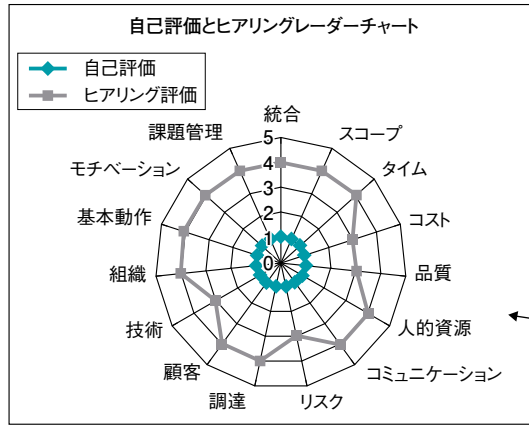
ヒアリング実施に当たり、まず10分程でオリエンテーションを行いヒアリングの目的や手順を話す。オリエンテーションを終えてからヒアリングの実施となる。

ヒアリング担当者（専門家チーム）は、ヒアリングシートに記載されているチェック項目をプロジェクトの現状

図表3-16●ヒアリングシート

NO.	知識エリア	チェック項目	個別のヒアリング要領	評価基準	エビデンス・確認方法	評価記入欄	範囲	エビデンス	判定
H1	技術	通信制御系・上流工程の成果物の確認	要求仕様、機能仕様書が、他の業務システム並みの文章、一般図表で書かれているだけでは不十分。中流工程以降、開発規模が過大とならないか、考慮漏れは無いかといった視点でヒアリングを実施	●スコープが有限範囲であることを全体を俯瞰しながら確認できる(1枚の状態遷移図に書ききれる)こと ●各状態ですべての内外トリガー(コマンド/オーダー受信など)を考慮した設計となっていること	状態遷移マトリクス(最低でも状態遷移図)		2	1	A
H2	技術	多重アプリケーション・プログラム走行環境におけるデッドロックも想定し詳細設計したか	特に旧資源へのアクセスを行うプロセスは、それだけ資源保留期間が長くなり、多重アプリケーション・プログラム走行環境ではデッドロックに遭遇する確率が高まることにも注意する	●デッドロック対策が明確であること ●対策実施後のデッドロック発生事象を網羅的に列挙し、その影響を評価していること	詳細設計書とそのレビュー証跡		2	2	A
H3	技術	ネットワークは十分な容量設計を行ったか	ネットワークは、異常時パケットの再送などで指数関数的にトラフィックが急増することもあるため、ネットワーク技術に長けたSEの協力を得て、設計する必要がある	技術的根拠を持ってネットワーク設計を行っていること	詳細設計書とそのレビュー証跡		2	2	A
H4	技術	アプリケーション・プログラムの走行性能低下が発生する要因と、その影響を調査・検証しているか	アプリケーション・プログラムの単体走行時性能に加え、アプリケーション・プログラム多重走行環境でアプリケーション・プログラム同士が同時には利用できない共用資源など SRR(Serially Reusable Resources) を調査し、それによる性能低下が無いか検証していること。また、特に高レスポンスが要求されるアプリケーション・プログラム・プロセスの実行優先度が高くなるように設計していること	●SRR の抽出方法が、思いつきではなく、論理的・体系的であること ●SRRへのアクセス頻度を加味した影響を、客観的手法で査定していること ●システム性能要件と実行優先度の設計とが整合していること	詳細設計書とそのレビュー証跡		2	2	A

評価結果を1～5の数字で記入、判定が表示される



診断結果がわかりやすくレーダーチャートで表示される

に合わせた上で、言葉の読み替えなどを行いながら質問を受診側（プロジェクト・マネージャ）に行っていく。

質問の際には必要に応じて、事前に用意されている各種資料を活用するとともに、実際に行われている証跡（エビデンス）を確認するなどの補足的行動を行う。

図表3-17●ヒアリングシート項目

項目	項目説明
No	“H”で始まる連番が付けられている。チェック項目を識別するのに使用する
知識エリア	チェック項目が属する15種類の知識エリア名。PMBOKの9つの知識エリア（統合、スコープ、タイム、コスト、品質、人的資源、コミュニケーション、リスク、調達）に加え、PMBOKにヒューマンウェアの部分を追加している。追加しているのは、「顧客」「技術」「組織」「基本動作」「モチベーション」の5つとプロジェクトの監視コントロールに注目した「課題管理」である
チェック項目	チェックする質問が書かれている。ヒアリング時にこのチェックに対する回答を評価記入欄に入力する
個別のヒアリング要領	ヒアリングを行う際のポイント、チェックする内容の詳細、そのチェック項目の確認方法が記載されている
評価基準	チェックする際の評価基準が説明されている
エビデンス・確認方法	チェック項目を判断する際に確認すべきエビデンスの種類がリストアップされている
評価記入欄	チェック項目に対する回答を1～5の数字で記入、具体的な入力方法は【3.4.5項】で説明する
影響度	チェック項目が判定に与える影響度合いを示す。この値が大きい（1.0）項目はよりリスクが高いと見なされる。チェックシートに記載されている値は、リスク分類表（第5章で説明）へのマッピング度合いや、チェック項目の評価の具体性を考慮して重み付けされた値が設定されている
判定	チェック項目に対する判定をA、B、空欄で判定する、判定内容は【3.4.5項】で説明する
対策案	チェック項目に対して問題があった際の対策案が記述されている、プロジェクトによっては読み替えが必要な場合もある

図表3-18●専門家チームの編成

役割	内容	経験または能力
ヒアリング担当者	専門家チームのリーダーとしてヒアリングを実施する	豊富なプロジェクト・マネジメント経験を持っていること。プロジェクトで「今後起こる問題」を見通す能力が求められる
観察者	ヒアリング対象者の表情・仕草などを観察する。ヒアリング担当者の兼務も可能	
記録者	ヒアリング結果を記録する	

これにより、評価側がプロジェクトの現状把握を行い、ヒアリングシートの評価記入欄へ評価レベル（**図表3-19**を参照）を記録する。

③評価

ヒアリングを終えると対象プロジェクトの「評価」に入るが、ヒアリングの場所とは違う場所で評価者のみによる評価会を開く。ここではヒアリング実施中に記入したチェック項目に対する「評価入力欄」の値が妥当であるかを評価者全員で見直しをかける。

入力した「評価記入欄」の値から「判定」欄の値が導き出されるわけだが、**図表3-16**に示したヒアリングシートからも分かる通り、この2つの項目の間には「範囲」「エビデンス」「影響度」といった欄が存在する。

「影響度」はプロジェクトに対する「重み付け」を設定できるようになっており、「範囲」と「エビデンス」の値を基に算出している。

よって、「評価記入欄」と「影響度」を独自の計算方法で掛け合わせて「判定」を導き出している。

- 「範囲」欄：チェック項目が明らかにしようとするリスクのばらつきを調整するもの
- 「エビデンス」欄：チェック項目に対して具体的、客観的な回答を求める度合い
- 「影響度」欄：影響度要因（範囲、エビデンス）の値は、チェック項目ごとに合計し、統合した影響度を算出する（4点以上：1.0　3点：0.7　2点：

図表3-19●ヒアリングでの評価レベル

評価レベル	評価基準
5	質問領域のプロジェクト・マネジメントがほぼ完璧にできている
4	質問領域のプロジェクト・マネジメントは分かっているが、内部・外部の要因によって、そのすべてはできていない
3	質問領域のプロジェクト・マネジメントは分かっているが、ほとんどできていない
2	質問領域のプロジェクト・マネジメントが分かっていないまま、勘と度胸のマネジメントを実施
1	質問領域のプロジェクト・マネジメントが分かっていない。しかも、的外れなマネジメントを実施
－1	対象外、当該プロジェクトには当てはまらない質問であり、読み替えることもできない

0.5)

・「判定」欄：「評価記入」と「影響度」を独自の計算方法で掛け合わせた結果。判定の意味は自己評価シートと同じ（**図表3-20**を参照）

入力が終わると、レーダーチャートに知識エリアごとの判定結果が表示される（**図表3-21**参照）。

ヒアリングの実施前に自己評価シートを準備した場合は、自己評価シートおよびヒアリングシートの判定より乖離の状態がレーダーチャートとグラフに表示される（**図表3-22**参照）。

専門家チームは、以上の結果から問題点のチェック項目を洗い出し、ヒアリングシートの「対策案」欄を参考にして、問題点をまとめる。問題点が明らかになったら評価結果と問題点をまとめ、評価案を作成する。

④対策検討

評価案ができたら、受信側プロジェクト・マネージャ、プロジェクト・リーダーに評価の結果説明を行う。説明会は、ヒアリングに参加した専門家チームとヒアリングを受けた受診側メンバー全員が出席する。

説明後、評価側と受診側が協同して問題点の対策を検討する。

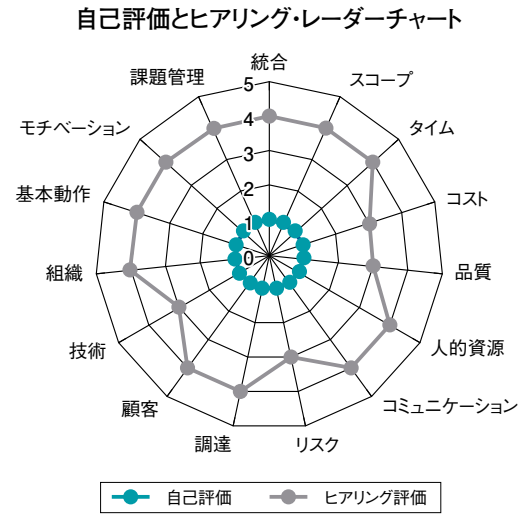
3.4.6　チェックシートの改良

自己評価シートおよびヒアリングシートは、IT業界の動向、自社の経営状態、自部署の状態などを勘案して、常にブラッシュアップを行う必要がある。ブラッシュアップを行わないとチェックシートの内容が現状と乖離の幅が大きくなり使い物にならなくなってしまうからである。

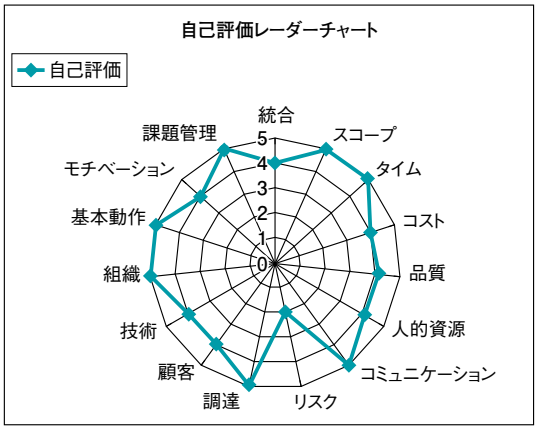
図表3-20●ヒアリングシートの判定

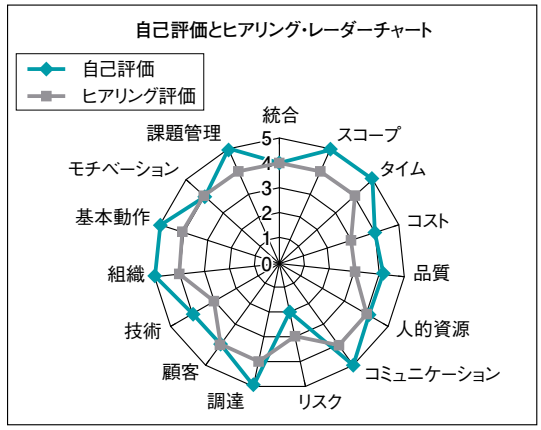
判定	判定の意味
A	対策が必要な項目
B	注意が必要な項目
空欄	問題なし

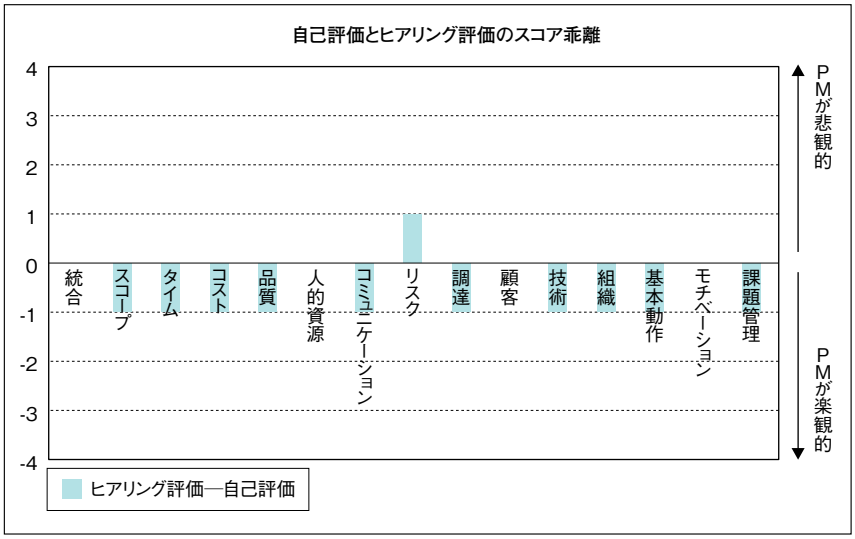
図表3-21●ヒアリングシートのレーダーチャート



図表3-22●自己評価シートとヒアリングシート入力後の評価







プラスに乖離が大きい場合、ヒアリング評価のほうが、自己評価よりも高いことを表している

マイナスに乖離が大きい場合、ヒアリング評価のほうが、自己評価よりも低いことを表している

3.5 中流工程事例集

3.5.1 事例集の公開

上流工程では、中流工程以降に予測されるリスクの発生とその影響拡大を防ぐために、リスクの兆候に重点を置き、問題事例を整理した。

これに対して、中流工程では上流に比べ、リスクは兆候というよりも、より具体的な姿として現れる。プロジェクト・マネジメント国際会議（IPMA）では上流から中流へとフェーズが変わる際に、プロジェクト・マネージャが直面するリスクも変わっていく様をリスク成長過程として議論している（参考文献[12]）。

例えば、詳細設計段階で予定よりも膨らんでしまった要求仕様の詰めに難航するプロジェクト・マネージャの姿、それをプロジェクト・マネージャの能力不足として非難する顧客の姿、他の新規案件ばかりに気が移りプロジェクト・マネージャの苦境を省みない上司や営業の姿……などである。

より具体的な姿を現し始めたリスクに対して、中流工程においては、早期の対策を講じる必要がある。ところが多くの問題事例が物語るように、中流工程において誰が何を行うべきか、必ずしも正しい判断が行われない場合が多く、このことが後に下流工程にて大問題へと発展する要因ともなっている。このような考えから、中流工程の事例集の整理に当たっては、上流工程編と比較し、対策に「誰が」の情報も明確にし、一層の見える化を図ってみた（**図表3-23**）。

対処例では、大手システム・インテグレータ、ソフト開発を行うITメーカーから中小ソフトハウスまでソフト開発を行うすべての企業である「SIベンダー」と、SIベンダーに対する発注を行う企業「顧客」とを区分することで、「誰が」を明確にした。

各項目についての概要説明は**図表3-24**を参照して貰いたい。この事例では、上流工程編と同様に「見切り」という言葉を使っている。これは「問題を予測したときに、どのような判断を下したか」「問題に対し何をしたか」というように、必ずどの場面においても、何らかの事象に対して判断（見切り）が入り、そしてその見切りに対する行動が発生する。問題の発生を予測できて、かつ見切りが適切であり、行動が取れることで、問題の発生はより抑止できるようになるはずである。

本書では、下流工程編、上流工程編に続き、中流工程での新たな失敗事例として、58件を付録で公開する。収録した事例のタイトル一覧を**図表**

図表3-23●事例サンプル

1 仕様変更による「品質問題」

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。

事例における見切りの内容

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。

本来の判断の考え方

・変更計画でも、プロジェクト計画ベースライン見積りについて、初期計画と同じ精度を確保しない限り、リスクも正確にあぶりだされないと判断するべきだった。

対処例

SIベンダー
・まずPMは緊急機能追加に見合う、プロジェクト負荷軽減策（既計画機能の納期変更、緊急機能の段階リリースなど）について顧客と折衝する。
・それでも無理があるのであれば、PMではなく、PMの上司自らが顧客や顧客の上司にリスクを伝え、顧客の対応を要請する。

顧客
・リスクを伝えられた顧客が、それでも仕様変更を決行する場合には、ベンダー任せで済ませず、リスクの発生に備えた、コンティンジェンシー・プラン（システム迅速復旧体制など）を自ら用意する。

顧客とPM/上司/営業の関係を「発注元―業者」の関係ではなく、プロジェクトリスクに関しては、双方が「リスクと闘うパートナー」となるような、リレーションシップを受注～上流工程前に構築しておくこと。

図表3-24●事例項目の概要

事例番号、タイトル、概要	事例に付けられた通番とタイトル、トラブルの概要を説明
事例における見切りの内容	プロジェクト・マネージャが、その事例でどのような見切りを行って、プロジェクトを推進したか、またそのときの状況
本来の判断の考え方	プロジェクト・マネージャが本来はどのような考えのもとで判断すればよかったのかの説明、および補足コメント
対処例	見切りを行うときの対処例を述べる。対処例ではプロジェクト・マネージャが所属する SI ベンダー側の立場、発注元の立場、並びにそれぞれの立場を超越した場合に分けて、対策内容を記載

3-25に示す。

3.5.2 問題の発生傾向と対策

プロジェクト見える化部会が収集した問題事例や、部会の委員が直接体験した事例を分析したところ、いくつかの特徴が見られた。

分析に当たっては、IPMAで報告されている大問題プロジェクトに関するリスク成長過程の考え方を参考にした（参考文献[12]）。すなわち、上流工程、中流工程、下流工程の各工程における

図表3-25●事例タイトル一覧

事例番号	事例タイトル
1	仕様変更による「品質問題」
2	仕様変更による「予算超過」
3	仕様変更による「納期遅延」
4	仕様変更による「システム老化」
5	営業が中流工程以降身を引き仕様変更で「不採算」に
6	仕様変更の連続で出口が見えないプロジェクト、メンバーが疲弊
7	変化即応型の開発を Scope 変更ルールが無く請負った
8	変更要求問題で、PM の交渉力、報連相が不足
9	開発中のリスクについて PM が上司にエスカレーションをあきらめた
10	PM・顧客間不信頼
11	仕様変更を一切受付けないことによる問題
12	複雑な顧客ステークホルダー組織で仕変が多発
13	仕様凍結確認せず中流工程に着手
14	マイグレーションの要件、実質はシステムレベルアップ。仕様追加極大
15	世代交代で顧客、SI ベンダーともに熟知した要員不在。マイグレーション仕様固まらず
16	正式要求仕様書無く作業。仕様あいまいにより変更多発
17	当初納期の大幅前倒し要求を受け付け QCD 問題に
18	当初価格の大幅削減要求を受け付け QCD 問題に
19	過剰な責任所在文化による契約締結遅れ、物事のあいまいさ
20	顧客責任部門、責任者不明
21	変えられない既存業務プロセスにパッケージを導入
22	システム共同化、再構築で自社独自のテスト不十分。重大障害発生
23	システム共同化を目指すが、先行ユーザーのサービスインの遅れで納期遅延
24	ハードウェア、OS 更改でテスト必要認識不足。重大障害発生
25	スケジュールが遅れ、業務主体のテストとなり、基盤・運用系不安定
26	障害の数の定量的管理にとらわれ過ぎ、品質を誤評価
27	システム統合の場合、企業文化の相違によるコンティンジェンシー・プラン不備
28	外部委託の SLA 不備
29	ソフト開発量の膨張で納期大幅延伸
30	性能の異なる新旧資源混在環境でのトラブル
31	異常時挙動を想定しない設備設計
32	基本設計書の品質不足の問題
33	仕様確定遅れによる問題
34	アプリケーションプログラムの単体品質にばかりとらわれ、システム全体の納期・品質が守れず

35	業務アプリケーション・プログラム性能にとらわれシステム性能問題を見落とす
36	仕様通りに開発したが運用障害で事業が長時間停止
37	システム移行設計不良
38	製品間インターフェース保証の見落とし
39	有名ソフトを未検証のまま使用し不具合発生
40	営業主導で決めたソフトで開発し要員不足、クレーム、不具合発生
41	パッケージ採用と言いながらカスタマイズ量過多（実質固有ソフト）
42	パッケージ機能の認識、期待感のギャップ大。規模、工数、費用、納期大幅ズレ
43	特定顧客で未完パッケージを完成させる計画が頓挫
44	システム全体としての信頼性設計の不備
45	過負荷・限界系の見落とし
46	方式性能ボトルネックの見落とし
47	最新ソフト、バージョンを検証不足のまま採用。不具合多発
48	自前ソフト以外の検証、確認認識薄く、後工程で問題発生
49	アプリケーション・プログラムの生産性で言語を選定したが、性能問題がネックとなった
50	開発ルール未確定のため保守性、流用性が悪い
51	新しい言語の熟練者不足でプログラム完成度が低い
52	客観的な生産性指標が少なく、工数、費用にインパクト大
53	担当者にシステム的認識欠落、障害時の切り分け、情報採取の仕組み不足
54	オフショアで品質と納期遅延の問題（発注側問題）
55	オフショアで品質と納期遅延の問題（受注側問題）
56	外注会社責任者は担当者に任せ切り
57	特定ノウハウについて外注依存。SI ベンダーのコントロール不可
58	行動規範順守が困難。機密情報漏えいの問題

負のスパイラルに対して、これと直接的な因果関係が見られる事象を

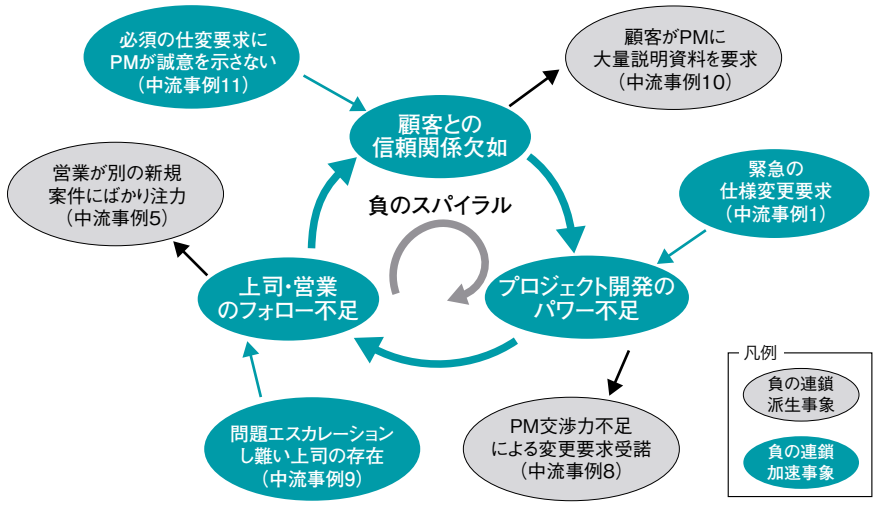
①負のスパイラルから派生して起こる事象（以下、派生事象）
②負のスパイラルを加速させる事象（以下、加速事象）

の2つに分ける考え方である。

例えば、中流工程では**図表3-26**のような相関関係がある。

大問題に発展したプロジェクトを分類し、発生件数を調べ、派主事象、加速事象に分類した。また、負のスパイラルとは直接的な因果関係がない事象についても、他の工程に影響を与える潜在原因として事例を収集した。分析の結果を**図表3-27**に示す。

図表3-26●中流工程における負のスパイラルと大問題プロジェクト事例との相関関係(例)



図表3-27●リスク成長過程(負のスパイラル)モデルとの相関関係分析

出典事例集	大問題に発展したプロジェクト件数	負のスパイラルから派生して起こる事象 [件数(比率)]	負のスパイラルを加速させるような事象 [件数(比率)]	負のスパイラルと直接的な相関関係はないが下流工程で問題発展[件数(比率)]
上流工程	39	27 (69.2%)	10 (25.6%)	2 (5.1%)
中流工程	39	5 (12.8%)	17 (43.6%)	17 (43.6%)
下流工程	29	18 (62.1%)	10 (34.5%)	1 (3.4%)
合計	107	50 (46.7%)	37 (34.6%)	20 (18.7%)
		87 (81.3%)		

これによって分かった中流工程ならではの傾向は、次のようなものがある。

①派生事象の比率が全体の1割強と、他の工程の6割～7割弱に比べて比率が低い

②加速事象の比率が全体の4割強と、上流工程の2割強、下流工程の3割強よりも高い

③負のスパイラルと直接的な相関関係は観測されないが、中流工程で潜在し、下流工程以降で最終的に大問題へと発展する事象が4割強と、他の

工程の数％に比べてかなり高い

①②の主な要因は、中流工程における突発的な要求仕様の変更であり、仕様変更の事象が負のスパイラルを加速させる危険性を孕んでいることを示している。

また③の主な要因は、抽象的な要求仕様を詳細設計以降で具体的な情報システム作りに落とす段階で混入する技術的な問題である（以下、中流工程におけるSI技術の問題と呼ぶ）。

以下、「要求仕様変更」と「中流工程におけるSI技術の問題」に分けて、対策を述べる。

3.5.3 要求仕様変更

プロジェクト・マネージャにとって中流工程における大きな課題の1つが仕様変更である。仕様変更が発生する原因としては、以下の2つである。

①上流工程における検討の詰めが甘い
②顧客側事業が対象とする市場ニーズの変化への即応

①の検討漏れは、システムの要件やシステム構造の複雑化によって、なかなか減ることはないが、事例集から読み取れるように、SIベンダー側責任だけでなく、顧客側責任あるいは双方の責任となる場合も多い。

②は近年増えつつある。②の変化即応型の仕様変更は、開発当初から予見することは困難である。しかし、既に決まっているプロジェクト計画に、仕様の追加・変更などを無理やり押し込むことにはリスクが伴う。公表したサービス開始時期に間に合わなくなったり、サービス開始後の品質不良が多発したりする多くの事例の原因を遡ると、やはりSIベンダーだけでなく顧客側、あるいは双方の責任に帰結する場合も多い。

顧客とSIベンダーとの関係については、従来の「発注元―業者」の関係ではなく、リスク軽減や事業の「パートナー」の関係を基盤にすることが大切である。本来なら上流工程でこの基盤を構築しておくことが望ましいが、中流工程でもその基盤をより強固にすることが、プロジェクトを成功に導いてくれる。

また、SIベンダーのプロジェクト・マネージャとしては、仕様変更に対する、厳格なリスク査定と顧客とのリスク対応策に関する交渉が特に重要であり、その実践力が問われる。

しかし、多くの事例から読み取れるように、プロジェクト・マネージャの実践力には限界があることがあり、この場合、プロジェクト・マネージャからの問題エスカレーションの風通しを

よくする部門コミュニケーション・マネジメントの工夫などSIベンダーのステークホルダー（プロジェクト・マネージャの上司、営業）も含めたSIベンダー組織としての対策も必要である（参考文献[13]）。

3.5.4 中流工程におけるSI技術の問題

プロジェクト・マネージャにとって中流工程で解決しなければならないもう1つの課題は、固まった要件を現実のシステムとして実装できる保証を得ることである。実装の保証という観点でいうと、上流工程では「森を見て木までは必ずしも見ていない」のに対して、中流工程では「木を一本一本、検証する」ことが要求される。特に大規模システムが実装可能であることの検証では、「見るべき木」が極めて多岐にわたり、1つの見落としが大きな問題に発展することから、中流工程で様々な観点から多角的かつ体系的な検証が必要となる。

例えば、ソフトウェア開発の実装可能性の検証事例（事例番号29）のように、下流工程の試験で大規模なロジック漏れが多数発覚し、予定より開発規模が大幅に増えたことにより、実装が納期に間に合わなかった事例もある。分野により様々であるが、通信制御ソフトなどの場合、中流工程で詳細な制御の流れを、状態遷移表などを使うなどして実装の網羅性を検証しておくことが肝要である。そのほか、SRR（Serially Reusable Resource）による性能ボトルネックの見落としによる問題（事例番号35）などが多数、付録・事例集に収録されている。

また、逆のことを言うようだが、実装の面で「木だけでなく森も見なければならない」ことも多々ある。例えば、事例番号34などで見られるように、アプリケーション・プログラムなどシステムの個別コンポーネントでは問題がないが、システム全体の信頼性などシステム・ライフサイクルを通した移行、運用といった観点では大きな問題となる事例がある。そのためには、システムの方式技術者（OSやデータベース管理システムなどコンピュータ・システムの「造り」に関する原理原則と、アプリケーション・プログラムもインテグレートした情報システム構築論の両面での知見を有する技術者）の知恵が必要である。「木も森も見てシステムの実装可能であることの確証を得る」ための知恵は、付録・事例集に加え、付録・ヒアリングシート、自己評価シートにできる限り盛り込んだので、これらを活用することを是非お勧めしたい。

第4章

定量的見える化アプローチ

4.1 概要

中流工程は、上流工程の成果物である抽象的な仕様から、実装可能性を検証できるレベルの詳細ドキュメントへの変換工程である。同時に、その詳細ドキュメントをプログラム、データ、運用手順などの物理的な情報に落として、その成果物を下流工程に引き継ぐ工程でもある。

そこでは成果物の品質がプロジェクトの成否を左右する。まず、上流工程の成果物の品質としては、機能の確定度合いを含む基本設計書の品質が求められる。さらに中流工程成果物の品質とは、詳細設計書、プログラム仕様書の品質、プログラムの品質、結合テスト計画書の品質などである。

中流工程のプロセスの品質や人の品質も求められており、具体的には、品質目標、品質保証活動、構成管理といった形で品質向上を図っており、そのための見える化する範囲は広い。

品質だけでなく、上流工程で確定されたスコープの見える化が必要である。スコープの変更は、中流工程でのシステム実現方法の変更、スケジュールの変更につながり、プロジェクト計画書の見直しとなることも多い。

次に、進捗を遅らせないためにタイム（時間）に関する見える化も重要である。進捗に影響する開発環境、テスト環境の充足度、管理状況が該当する。

中流工程では、これらを詳細に見える化し、状況を把握、評価してプロジェクトをこまめにコントロールしていかなければならない。そのためには、定性的アプローチだけではなく、プロジェクトの実際の動きをデータとして捉えることにより、定量的な見える化を進めていく。

4.2 測定項目リスト

本書では、中流工程での定量的見える化のために測定すべき項目のリスト

を、「測定分析データ一覧表」として提供している。

上流工程編の解説書では、測定項目リストとして「測定分析データ一覧表」と「ベース尺度一覧表」の2種類を提供している。「ベース尺度一覧表」は、測定計画を作る際に役立つものであり、上流工程編の「ベース尺度一覧表」を参考にして、中流工程のプロジェクトの測定計画に適合した「ベース尺度一覧表」を作成することをお勧めする。プロジェクトの状況を定量的に見える化するためには、測定目的を明確にして、測定項目を決める必要がある。「測定分析データ一覧表」は、測定目的とそれの対応した導出尺度（測定項目）をリストアップして、知識エリアで分類したものである。

測定分析データ一覧表には、「測定の目的」「重点項目」「利用者」「導出尺度とその定義（式）」「導出尺度の見方、分かること」といった項目が示されている（**図表4-1**）。

この一覧表は、測定目的から測定すべき項目（導出尺度）を求める際に活用する。また、「導出尺度の見方、分かること」に記述されている内容から、測定した導出尺度の値、傾向を見て次工程以降に起こり得る問題、発生している問題の原因を見つけられる。

ただし、測定分析データ一覧表に列挙した項目のすべてを測定することは、管理負荷が高くなり、現実的には難しい。そこで本来押さえておくべき測定項目について、測定分析データ一覧の「重点項目」欄に★印を付記した。しかし、1つの導出尺度だけを見ていてもだめで、複数の尺度や定性的に捉えたプロジェクトの状況を統合的に評価する必要がある。

4.3 導出尺度の見方と分かることの例

測定した導出尺度の見方と分かることの例を、以下の項で説明しよう。

4.3.1 タイム（時間）に関する例

導出尺度の例として、詳細設計工程でのアクティビティに着目する。総アクティビティ数に対する終了したアクティビティ数の比を進捗率として用いる。他にもドキュメントのページ数や、計画した総ドキュメント数に対する作成数の比を進捗率として用いる。

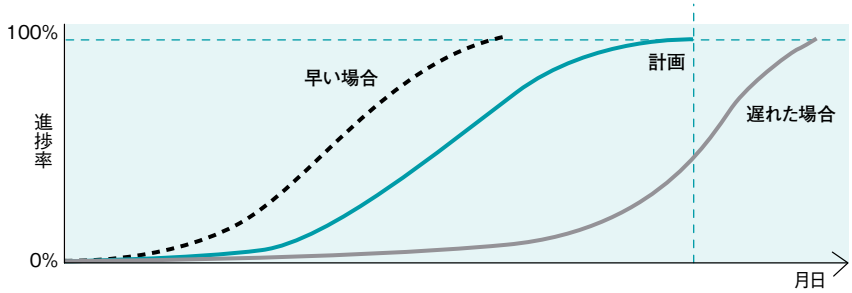
アクティビティの負荷に大きなばらつきがある場合、アクティビティに重み付けを乗じた数を用いることも考えられるが、通常アクティビティは、粒度（負荷）が同程度にそろえられるため重み付けまではしないことが多い。

また、アクティビティの終了数を測

図表 4-1 ●測定分析データー覧表(品質)の例

項目番号	知識エリア(主)	知識エリア(関連)	測定の目的	重点項目	利用者	導出尺度	収集者	導出尺度の見方、分かること
Q1	品質		【保守性・移植性】詳細設計プロセスにおける標準順守率を把握する		PM、PMO	詳細設計プロセスでの標準順守率 標準順守率＝順守している標準項目数／順守すべき標準項目数 ここでの標準とは ●詳細設計書レビュー計画書 ●詳細設計書レビュー基準 ●詳細設計書フォーマット ●詳細設計書記入要領	詳細設計者もしくはレビュー担当者	標準順守率が低い場合、要件の検討が不十分で必要な要件が網羅されていないなど、詳細設計書の品質が低い可能性がある
Q2	品質		【信頼性】詳細設計書レビューが有効に機能しているか把握する	★	PM、PMO	●詳細設計書・レビュー時の指摘事項数 ●ドキュメントページ数当たりの指摘事項数 ＝レビュー時の指摘事項数／ページ数 ●ファンクション・ポイント数当たりの指摘事項数 ＝レビュー時の指摘事項数／ファンクション・ポイント数 レビューでは、標準に順守できていないことも指摘する	PMもしくはレビュー担当者	指摘事項総数、ドキュメント・ページ数当たりの指摘数あるいはファンクション・ポイント数当たりの指摘数が少ない場合、レビューが不十分で詳細設計書の品質が低い可能性がある
Q3	品質		【信頼性】詳細設計書レビュー結果への対処が確実に行われているか把握する	★	PM、PMO	レビューに基づいた ●詳細設計書の指摘事項修正数 ●詳細設計書の指摘事項未修正数 未修正数＝レビュー時の指摘事項数累計－指摘事項修正数累計 ●詳細設計書の指摘事項対応率 対応率＝指摘事項修正数累計／レビュー時の指摘事項数累計	PMもしくはレビュー担当者	修正数が少ない、対応率が低い場合を詳細設計書の品質が低い
Q4	品質		【信頼性】詳細設計書レビューの指摘事項への個別の対応だけでなく、共通的あるいは基本的な対策を実施しているかを確認する		PM、PMO	詳細設計書の ●指摘事項分析の有無 ●指摘事項分析結果による対策実施の有無	PM	詳細設計書の指摘事項分析および対策の実施がされていない場合、詳細設計書の品質が低い可能性がある
Q5	品質		【保守性・移植性】詳細設計プロセスの品質を把握する		PM、PMO	詳細設計工程での手戻り工数の分布(平均、分散、個別、総計)	詳細設計者もしくはレビュー担当者	●手戻り工数が多い場合、レビューの仕方が悪く、詳細設計書の品質が低い可能性がある ●レビュー方法を見直すことなどにより、効率の良いレビューとなり、生産性の向上が期待できる
Q6	品質		【保守性・移植性】詳細設計プロセスの品質を把握する		PM、PMO	詳細設計工程での手戻り回数の分布(平均、分散、個別、総計)	詳細設計者もしくはレビュー担当者	●手戻り工数が多い場合、レビューの仕方が悪く、詳細設計書の品質が低い可能性がある ●レビュー方法を見直すことなどにより、効率の良いレビューとなり、生産性の向上が期待できる

図表4-2●詳細設計の進捗



定するだけでなく、アクティビティの着手数も測定する。終了数、着手数だけの測定では、計画より終了が遅れている数と早く終了した数が相殺されて、進捗の遅れを見える化できないこともある。遅れの見える化のために、計画より着手が遅れている数、終了が遅れている数を測定する必要がある。

詳細設計の進捗率の推移（**図表4-2**参照）を計画と比較してみよう。

まず、詳細設計の進捗が計画より遅れている場合は、基本設計が不十分なことにより詳細設計が進んでいない可能性がある。スケジュール挽回のため、基本設計が不十分なまま詳細設計を無理やり進めると以降のプログラミング、テスト工程において設計の見直しが発生して手戻りが生じ、さらなるスケジュール遅れにつながる。この場合さかのぼって基本設計をやり直す必要がある。基本設計のレビューも、人員と時間をかけ、標準のチェックリストなどを用いてきちんと行う必要がある。

また、クリティカル・パスに遅れがあるときや、重要度の高い機能を実現する部分に遅れがある場合、クラッシング（人員や予算の追加投入）とファスト・トラッキング（並列処理）の対応を迅速かつ適切に取らねばならない。

あるいは、クリティカル・パスに対しては、遅れた場合のコンティンジェンシー・プラン（不測の事態に対する代替案）を事前に決めておき、そのプランを実施することも必要である。

詳細設計の進捗が計画より早い場合は、新規システムの場合や担当者の経験が少ないことがあり、詳細設計における検討度合いが不十分で、品質の低下につながるリスクがある。この時、計画より早く進んでいる理由を正当化できていれば、このリスクは少ない。

4.3.2 品質に関する例

中流工程で、詳細設計を漏れなく確実に行うことにより、以降のプログラミング工程における成果物の品質を向上させるだけでなく、テスト工程からの手戻りを少なくしてコストや工期の目標達成に大きく貢献する。このためには、詳細設計書のレビューを確実に実施することだ。詳細設計工程でのレビュー状況、レビュー結果への対処状況を見ることにより、詳細設計書の品質が、ある程度分かる（**図表4-3**参照）。

まず、詳細設計書のレビューの回数や時間が計画より多い場合は、詳細設計書の品質が元々悪い可能性が考えられる。レビューで十分な品質レベルに上げられていなければ、以降の工程において手戻りが発生し、コスト増加やスケジュール遅れにつながる。

レビューにおける誤りの検出や修正の状況、再レビューの状況（手法、かけた時間、参加した人、検出された誤りと修正、終了時の状況など）を調べ、誤りが出尽くしていると考えられるならば、品質問題が残るリスクは小さい。

一方、レビューのやり方が非効率なことも考えられる。これはプロジェクトのスケジュールやコストに慢性的な悪影響を与えるので、レビュー手順の改善や研修などの対策が必要である。

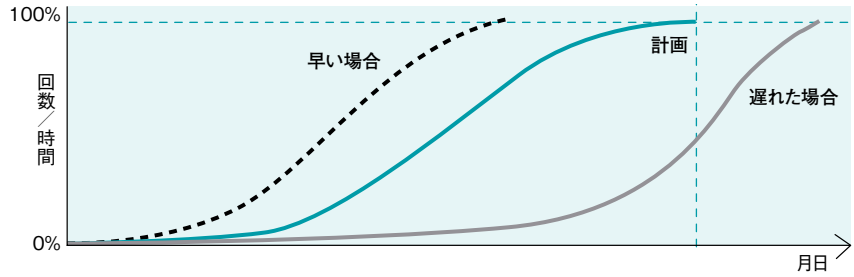
一般に、ドキュメントの品質が悪くてレビューに時間がかかり、レビューが繰り返されている場合には、多少のスケジュール遅延が発生しても、しっかりしたレビュー完了基準に至るまでレビューを繰り返した方が、ドキュメントの品質が向上してプログラミング移行の工程がスムーズに進み、プロジェクト全体としてスケジュール遅れを最小化できる場合が多い。

次に詳細設計書のレビューの回数や時間が計画より少ない場合は、レビューを行った範囲（ドキュメントの数やそのページ数など）が計画より少なければ、設計そのものが遅れていることが考えられる。

レビューが不十分な場合は、詳細設計の品質問題のため、プログラミング以降の工程において手戻りが発生し、コスト増加やスケジュール遅れにつながるリスクがある。この時、レビューの状況（手法、かけた時間、参加した人、検出された誤りと修正、終了時の状況など）を調べ、問題がなければ、品質のリスクは小さいと考えられる。

また、詳細設計書レビュー時の指摘事項数、指摘事項対応状況を分析する。例えば、詳細設計書の指摘事項数が少ない場合、レビューが有効に行われていない可能性が考えられる。あるいは、詳細設計書のレビュー時の指摘

図表4-3●詳細設計書レビューの進捗



に対する修正数が少なかったり、指摘事項対応率が低かったりすると、レビュー結果への対処が確実に行われておらず、詳細設計書の品質が低いことが分かる。指摘事項対応率は、次のようにして計算する。

指摘事項対応率＝指摘事項修正数累計÷指摘事項数累計

4.3.3　人的資源に関する例

詳細設計工程でのプロジェクトの体制の強弱（携わったメンバーのスキル、かけた負荷）を見ることにより、詳細設計書の品質を推定できる。

プロジェクト体制の強弱が、過去の体制のデータをもとに定めた一定の値以下であれば詳細設計が十分にされていない可能性がある。また、マネジメントの体制が弱い場合、管理が十分にされていない可能性がある。具体的な測定項目として下記のものがある。

①詳細設計工程の体制
- 詳細設計経験者数÷詳細設計メンバー数（詳細設計メンバーの経験者率）
- 対象業務分野経験者数÷詳細設計メンバー数（対象業務分野の経験者率）
- 対象プラットフォーム経験者数÷詳細設計メンバー数（対象プラットフォームの経験者率）
- Σひも付け率÷基本設計メンバー数（詳細設計メンバーのプロジェクト間兼務率）

②プロジェクト管理体制
- プロジェクト・マネージャのITスキル標準のレベル
- プロジェクト管理チームメンバーの人数、ITスキル標準のレベル
- 品質管理担当者の専任者数
- 構成管理者（ライブラリアン）の専任者数
- リリース管理者の専任者数
- テスト開発環境管理者の専任者数

第5章

統合的見える化アプローチ

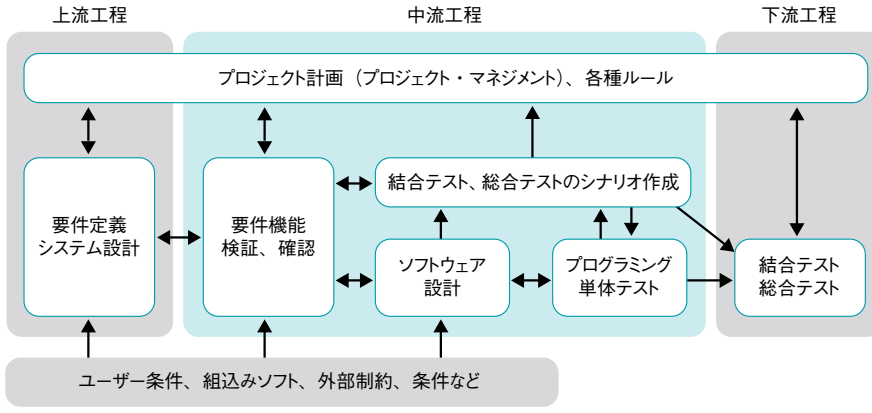
5.1 概要

この章では、中流工程にとっての統合的見える化アプローチを解説する。定量的・定性的アプローチで収集した情報から何を読み取れば、統合的見える化アプローチにつながるのか。また、何を読み取るために定量的・定性的情報を収集しているのかを明確にする。

5.2 中流工程の統合的見える化アプローチ

中流工程は、上流工程で設計した内容を現実に形ある物に作り込んでいくフェーズである。言い換えれば、設計時の不良を現実の製品に組み込むフェーズであり、また、設計にない不良を作り込むフェーズでもある。

図表5-1●中流工程と上・下流工程の関係図



◀▶「見える化」「直せる化」の関連（6.2 中流工程における見える化の全体像とプロジェクト・マネジメント参照）

上流工程の成果物が中流工程へのインプットとなり、中流工程の成果物の不良が下流工程で混入し、最終的に下流工程で問題が判明する状況を**図表5-1**に示す。

中流工程での見える化を怠ったため、プロジェクトに致命的な問題を起こした場合にプロジェクト・マネージャから「詳細設計から製造にかけては一括して外注したので、外注の製造品質が悪かったことが今回の問題の原因である」という言葉が出ることがある。しかし、内製と同様に一括外注だとしても製造状況を把握し、発注者としての責任を果たす必要があることを忘れてはならない。

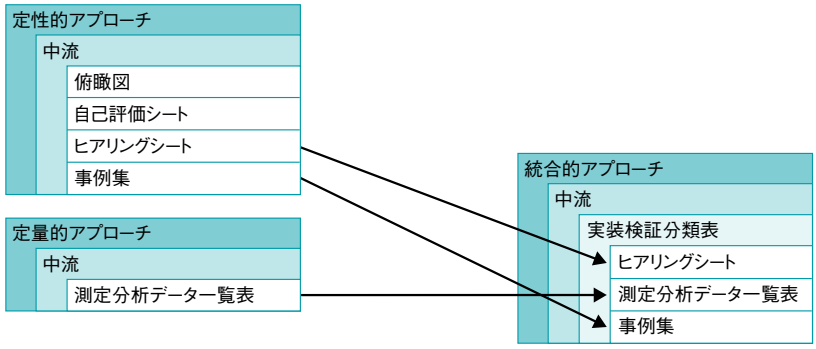
上流工程で作り込んでしまった不良については、他システムとのインターフェース仕様など、製造に入る中流工程では防げない問題も存在するが、現実に発生する中流工程の問題は、プロジェクト・マネージャや各チームのリーダーが製造の状況を見える化し、監視・監督していれば防げる問題ばかりである。

中流工程で見過ごし、下流工程で判明するようなリスクや問題を可能な限り早期に発見し、対策を打つために定性的・定量的アプローチを用いて中流工程を見える化していく必要がある。

上流工程ではリスク分類表、下流工程では症例分類表で統合的アプローチを行ってきた。

中流工程では、「実装検証分類表」を使用して統合的アプローチを行っている。実装検証分類表は付録に記載している。**図表5-2**には、中流工程の統合的アプローチで使用しているツール類

図表5-2●中流工程の統合的アプローチで用いられるツール



を示す。ここで示される通り、中流工程ではヒアリングシート、測定分析データ一覧表、事例集を用いて統合的なアプローチを行っている。

5.3 中流工程における統合的見える化アプローチの観点

中流工程は大きく3つのブロックに分割できる。序盤のソフトウェア設計、中盤のプログラミング、終盤の単体テストである。この3つのブロックでは、それぞれ見るべきポイントが大きく変わるので、中流工程の統合的アプローチは、この3つのブロックそれぞれに対して個別に実施する。

中流工程に対する統合的なアプローチのポイントは、上流工程からの残課題の解決状況および中流工程で新規に発生する課題の確認である。

5.3.1 中流工程の序盤（ソフトウェア設計）

ウォーターフォール・モデルでの開発の場合、中流工程の序盤、ソフトウェア設計を実施する際にはシステム設計は完了しているはずである。しかし、昨今のシステム開発においては、ビジネス・ニーズに応えるために開発期間が短縮されたり、開発依頼元の体質により仕様確定ができないまま、開発に着手せざるを得なかったり、中流工程に至っても上流工程の積み残しが発生する場合が大半である。

上流工程の積み残しがある場合、中流工程に入ることを止められるならば、ウォーターフォール・モデルの開発としては教科書通りである。しかし、実際は開発モデルの型通りにシステムを構築できないため、実情に合わせてシステムを構築せざるを得ない。そのため、本来あるべき姿との差分を明確に管理し、開発依頼元と開発ベンダーの間で積み残しリスクを共有した状態でプロジェクトを進める必要がある。

ここでポイントとなるのは、システムのクリティカル・パスの見極めである。ソフトウェアは顧客が構築中のシステムを肌で感じられないため、作り手と発注側の相互理解を深めることは難しい。そこで、航海のアナロジーを通してITプロジェクトを見て想像してみよう。

航海のアナロジーで見たとき、出航の際「最終目的地には12月1日には到着する必要があるのだが、寄港地については出航後に決めさせてくれ」と言われては、最初の寄港地までの航路も確定できず、これでは到着期限はもちろんのこと、燃料、食料、天候を含めた検討ができない。従って、無事に目

的地にたどり着く保証はなくなってしまう。

また、発注側が航海の目的地を決めず、航海計画を真剣に確認せず、仮決めという名の下に船長に船の方向を決めさせ、船がなんとか港に到着するというところで「こうして見るとイメージと違うので、目的地はあちらの港にしてくれ。ただし、到着までを一括で依頼しているのだし、私は航路をこれでよいと言ってはいなかったのだから、費用は当初の計算通りで頼む」と言われても、実際に船を進めてきた費用は実際に経費としてかかってしまう。「決められないから船を止めてくれ」と言われても、船長にはどうにもできない。一度出航してしまったら、船を海の真ん中で止めても経費はかかり続けるのだ。

ITプロジェクトの現場では、これと類似した状態が頻繁に発生している。プロジェクトが動き始めても仕様を決められない発注側。システムが構築されるまで真剣にプロジェクトに関与せず、発注側の業務担当者がシステムを触ってから初めて「これでは業務がまわらない」と言い始めるケース。教科書的には「目的地が決まる前に船を出す方が悪い」ということになるが、船を持っている（プロジェクトで人員を抱えている）以上、船（体制）を維持するにもコストがかかり、期限という制約も迫ってくるので、現実として船長（プロジェクト・マネージャ）は発注側の気が向くまで港（上流工程の懸案・仕様の全面確定）に足止めをさせるわけにはいかないのである。

そのため、中流工程序盤のプロジェクトでは、上流工程で積み残した事項を中心にリスクの監視を実施していく必要がある。つまり、クリティカル・パスを把握したリスク管理を進めないと、持ち越してはいけない決定事項を持ち越してしまい、システムの中心部に巨大な空白地帯を作り込んでしまうことになる。

5.3.2　中流工程の中盤（プログラミング）

中流工程の中盤、プログラミングを実施する際には、すべての設計が完了しているはずである。しかし、現実には設計がすべて完了するまでプログラミングの開始を待っていられないことが多く、結果として、全体の仕様が固まる前に、仕様が見切れたと考えられた部分からプログラミングを開始することが多くなっている。

この時多発するのが、機能単体とシステム全体の整合性の問題である。例えば、機能単体では成立するが、他の機能の仕様が出来てから見ると、整合

性が取れていない仕様である。システム全体としては問題があるため、後工程で作り直しが発生する。他にも、システム全体で共通化できる部品が、機能ごとに個別に作成されたため、製造時のプログラミング・コストが大きくなるだけでなく、その後のシステム維持で調査や仕様変更などの対応に非常に大きなコストがかかるようになってしまう。

航海のアナロジーを通して見た場合、目前の悪天候を根拠として、船を右に左に回避行動を繰り返しているうちに、本来の目的地から遠くかけ離れた場所にたどり着いてしまうような状態といえる。食糧や燃料の備蓄、期限を踏まえたうえで個々の進路変更を検討していかなければ、本来の目的を見失い、リスクを冒してでも進まねばならないタイミングや、速度を上げて切り抜けるところ、しばし船を足止めしてでも嵐が過ぎるのを待って再出発した方がよいところ、などを見極めて判断することができなくなってしまう。

出航後は船の内部で発生するトラブルや、嵐や潮の流れによる外部要因のため、完全に計画通り物事は進むことはないと考えてよい。そのため、本来の目的と、船の現在地を把握したうえで、計画を修正し、差分を確認しながら、船を本来の目的に向かって進められるよう、常に監視・制御し続ける必要があるのだ。

航海のアナロジーで示した課題への対処の結論をITプロジェクトで実現するため、プロジェクト・マネージャはプログラミングが順調に実施されているかを監視し、見積り通りのスケジュールとコストで製造が実施されているかを監視し、製造を阻害する要因の摘出や計画の修正変更の必要性を絶えずチェックし、プロジェクト全体の制御をしていく必要がある。

5.3.3 中流工程の終盤（ソフトウェア・テスト）

中流工程の終盤になって、単体テストを実施する際には、計画通りのテストが実施されているかどうかを中心に確認していく必要がある。

テストの網羅性の確認を怠れば、単体テストの不良を残してしまうが、結合テスト以降では単体テストと観点が違うので、偶然不良が発覚しない限り、次に不良が顕在化するタイミングは本番稼働後ということになる。

単体テスト工程で摘出された不良が基準を大幅に下回っている場合、品質が良いから不良摘出数が少ないのではなく、テスト項目に網羅性が欠けていたり、作業品質が悪かったりする可能性があるので、不良摘出数が少ない原

因を追究する必要がある。

航海のアナロジーで見た場合、目的地に到達する前に寄港地で積み込む荷物や、出荷品を確認せずに船を出港させることがないようコントロールする状況にある。船を時間通りに出航させたとしても、必要な荷物が正しく積まれていなければ目的地に到着したところで意味のない状態になってしまう。

顧客の着荷検収で積荷の不足が確認された場合、不足した積荷を調査し、積み残しを発生させた寄港地に船を戻し、荷を積み直したうえで再度目的地に向かって船を走らせる必要がある。この大きな痛手は、単純にその港で「積むべき荷をすべて積んだか」「降ろすべき荷をすべて降ろしたか」という、確認をするだけで回避できる類のものである。出航時刻が迫っているという理由でこの確認を怠った場合、後でどれほどの被害が発生するかは想像に難くない。

ITプロジェクトにおいても、航海においても、その場で確認すべきことは、その場で確認しなければ手戻りを発生させる。作業ミスが判明した場合は、原因を追究して再発防止策を打っておかなければならない。これを怠れば次も同じミスが発生するとともに、いつかはチェック漏れで手戻りを引き起こす時がやってくる。

5.4 実装検証分類表の使い方

統合的見える化アプローチで使用する実装検証分類表は、定性的・定量的な情報と、事例を組み合わせて分析することにより、統合的な見地で判断を下せるようになっている。

使用する事例や測定項目などは、自社や自部署、自プロジェクトで保有する関連情報を追加していくことで、より精度が高く現場の状況にあった情報になることが期待される。

上流、下流工程でも分類表を作成して利用方法を提示してきたが、中流工程でも同様に中流工程での観点に合わせた分類表を作成している。中流工程の分類表は、**図表5-3**で示すように、各工程および工程間での見える化のために、横軸に定性的、定量的情報と事例で構成している。

この分類表を用いることにより、各情報を関連付けた統合的な分析が可能となるが、代表的な使用方法を以下に述べる。

5.4.1 定性的情報の裏付けを取るための定量的情報を探る

プロジェクトに対するヒアリングで、問題の可能性が認知された場合、実際にその問題が発生しているのか、

現実に何が起こっているのかを確認する必要がある。また、課題意識や気になるポイントがある場合も同様に、その課題が現実にどのような状況になっているかを確認する必要がある。

こういったときは、ヒアリングシートの項番を分類表の中から摘出し、その情報と同じ行にある定量的な情報（測定項目）を特定する。測定項目を特定し、その情報を収集することにより、懸念事項が現場で実際に発生しているのか否かを判断できる。

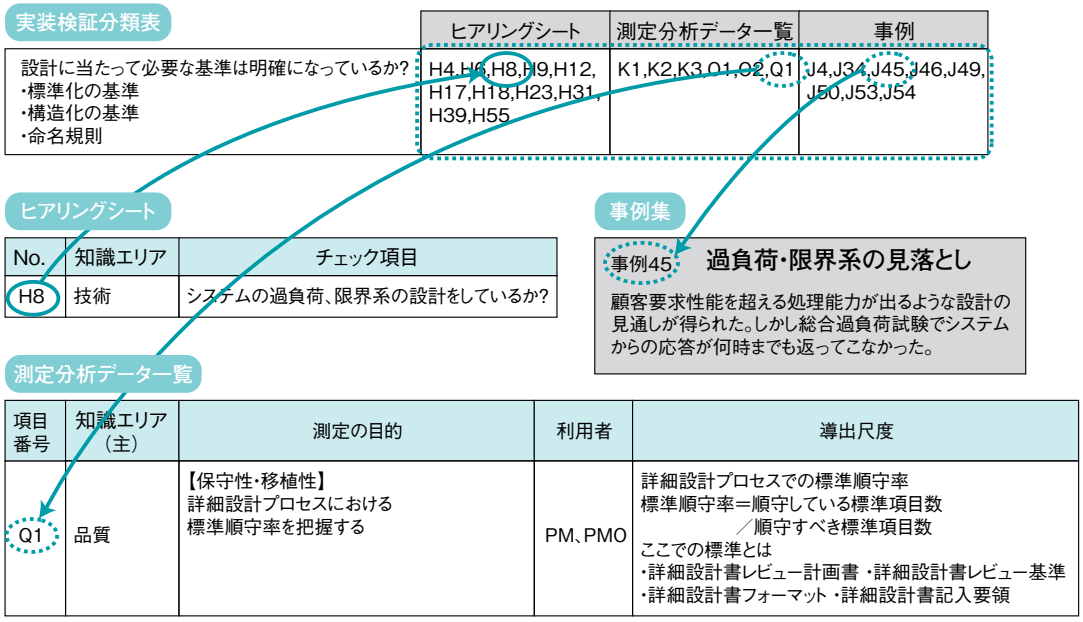
同時に、このプロセスで問題の存在が明確になった場合、どのようなことが発生するかを、同行の事例から割り出せて、その対策として中流工程内で取れるアクションが行のタイトルから判断できる。

図表5-3は、実際にツールを使った検証方法の一例である。これは、次のような段取りで実施する。

①ヒアリング結果を実装検証分類表と突き合わせる

ヒアリングの結果、H8の限界・境界設計に不安があった場合、実装検証

図表5-3●定性的情報の裏付けを取る手順



分類表のヒアリング列でH8に相当する場所を探し、その行をマークする。

②測定分析データ一覧表を参照し、事実を確認する

マークした行の測定分析データを見ると複数の測定項目がヒットしているが、特に設計についてはQ1を用いて確認できるとあるので、その測定結果を基に、事実確認を行う。

③事例集を基に、問題発生時の実例を確認する

現在プロジェクトで発生している問題やリスクが顕在化した際の事例が、事例集から確認できるので、問題点や事例における対策内容を知ることができる。

④実装検証分類表を基に、対策の観点や実施すべきアクションを確認する

5.4.2 定量的情報から問題の根幹を特定する

プロジェクトで測定している項目は大抵の場合、その数値情報だけでは問題の根幹を捉えられず、ポイントを押さえたいくつかのヒアリングによって判断されることが多い。

分類表を参照し、対象の測定項目を使用している個所と同じ行に相当するヒアリング項目を参照することにより、同じ測定項目から問題の本質を捉えるために必要なヒアリング項目をピックアップしていける。また、同時に発生事例や現時点で取り得るアクションも判断できる。

図表5-4に、実際にツールを使った検証方法の一例を示している。これは、次のような段取りで実施する。

①測定分析データで異常値を示した部分を実装検証分類表と突き合わせる

定量的データの測定結果で、測定項目Q1詳細設計の品質に不安があった場合、実装検証分類表の測定分析データ列でQ1に相当する場所を探し、その行をマークする。

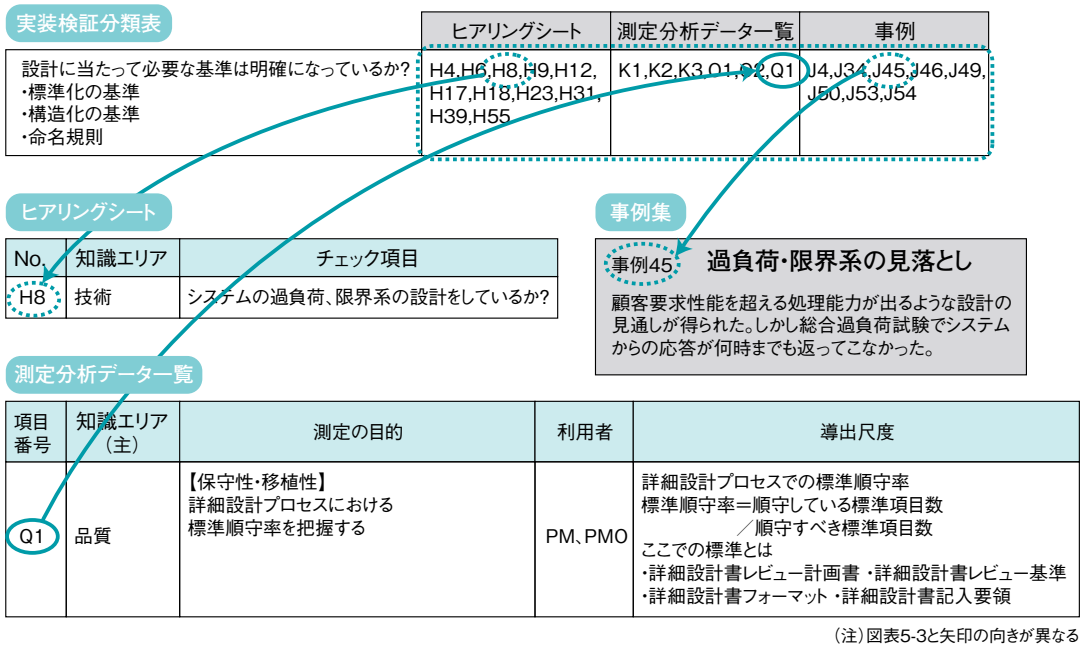
②ヒアリング結果を参照し、事実を確認する

マークした行のヒアリング列を見ると、兆候を確認できる複数の項目が存在している。その内容を基に、ヒアリングを実施し、例えばH8に問題ありというヒアリング結果が確認されれば、プロジェクトのリスクが定性的・定量的に確認されたことになる。

③事例集を基に、問題発生時の実例を確認する

測定分析データで異常値を示した内

図表5-4●定量的情報から問題の根幹を特定する手順



(注)図表5-3と矢印の向きが異なる

容とヒアリングの結果を組み合わせることで、現在プロジェクトで発生している問題の行を絞り込めるので、同じ行を確認するとリスクが顕在化した際の事例を事例集から確認できる。

④実装検証分類表を基に、対策の観点や実施すべきアクションを確認する

本来中流工程で実施すべきことを基にして実装検証分類表が構成されているので、行のタイトルとなる観点部を確認することで、本来実施すべきアクションや対策の観点を絞り込める。

5.4.3 想定される事例から監視すべき定性的、定量的情報を知る

過去の事例を繰り返さないため、類似プロジェクトや現状から類推されるトラブル事例を基に、監視すべき定性的・定量的な情報を特定することにより、必要最小限のコストで類似トラブルの再発を防止できる。

実際にツールを使って監視対象を特定していく手順を示す(**図表5-5**)。

①対象とする事例を事例集から特定する

事例集にある過負荷・限界系の見落としという事例が同プロジェクトで過去に発生していた場合、実装検証分類表の事例集列で事例45の、過負荷・限界系の見落としに相当する場所を探し、その行をマークする。

②ヒアリングシートを参照し、フォローすべきポイントを確認する

マークした行のヒアリング列を見ると複数のヒアリング項目があるが、特にH8については直接チェック内容が示されているので、その内容を基に、ヒアリングを実施することで、問題が顕在化しているか、リスクとして内在しているかを確認できる。

③測定分析データ表を基に、監視すべき測定項目を確認する

マークした行の測定分析データ列を見ると「Q1を用いて状況を確認できる」とあるので、その内容を基に、適宜情報を測定することで、問題発生の

図表5-5●事例を基に監視項目を特定する手順

実装検証分類表

	ヒアリングシート	測定分析データ一覧	事例
設計に当たって必要な基準は明確になっているか? ・標準化の基準 ・構造化の基準 ・命名規則	H4,H6,H8,H9,H12, H17,H18,H23,H31, H39,H55	K1,K2,K3,Q1,Q2,Q1	J4,J34,J45,J46,J49, J50,J53,J54

ヒアリングシート

No.	知識エリア	チェック項目
H8	技術	システムの過負荷、限界系の設計をしているか?

事例集

事例45 **過負荷・限界系の見落とし**

顧客要求性能を超える処理能力が出るような設計の見通しが得られた。しかし総合過負荷試験でシステムからの応答が何時までも返ってこなかった。

測定分析データ一覧

項目番号	知識エリア（主）	測定の目的	利用者	導出尺度
Q1	技術	【保守性・移植性】 詳細設計プロセスにおける 標準順守率を把握する	PM、PMO	詳細設計プロセスでの標準順守率 標準順守率＝順守している標準項目数 ／順守すべき標準項目数 ここでの標準とは ・詳細設計書レビュー計画書 ・詳細設計書レビュー基準 ・詳細設計書フォーマット ・詳細設計書記入要領

（注）図表5-3と図表5-4と矢印の向きが異なる

兆候を継続的に確認していける。

5.5 実装検証分類表の改良

実装検証分類表は、いったん作って利用するだけのものではない。プロジェクトを取り巻く環境は、常に変化しつづけている。現実のプロジェクトを実行することによって得られる情報とノウハウを整理・蓄積し、より精度が高く、より早い時期にリスクが特定できるように実装検証分類表を成熟させ、磨き込んでいくことによって、プロジェクトを成功に導けるようになる。

実装検証分類表は、中流工程で実施すべきアクションと、そのポイントを挙げ、それに対して各種ツールの要素を緩やかにマッピングしたものであり、各組織で個別にカスタマイズしてもよいものである。例えば、利用者の個別の事情に依存する新たなヒアリング項目を追加したとして、そのヒアリング項目を実装検証分類表にマッピングしたり、組織の事情に合わせて既存のマッピングを変えたりしてもよい。組織あるいは業界ごとに、このマッピングを見直すことで、より高い精度が期待できる。

また、各社独自の見える化ツールを作成した場合、この実装検証分類表に新たな列を追加して、その独自ツールの要素番号をマッピングしてみると、ヒアリングシートや測定分析データ、事例集と、独自のツールを連携させることもできるようになる。

第6章

中流工程の見える化によるプロジェクト・マネジメント

6.1 中流工程におけるプロジェクト・マネジメントの特徴

中流工程は、上流工程で決定した要件およびプロジェクト計画を基に、システムを実現する工程である。

上流工程は、プロジェクトの位置付けとして、要件を基にプロジェクトの立上げ、計画立案、顧客との分担の確定などプロジェクト・マネージャが主体となって行動する、いわゆるマネジメントの色彩が強い。また、上流工程は、顧客と密接な関係を保ち、経営層も注目するところであり、発注側である顧客および受注側であるSIベンダー社内からの支援、協力を得やすい。ここで「SIベンダー」とは大手システム・インテグレータ、ソフト開発を行うITメーカーから中小ソフトハウスまで、ソフト開発事業を行うすべての企業を総称したものである。

これに対して中流工程は、活動主体がSIベンダーや国内外協力会社などの技術専門集団に移る。顧客も作業進捗把握、情報提供をSIベンダーに委ねることが多い。そのため、中流工程は、プロジェクト（顧客、SIベンダーともに）にとって見える化がより強く求められる。

このように活動主体が移ったことによる影響がいろいろな形で現れる。順に起こり得る事態を見てみよう。

6.1.1 上流工程での完成度

要件のあいまいさ、未確定など上流工程での完成度が低いことで顕著になる事態がある。例えば、要件をいざ実現しようとすると、ハードウェア／オペレーティング・システム／既存のプログラム／組込みソフトなどの制約、条件が明確になってくる。あるいは、要件の不明確、不整合などが現われることが多い。上流工程での完成度が低いと、このようなことが起こり、仕様の変更・追加や、要件の見直し、システムの実現方法の見直しなどを迫られることになる。

6.1.2　発注者と受注者の意図の違い

発注者と受注者は、もともとプロジェクトに要求する意図が異なっていて、それがすれ違いに発展する場合がある。

つまり、発注者は、できる限りの機能を本プロジェクトに盛り込もうとするが、機能が明確にイメージできないため後工程で不具合が発現することになる。一方で、受注者は、QCDの制約で、可能な限り変更要求を排除して品質確保を優先する。

このようなすれ違いが、仕様の変更・追加や、要件の見直しにつながると、中止や延期といったプロジェクト計画そのものの見直しに発展することもある。

しかし、一般にはプロジェクト計画の見直しにまで踏み込むことは稀だ。その代わり、次のような無理を抱え込むことになる。

第1に、顧客／SIベンダー間、あるいはSIベンダー／協力会社間で発注／受注という暗黙の力学が働き、当初のプロジェクト計画に収めようとする。

第2に、プロジェクト関係者のスケジュール保持の熱意から、中流工程での作業項目を、本来直列で行うべきものを並列に行う計画に変えるなど、当初の計画に収める工夫をする。

第3に、中流工程において、生産性などの客観的、標準的な定量値が存在しないため、たとえ仕様変更や追加により工数が10％増えても、10％の生産性改善といったことで、形式上は当初計画に収めてしまう。

生産性などの定量値の把握は、なかなか理想と現実のギャップを埋められない。これらの定量値は、構築済みプロジェクトの実績値や経験値の累積である。実績としては意味を持つが、その値を標準として扱ってよいか否か、または、標準値とするにはどのような補正が必要かなどの精査は、通常行われない。そのため、数値だけが独走し、納得感のない値になるため、その数値の差が、工数、期間、費用、要員数などにしわ寄せすることがある。

6.1.3　マネジメント手法が異なる

プロジェクト・マネジメントの手法が他の工程と異なっているのも、意外に影響が大きい。中流工程の性格上、マネージャが立ち入る度合いが浅くなり、プロジェクト・マネジメント作業の大部分をメンバーに委ねざるを得ない。そのため、この工程のマネジメントは定量的、表面的、論理的になる傾向がある。

一方、管理上の見える化は、容易な

工程である。世の中に管理手法、管理ツール、事例などが数多く出回っており、各企業の生産性効率化の事例や学会での研究もここを中心にしたものが多い。

しかし、世の中でこれだけプロジェクト管理技術が研究され発展し、かつ広く浸透しているにもかかわらず、状況に対して有効なアクションや予測につながらず、プロジェクトの問題低減、失敗プロジェクトの撲滅に結びつかないのは、次のことが考えられる。

まず考えられるのは、世の中の管理体系は、数字の管理が主体であり状況の表示が大半である。

次に、管理の基本となる基礎数値（生産性、単位工数、所要時間など）が正しいものとしている。例えば、仕様変更要求管理については、要求済み仕様変更と実施済み要求変更の値の管理を行うのみで、その前段階で仕様変更を行わないための仕組みの研究はなされているがあまり実用例がない。

もうひとつ、管理の仕方、手法が時代の変化に則していないことも考えられる。かつての大型コンピュータによるプロジェクトのように、大半を自前で開発するものと、昨今のマルチベンダーによるプロジェクトでは、実践的プロジェクト・マネジメントのやり方は大きく異なる。ところが、最近の管理手法の中には、それを区別していないものが多い。

6.1.4 文化・言語の違いからくるあいまい要件

最近、中流工程を海外に発注することが多いが、文化、言葉の違いから、従来のように国内に発注することと事情が異なる。特に、契約、ドキュメントのレベル、検収条件などは、本来、国内外を問わず必要なプロジェクト・マネジメントとしての共通事項であるため、特に、海外に発注する場合には強く問われるが、実際にはあいまいな事例がみられる。

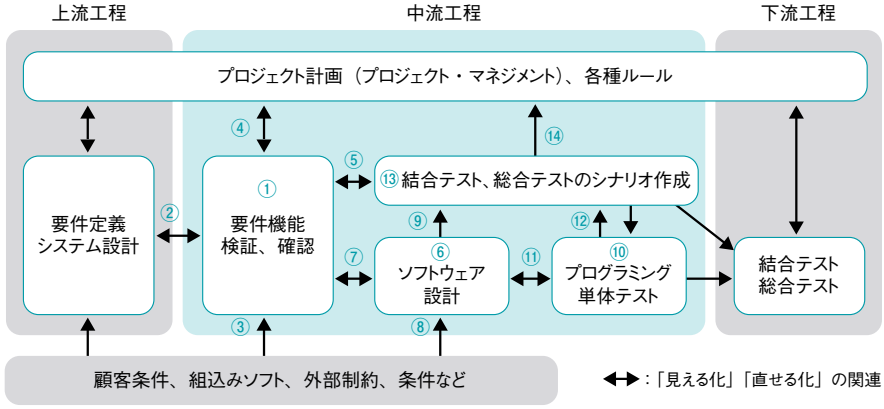
6.2 中流工程における見える化の全体像とプロジェクト・マネジメント

中流工程における見える化の全体像を俯瞰するため、**図表6-1**に中流工程における作業区分と「見える化」「直せる化」の関連図を示す。次節で、これらの関係と流れを説明する。

プロジェクトにとって成功の鍵を握るのは、要件定義、システム設計の出来具合、そして、契約、分担、管理方法など広義のマネジメントの良否、つまり上流工程の完成度である。

上流工程が完全であれば、以降の工

図表6-1●中流工程における作業区分と「見える化」「直せる化」の関連図



程、特に中流工程で大問題が発生することは少ない。仮に問題が発生しても、この工程での問題処理は容易である。別の言い方をすると、中流工程に着手する前に要件定義、システム設計の検証、確認が不可欠であり、この活動を行ったか否かによってプロジェクトの成功、失敗が決まってしまう。

過去の例を見ると、失敗したプロジェクトの大半では、結果的に要件定義、システム設計が不十分のまま中流工程に入るか、入らざるを得ない状況を内包していた。

プロジェクトを成功させるために、中流工程でのプロジェクト・マネジメントの留意点を、**図表6-1**における①～⑭に沿って説明する。

①要件機能の検証と確認

この作業は、ソフトウェア設計に着手するための条件が整っているか、ソフトウェア設計に着手してもよいか、といった上流工程（要件定義、システム設計およびプロジェクト計画）の状況を検証、確認するものである。

昨今、プロジェクト・マネジメント上のあるべき姿と、現実とのギャップは、従来に比べ大きくなっている。例えば、上流工程では、顧客も責任をもって参画すべきなどと言われていても、上流工程における要件定義やシステム設計があいまいなまま中流工程に入っている。プロジェクト・マネジメント上のあるべき姿と現実のギャップを大きくしているのは、ITを取り巻く要求や状況が変化し、多様化してい

るためだ。

①の作業で最も大きな阻害要因となるのは、上流工程で「何を」「どこまで」「どのように」決定するかという一般化した基準ルールがないことである。上流工程で明確にすべき主な項目を列挙すると次のようなものがある。

・RFP、提案の視点（目的、要件、対象範囲、請負条件、制約、完了基準など）
・コストの視点（資源、コスト、コスト・コントロールなど）
・品質の視点（計画、品質管理など）
・技術の視点（プラットフォーム、実績、開発方法論など）
・顧客の視点（オーナーシップ、体制、責任、社会的影響など）
・リスクの視点（リスク・マネジメント、定量、定性分析、リスク・コントロールなど）
・コミュニケーションの視点（会議、情報共有、文化など）
・法令の視点（各種法令など）

要件機能の確認方法として関係者により全体レビューを実施することである。その際の留意点としては、顧客とSIベンダーがともに責任あるとの認識を持つこと。特に、要件確定は顧客の責任であることは忘れてならない大前提である。レビューに当たっては、顧客とSIベンダーおよび協力会社が、対等な立場になるようにして、上下（受発注）の力関係が働かないよう配慮することである。そうでないと、結果的に顧客とSIベンダーの双方が苦労することになる。

②要件定義の再確定

①の作業の中で上流工程に不具合があれば、再度、要件定義、システム設計のやり直し、追加作業を実施する。現実は実施するのが難しいが、今まで問題を起こしたプロジェクトは、この作業を怠ったためだ。仮に、要件定義、システム設計の完成度が低い場合は、②の位置付けで、契約の見直しなど経営的視点での判断も必要である。

③実現検証に基づく要件定義・システム設計の見直し

上流工程で行った要件定義とシステム設計に関して、①で検証・確認、②で必要により見直しを行うのだが、いずれも顧客条件や外部制約などの実現を念頭に置いた検証が求められる。例えば、外部制約なら、契約などの営業的制約、流通ソフトなどの技術、保守などの制約、調達できる要員などの人的制約などを照らし合わせる作業を伴う。この作業に当たっては、それぞれの専門家、担当を含めて確認、検証、または対策（直せる化）を講じなけれ

ばならない。

④要件・設計の見直しに伴うプロジェクト計画の見直し

上流工程で行った要件定義やシステム設計を、①②③を踏まえて見直しが行われた場合、結果としてプロジェクト計画の見直しになることがある。通常、上流工程で一度決めたプロジェクト計画は、よほどのことがない限り変更しない。これが、本質的要因になっている。

⑤結合・総合テストのシナリオ作成

ここでは、要件機能の検証・確認した結果を基に、下流工程で実施する結合テストや総合テストのシナリオを作る作業である。そのための情報提供が肝心だ。プログラムが出来上がってから取り掛かるのではなく、要件機能が定義された段階で、テスト・シナリオ作成に着手することが必要である。結合テスト、総合テストは、要件の確認・実証である。それに、早期にテスト・シナリオを作成しないと、時間的余裕がなくなり、テストの充足度が低くなる。または、中流工程のソフトウェア設計、プログラミング、単体テストの状態からのテスト・シナリオ作成となり、本来の結合テスト、総合テストの意味がなくなる。

⑥ソフトウェア設計

中流工程の着手前の検証で問題がないと判明したら、いよいよソフトウェア設計を実施する。その際、要件定義、システム設計に仮に過不足があれば、冷静に分析し、上位者に速やかにエスカレーションする。プロジェクト・マネジメント上は当然、標準化、構造化、ネーミング・ルールなどが行われていなければならないからだ。

⑦ソフトウェア設計を踏まえた要件定義の再検証と再確認

⑥のソフトウェア設計で不具合が発生したら必ず、再度要件機能の検証・確認を行うことだ。ソフトウェア設計は分担して行うことが多く、一部に問題があれば、通常限定した個所にのみ問題が発生することは少なく、全般に問題が潜んでいることが多い。そのため、個々に改善するのではなく、全体的にフォローが必要である。そして、改めて⑨でテスト・シナリオを見直す必要がある。これを怠ると、テストに漏れが発生してしまう。

⑧ソフトウェア設計に伴う実現性検証

⑥のソフトウェア設計を行うに当たり、オペレーティング・システム、組み込みソフト、流通ソフトなどの制約、条件の検証・確認を行う。その際、

以下の作業が大変重要になる。

・機能、性能、インターフェースなどを必ずレビューして、Q＆Aの中で確証を得ること。
・異常時、障害時、不整合時のインターフェースと動作などを必ずレビューして、Q＆Aを実施し、確証を得たうえでドキュメント化しておくこと。必要に応じて、テスト・シナリオ、運用ドキュメント、障害時運用などに反映させること。
・異なるバージョンとの互換性、システム制限値、システム変更時の影響範囲などをレビューして、Q＆Aを実施し確証を得るとともに、保守・維持ドキュメントに反映させておくこと。

ひとたびシステム構築が始まると、システム要件の変更を考えることは稀であるが、⑧でまとめられたドキュメントは、後日システムに追加、変更が発生した場合、重要な情報となる。

⑨結合テスト、総合テストのシナリオ準備

ソフトウェア設計の結果を基に、さらに結合テスト、総合テストのシナリオを作成する情報を提供する。早い段階からテスト・シナリオを検討、作成しないと、テストの抜け、片寄りが発生する。また、テスト環境、テスト構成、テスト工数、検証方法などの準備が不完全となりシステム品質の低下につながる。

⑩プログラミングおよび単体テストの実施

この作業は、ソフトウェア設計を機械的にプログラミングし、単体テストを行うことである。もし、これが不十分なら、迅速にソフトウェア設計の見直しまでさかのぼる必要がある（⑪）。不十分なソフトウェア設計をプログラミングで解決しようとすると、標準化、プログラム構造化などが崩れ、将来の保守性まで影響する。そのためには、次のことに留意する。

・プログラミングと単体テストで顕在化した問題を軽視せず、必ず見える化を図る。一般的に、プログラミングと単体テストは、あまりにも専門的なため、担当者に任せ切りになったり、軽視したりする風潮がある。
・プログラミングの成果は、システム完成度の源泉であり、コードインスペクション（検証）を実施し、問題点を撲滅するとともに、分かり易いプログラムにすべきである（チェックの容易性）。
・プログラミングはシンプルなものにすること。かつては、職人技のように難解なプログラミングを目にする

こともあったが、現在はハードウェアの性能がよく、価格も下り、無理をして使用メモリー容量を小さくする必要は全くない。むしろ、機能追加変更の容易さ、事象の切り分けの容易さなど、保守運用性を重視すべきである。

⑪プログラミングと単体テスト結果によるソフトウェア設計の立ち返り

⑩のプログラミングと単体テストで不都合が発生したら、ソフトウェア設計に立ち返って検証することになる。特に注意を払わなければならないのは、プログラミングとドキュメントの整合性である。原則論を言えば、ソフトウェア設計のドキュメントに沿って、プログラミングするものであるが、実質は必ずしもソフトウェア設計のドキュメントとプログラミングは一致しないことが多い。特に、単体テストの結果によりプログラムを変更、修正した場合、ドキュメントの改訂まで手が回らない。しかし、システムを検証、維持するためには重要なことである。

⑫単体テストの結果から結合テストと総合テストのシナリオへ反映

⑩で実施した単体テストの結果は、⑨で準備している結合テストと総合テストのシナリオ作成に反映して、追加変更をしなければならない。このような情報提供が、テスト・シナリオ作成にとって重要である。

⑬テスト・シナリオの検討と作成

①⑨⑩の結果を踏まえて、結合テストと総合テストのシナリオを仕上げていく。ここで留意しなければならないことは、単体テスト、結合テスト、総合テストのそれぞれにおいて、目的／内容／方法／期待／確認者が異なるということである。ややもすると、スケジュール遅れをカバーするためそれらの区別が無くなったり、総合テストのシナリオ作成や検証確認をSIベンダーに任せ切りになったりすることが多い。特に、顧客は総合テストのシナリオ作成に当たって、業務サービスで利用することを想定してテスト・シナリオ作成に参画すべきである。

⑭プロジェクト計画の見直し

中流工程の作業において、①の結果を踏まえて、④のプロジェクト計画を更新した後に、⑥ソフトウェア設計や⑩プログラミングと単体テストを実施すると、後からプロジェクト計画の見直しが必要になることがある。例えば、「作業進捗から見たスケジュールの見直し」「体制、工数、要員、設備、費用などの見直し」「報告内容、報告方

法などのプロジェクト運営上のルールの見直し」が必要となったら、プロジェクト計画を再度更新し、全体計画の整合性を取るとともに、顧客を含めたプロジェクト・メンバーでの情報共有を図る。通常、⑭においてプロジェクト計画を見直すことは希で、進捗上問題が発生していても、管理されないまま体力勝負、または物量作戦により運営されるケースが多い。

おわりに

本書では、中流工程を、上流工程では見えなかったような主要問題（ドミナント・アイテム）を下流工程に持ち越さないための「品質作り込みの砦」としての工程と捉え、「砦」としての役割を果たすために、問題を速やかに見える化する、あるいは問題是正結果も見える化するための具体的・体系的な手法やプロジェクト・マネジメント論について述べた。

また、本書では、「中流工程のインプットとなるシステム要件、ソフトウェア要件の完成度と仕様変更要望への対応」「分業化するプログラミング作業のプロジェクト管理の要点」と「ソフトウェアやシステムの開発における現場、現実、現物による管理」などが、「本来はそうすべきかもしれないが、現実的には様々な事情から、理想通りにはできない」というプロジェクト管理の現状を考えて、現場で実践的に使える方法を紹介した。

多くのプロジェクトの現場で、未然にプロジェクトの課題を見える化し、トラブルが防止されるとともに、ソフトウェアへの不良の作り込みを抑止するために、本書が活用されることを期待している。

付録 1. 自己評価シート

No.	知識エリア	チェック項目	評価基準	
S1	統合	プロジェクト計画書が作成され、レビューされているか さらに、これに基づき実行されているか	●プロジェクト計画書は、プロジェクト開始から規定期限以内に作成され、レビューされていること ●プロジェクト計画書は、ひな型になるような過去事例や社内標準などをもとに作成していること	
S2	統合	プロジェクトの全体像を俯瞰できているか キーマンや重要機能が規定されているか	●今回の開発対象となるシステムを含めた、顧客側の関連システムを表すドキュメント（システム構成俯瞰図など）を作成していること ●顧客や開発側の関係会社（協力会社など）に関係する体制図があること ●ドミナント・アイテム（プロジェクトの成否を左右する支配的要因）を把握できていること	
S3	統合	納入物件および必要な作業成果物が明確となっているか 契約書との比較は済んでいるか	顧客との間でどういう記述方法で、どういうドキュメントを納品するかが明確になっていること	
S4	統合	前段の段取り作業を含めてすべての作業項目がスケジュール化されているか	●WBSが書き出されており、その内容について十分検討されていること ●すべてのWBSがスケジュールに落とされていること	
S5	統合	チェックイン、チェックアウト、作業責任者などの構成管理のルールが明確になっているか	●計画書、設計書なども含めた、成果物の構成管理の手順やルールが文書化され、メンバーに周知徹底されていること ●プログラムのソースコードや仕様書、テストデータ管理、システムの設定ファイルなどの構成管理方法についての計画がされていること ●計画書、設計書なども含めた、成果物の構成管理の手順やルールが文書化され、メンバーに周知徹底されていること ●プログラムのソースコードや仕様書、テストデータ管理、システムの設定ファイルなどの構成管理方法についての計画がされていること	
S6	統合	全体テスト方針と役割分担の検討は十分か	全体テスト方針（何をどのテストで誰が確認するか）とテストの役割分担（誰がどの範囲をカバーするか）を記述したドキュメントを作成し、関係者とレビューしていること	

	マネジメントにおけるヒント	対策案
	●計画の無い所には、マネジメント無し ●レビューはステークホルダー間の合意形成の場である。プロジェクト計画書のレビューを通じた合意は最初の、最も重要な合意事項である ●工期、品質（重点機能）などの目標値をどの範疇にいれるのかなど、ターゲットを明確化する	●プロジェクト計画書が作成されていない場合は、早急に作成する。名ばかりのプロジェクト計画書にならないよう、十分な記述内容を心がける必要がある ●レビューを行った後に、スケジュール、納品物、責任分担については、レビューを通じてステークホルダーの同意を得る
	●プロジェクトの全体像を俯瞰できないと、プロジェクト計画がきちんとできていない可能性がある ●完全なものが作成できなくても、トライすることが重要である。トライの積み重ねとその後のプロジェクト・マネジメント経験の反省を通じて、俯瞰図作成レベルは着実に向上する	●俯瞰できていない場合は、まずは顧客側キーパーソンに十分にヒアリングしたり、マススケジュールを見直したり、関係者（例えば接続先となる他システムの担当者）と打ち合わせを持つなどして、意見交換をすることである ●各担当者へのヒアリングなどの作業を通じて、どの部分がキモになるかを把握する（ドミナント・アイテムの把握）
	●顧客との間で成果物の質と量が明確でないとその後の作業の割り振りや規模の見積もりなどで大きな判断ミスをする可能性がある ●事前に成果物サンプルを提示することで、納入物件に関する双方の共通認識を確立することが望ましい ●各種計画書や仕様書、運用テストの項目一覧など、具体的な成果物のサンプルを示すことで、どのような作業が必要になってくるかを事前に検討したり、要員のアサインを考えたりしやすい ●各成果物のボリューム観の顧客との認識あわせ	ドキュメントによる成果物の場合は、目次、類似する成果物サンプルを用いて記述項目と記述レベルについて、顧客と合意し、議事録として残す
	●必要な作業項目を洗い出すために、例えば各社で標準化しているWBSのひな型を利用したり、過去の他のプロジェクトを参考にするなどの工夫をすること ●システム構築のための設計書作成やプログラム開発作業だけではなく、準備作業、顧客との調整事項など、想定される作業をなるべく網羅的に洗い出すこと ●WBSをもとにしてスケジュール中の各作業の前後関係の整合が取れていること	●標準WBSに基づいてテーラリングする。標準WBSがない場合であっても、過去の成功プロジェクトなどを参考にして、WBSを作成する。「このWBSで必要な作業がすべて洗い出されているか」という視点を常に持つこと ●各作業の前後関係の整合が取れていない場合、前作業の終了条件、後作業の開始条件を明確にして、各作業の前後関係が整合するように修正する ●ドミナント・アイテム関連作業のスケジュールは、計画段階から重点的にマネジメントする。例えば、ミッション・クリティカル性の高い機能要件などもドミナント・アイテムである
	●各種成果物の構成管理は、プロジェクトの初期段階でルールを決めて運用することが大切である。あとで整理しようと思ってもなかなか手を付けることが難しい ●構成管理をきちんと行うことは、成果物および活動の品質を上げることに寄与する ●構成管理ツールを新規に導入する場合には、環境設定や試験運用などの計画を明確にしておく必要がある。また、運用ルールを明確にし、入力するデータの意味などを詳細に定義しておく必要がある	●構成管理ルールがある場合には、ルールが周知徹底されているか確認する ●構成管理ルールがない場合には、以下を考慮してルールを作成する ①構成管理のためのツールの導入（費用の確保、環境の整備、運用手順の明確化が必要） ②作業工程によって管理ルールが変わる（使用するツール、管理対象物、管理頻度） ●構成管理ルールが徹底されているかを確認する手段を検討する（ヘルプデスク、進捗会議でのチェックなど）
	●詳細なテスト方針はともかくとして、どのようなテストを行うかを決めておくことが重要である ●方針と役割分担を明確に決めておかないと、場当たり的なテストになり混乱するおそれがある ●上流工程でテスト工程をある程度想定しておくことで、顧客側の体制や作業場所などの準備作業、テストデータの準備など、協力も得られやすくなる。特に、全国展開するようなシステムの場合は、各事業所ごとにテストをするなど、集中的なテストができないなどの理由で、テストそのものがクリティカルパスになることもある	●単体テスト、結合テスト、運用テストのそれぞれのスケジュールと品質目標が未計画の場合は、過去事例などをもとに整備し、計画する ●テスト体制も明確にする。特に、結合テスト以降は優秀な人材（各チームのキーパーソンや優秀なエンジニア）を投入し、遅れない体制を計画しておくことが重要である

No.	知識エリア	チェック項目	評価基準	
S7	統合	システムにとって致命的な障害の要因となる機能やモジュールを提示し、管理しているか	プロジェクトの重要機能の優先順位や留意すべき項目が挙げられていること	
S8	スコープ	ベンチマーキングするなどして、顧客の要件とシステムの機能、性能に差異がないことを確認しているか	顧客の要件と機能の対応表またはそれに相当するドキュメントがあること	
S9	スコープ	仕様として設定された要求の実現可能性に問題がないことをレビューし確認したか	●実現可能性を新技術面、性能面から検証していること ●検証結果について、有識者によるレビューがされていること	
S10	スコープ	要件定義と仕様変更要望との関連付けが明確になっているか	顧客側のニーズ（要求）と要件定義書の項目の関係性について、顧客側、ベンダー側双方のキーパーソンに確認を取ること	
S11	スコープ	仕様変更は一覧などにより管理され、変更履歴を残すことになっているか	仕様変更管理表などで管理することが明文化されていること	
S12	スコープ	他のステークホルダー（顧客、関係会社など）からの提供物のスケジュールが守られているか	●提供物のスケジュール状況を継続的にウォッチし、守られなかった場合の影響度に応じて、事前に対策を立てていること ●遅れることも想定し、リスク項目として管理していること	
S13	タイム	進捗状況を定期的に確認できるようになっているか	●定期的に進捗管理が行われ、プロジェクト・マネージャやプロジェクト責任者、顧客が状況を把握できる仕組みが設定されていること ●進捗管理基準（着手基準、進捗基準、完了基準）を設定していること	

	マネジメントにおけるヒント	対策案
	ミッション・クリティカル性の高い機能要件は失敗してはならない機能なので、計画・設計時から入念に管理され、実行され、監視される必要がある	
	●要件定義書と設計仕様書との対応は、仕様管理だけでなく、テスト計画にも利用できる ●顧客における常識、業界での常識などは、明文化されていない可能性があり、注意が必要である	●要件定義書がない場合には、早急に作成する（最低限、要件項目を明確化する） ●要件定義書がある場合は、現状と要件定義書を比較して特に乖離の大きい部分があればその部分の状況報告をし、顧客側と認識をあわせておく必要がある ●要件定義書には記述されていない要件がある場合、早めに洗い出しをし、クリティカルな要件になるようであれば実装を機能面・費用面から検討する ●要件定義書の作成段階で頻繁に修正があった機能については、最終的な要件について基本設計書や仕様書との擦り合わせを行うなど、顧客との認識を十分に合わせておく
	●計画または設計時に実現可能性そのものについて検討を行っているかは重要である。さらに、内容について有識者などの第三者によるレビューを行うべきである ●実現可能と判断するには、例えば、コスト、機能、納期に対してリスクを認識し、その対応策が見えている必要がある ●実現可能性について疑問視される部分については、顧客側によく確認すること。思い違いにより、ベンダー側が仕様を重く捉えている場合もある	●実現可能性を検討していない場合は早い段階で検討する ●実現可能性が見えていないものがあるかもしれないが、それらも含めて全体を俯瞰できるよう、要求された項目の実現可能性の一覧を作成する ●実現可能性が低いと思われる機能の場合は、代替機能での実現可能性や機能削除などについて顧客とともに検討する ●ドキュメント間の整合性を確認するための色塗りコンペア・チェックを実施する
	●要件定義と仕様変更の対応付けを管理することによって、顧客要件との合致性を確認する ●システムは要件定義書をもとにした各種設計書によって作られている。従って、この要件定義内容と顧客からの仕様変更要望との対応付け確認は、システムでの実装内容を把握しているキーパーソンも巻き込んで確認する必要がある ●要件定義書と、顧客側からの仕様変更の要求に論理的な矛盾があると、修正したあとのプログラムではシステムトラブルを起こすことにつながる可能性がある	要件定義と仕様変更との関連付けを明確にするために、要件定義と仕様を対応表にして、相互に影響する要件定義変更や仕様変更を管理する
	以下の項目が明確になっているかを確認する ●仕様変更管理表の保管場所 ●仕様変更管理表の記入者・保管責任者 ●仕様書の変更履歴と仕様変更管理表の対応確認方法	要件定義との関連付けが可能な仕様変更管理表（一覧）を作成する
	他のステークホルダーの役割と提出物のスケジュールを別途マネジメントすることにより、役割分担があいまいになっていることを原因とするプロジェクトの遅延を防止できる	●他のステークホルダーからの提供物がないことによって影響を受けない作業から着手するが、作業の進捗状況とステークホルダーとの調整を並行して行い、作業が滞留しないようにする ●ステークホルダーへ情報提供の督促を行うときは、先方の担当者だけでなく、先方の上位上司にも一緒に依頼することが効果的である
	●進捗の状況を把握するための項目を明確にし、協力会社を含めた進捗報告確認のための会議体を設定する ●進捗の状況は、プロジェクト・マネージャだけではなく、プロジェクト責任者、顧客にも報告する必要がある ●言葉の統一をはじめ、進捗管理基準を設定する ●進捗状況に加えて、遅延理由、遅延対策まで検討する	●進捗会議を定例的に設定する ●進捗会議で報告すべき管理項目を明確にする ●管理項目は作業工程によって変わる （例）管理項目例 ①計画の変更数 ②確定した要件定義の個数 ③実施レビュー数と種類 ④日程の予実績の差異 ⑤リスク数（定義数、監視数、軽減数） ⑥課題管理表の解決した課題数、未解決課題数

No.	知識エリア	チェック項目	評価基準	
S14	タイム	要員の手配（量的）計画はできているか	●工程の変わり目など、体制の増員が必要なタイミングに間に合うように要員手配が計画されていること ●要員の手配計画について、関係各所にあらかじめ説明をしておくこと ●コア部分とそうでない部分の切り分けができていること ●外注先の特性を把握していること	
S15	タイム	マスタースケジュールが作成され、ステークホルダー間で合意されているか	●マスタースケジュールが作成されていること ●マスタースケジュールについて、ステークホルダー間（顧客、関係会社、プロジェクト責任者など）の合意が得られていること ●変更ルールに関しても合意していること	
S16	タイム	スケジュールにおけるクリティカルパス（そのパス上の作業が少しでも遅れると、プロジェクト全体のスケジュールが遅れてしまうパス）は明確か	プロジェクトの根幹にかかわる機能や成果物に関する計画はどのパスであるかを認識し、管理していること	
S17	タイム	前工程の作業が完了し、プロジェクト内（プロジェクト・マネージャ）の承認を得ているか	●前工程に積み残し課題がある場合は、解決のメドをつけた上で次工程に引き継いでいること ●見切り基準をもうけていること	
S18	タイム	テスト計画について、実施責任者（プロジェクト・マネージャ、プロジェクト・リーダーなど）を含めたレビューにて内容を確認しているか	●計画の妥当性を検討していること ●テスト計画の俯瞰ができていること	
S19	コスト	予実績管理は実行されているか	●収支管理計画を事前に定義していること（できる限り定量的に） ●週次 / 月次の実績把握の仕組みを確立していること	

	マネジメントにおけるヒント	対策案
	●要員を出してもらうことについては、事前に関係会社に確認をしておくべきである。要員を出してもらうつもりだった関係会社から、突然協力できない旨の連絡がくる場合も想定しておくべきである ●担当者個人ごとに作業開始時期、終了時期を明確にして作業をアサインするようにする ●前工程における仕様変更などで必要な人員の増減が必要になった場合は、工程の終了を待たずに速やかに手配する必要がある ●開発技術者のスキル（自社における過去の実績など）を把握する必要がある	●要員遷移俯瞰図の作成を通じて、山積みではなく要員遷移分析を行い、WBSごとに、必要なスキルを持った要員の過不足を明らかにする ●前工程の終了前に、プロジェクト・マネージャがどれだけ次工程の要員手配に注力するかが重要である ●要員の手配は、早めに、過大気味に行う ●要員不足に陥ったときには、急激な破綻を起こさず、軟着陸できるようにするために、スケジュールの延長、または作業量の低減の余地を検討する
	●マスタースケジュールの各工程での作業内容について、十分に検討していることが大切である。特に、納期を基準にして、作業工程の比率だけで、後ろからスケジュールを引いている場合、それぞれの工程がその与えられた期間で完遂可能かを十分に検証しなければならない ●マスタースケジュールの中で、既にわかっているマイルストーンについて記載しておくこと ●マスタースケジュールの各工程について、事前準備の内容と時間なども考慮してスケジュールに記述しておくことが望ましい	●納期、総合テストの開始時期、他システムとの接続がある場合は接続テストの実施日など、明確なマイルストーンに基づいてマスタースケジュールを作成する ●マスタースケジュールの妥当性については顧客との合意だけではなく、プロジェクトの担当者、開発委託先のメンバー、有識者などに確認してもらうこと
	●「粛々とやっていくだけ」と考えてプロジェクトを進めていると、問題に気づかないことがある。計画している作業タスクの優先順位を検討して常に配慮することで、問題が発生したときにどの作業を優先させるかを冷静に判断できる ●クリティカルパスを固定的なものと捉えないこと。状況の変化によってクリティカルになる可能性のあるパスも考慮すること ●複数のアウトプットが集まるタスク、複数のタスクのインプットにつながっているタスクはクリティカルパスの候補の1つになる ●機能間の相互依存となっている度合いが大きいとクリティカルパスは非常に多く発生する	●キーパーソンを集めてクリティカルパスの抽出作業を行う。そのうえで、クリティカルパスに関連する作業は特別体制を敷く ●作業工数の見積りが確定できず、ある程度進まないと見通しが立たない場合には、作業が少し進んだ段階で再度工数を見積り、クリティカルパスを明確にする
	●上流工程の品質の悪さは下流工程でより大きな悪影響を及ぼすことがある ●しかしながら、すべてを完了させてから次の工程にはいることは困難な場合も想定されるため、次善の策を設ける	●すべてを完了させてから、次の工程にはいることは通常考えにくいため、見切り基準を準備する
	テストによっては、顧客の確認が必要な場合もあるため、顧客の確認も必要である	顧客を交えてテスト計画のレビューを実施し、確認する
	●期間（工程の開始／終了など）、要員の山積み、およびWBSについて予実績管理を行う ●予算を順調に消化していても、仕事が進捗しているわけではないということを分かったうえで、マネジメントすること	●予実績情報を収集・整理して、今後の計画の見直しを実施する ●すべての作業の予実績情報の収集が困難な場合は、クリティカルパスについての予実績だけでも集めて評価する ●過去に多数の類似プロジェクトの実績がある場合は、上流工程でのコストの振れや工程の遅れは抑制しやすい。ただし、「過去のプロジェクトと類似だ」という判断そのものが誤っている場合があり、その場合は後工程で工数が数倍に膨れあがることもある。類似プロジェクトかどうかは慎重に判断しなければならない

No.	知識エリア	チェック項目	評価基準	
S20	品質	品質計画があるか	品質計画(右の欄参照)があり、レビューされていること	
S21	品質	前工程の作業が完了し、その品質を確認した上で、プロジェクト内の承認を得ているか	●前工程の結果について品質を確かめていること ●前工程に積み残された課題がある場合は、解決のメドをつけた上で次工程に引き継いでいること ●条件付きの合格の場合や改善点を指摘された場合は、課題管理の対象としていること	
S22	品質	日々のテスト項目消化件数、不良摘出数、不良対策件数を入力する仕掛けが整備されており、バグ曲線による管理が行われているか	●指標となる項目が明確でデータ収集方法が明確になっていると ●テストの進捗と結果を管理する仕組みができていること	
S23	人的資源	必要なスキルを持った人材はそろっているか	必要な業務・知識領域と要員のスキルの比較を行い、不足している業務・知識領域について、コンサルタントや専門家のアサインを考慮していること	
S24	コミュニケーション	プロジェクト計画書は、全員が見ることができるようになっており、全プロジェクト・メンバーへ展開しているか	●プロジェクト計画書の所在を全員に連絡していること ●プロジェクト・メンバーにプロジェクト計画書の内容を確認し、理解してもらうよう配慮していること	
S25	コミュニケーション	顧客、顧客側キーパーソンなどの要求仕様を十分に考慮し、必要に応じて要求仕様のレビューが行われているか	●顧客側の要求仕様に対する理解に問題がないかをチェックしていること ●顧客からの要求仕様を整理していること ●要求仕様が有識者やキーパーソンによってレビューされていること	

	マネジメントにおけるヒント	対策案
	●例えば、次のような点について、計画とレビューがされていることが望ましい ・レビューの種類 ・レビューの日程 ・レビュー不良率(件／ページなど) ・下流工程における品質基準（不良摘出密度、レビューにおける不良摘出目標(○件／頁）などの指標） ●そのプロジェクトで特に重要な品質は何かを検討しておくこと（信頼性なのか性能なのかなど） ●起こってはいけない障害は何かを考えて、その障害を起こさせないためにはどのような品質を向上させる必要があるかを検討しておくこと ●品質は様々な要因が総合的にからむので、レビューを通じて、問題点を早期に把握して是正	●品質計画が作成されていない場合は、すぐに作成する ●品質レビューの検討結果に基づき、対応策を策定して、実施する
	●上流工程の品質の悪さによって、下流工程でより大きな悪影響を及ぼすことがある ●前工程での積み残し課題などがあればそれらの位置付けを明確にし、どのような対応を行うかを計画に入れる ●エビデンスを取得する機能とそうでない機能の切り分け	品質に問題がある可能性がある。品質状況を確認して、問題がある場合は品質向上策を実施する
	● 以下の帳票類があること(現物で確認) ・記入済みのバグ票 ・現時点までのバグ曲線(あることが望ましい) ・テスト実施要領 ・テスト実施・結果の管理票 ●集計の仕組みと責任者が明確になっていること（専用ツールを使っている/Excelなどのツールで集計/人手集計） ●毎朝会議を実施していること(結合テスト以降)	
	●要員のアサインには時間がかかることが常であり、場合によってはアサインできない場合もある。その場合、トレーニングなどによって要員を育てる時間が必要になるので、事前の要員計画の中で必要なスキルについても十分計画しておくことが大切である ●システム基盤に問題が発生した場合はスペシャリストのアサインが有効である ●特殊な業務に関する知識が不足している場合は、外部の専門会社などに依頼することを考慮しておく必要がある ●パッケージ利用の場合、フィット＆ギャップ分析ができる要員や、該当アプリケーションの経験者をアサイン	●最適なスペシャリストを投入する計画を立てる ●必要な人材確保が遅れている場合、最初のうちはそれほど高度な業務知識や専門知識がなくても問題ない作業を行い、作業が進む中で必要な人材を確保する
	プロジェクトの概要や目的、スケジュール、スコープ、体制図など、プロジェクト計画書に記述されている内容をプロジェクト・メンバーに周知することによって、プロジェクトに参画してもらううえでの心構えを持ってもらう必要がある	●プロジェクト計画書を最新に保つようにし、重要な項目については必ず目を通すように徹底する ●新しくプロジェクトに参加したメンバーに対しても説明することを忘れないようにする ●重要な項目については、プロジェクト・メンバーに何度も徹底する
	●顧客からの要求仕様に対して、レビュー会などの打ち合わせの場を設けるなど、直接的な会話で内容を確認するべきである ●顧客からの要求仕様は、文字にしがたい背景を含んでいることが多く、背景も含めて要求仕様をヒアリングするべきである ●顧客からの文書による依頼内容だけで要求仕様を理解した気になっていると、要求が十分に抽出されていないおそれがある	●要求内容に関して、顧客と直接打ち合わせをする ●そのためには、必要に応じて顧客から直接、説明を受けられる関係構築が重要である ●打ち合わせに出席しているメンバーに、責任者がいることを確認すること

No.	知識エリア	チェック項目	評価基準	
S26	コミュニケーション	顧客との「仕様」「価格」「納期」に関するやりとりは文書で行っているか	「仕様」「価格」「納期」に関する経緯と契約が文書化され、保管されていること	
S27	コミュニケーション	顧客、開発チーム間、協力会社との会議体が決められ実行されているか	ステークホルダーと協議し、必要な会議体が設定され、実行されていること	
S28	コミュニケーション	顧客への報告は随時行っているか	●顧客との会議体を決めて、実際に行われていること ●社内における会議体など全体の会議体のスケジュールを調整すること	
S29	リスク	リスク分析により、リスク項目を明確にし、対応策まで検討しているか	●リスク管理表などを作成して管理すること ●個々のリスクについて、リスクの対応策とその実行タイミングが明記されていること	
S30	調達	協力会社の作業と成果物を定期的に確認する計画があるか	●協力会社の作業と成果物を確認する計画になっていること（例えば定期的な進捗会議や成果物確認会議、工程ごとの成果物の判定会議など） ●成果物の確認は、形式的なチェックだけではなく、現物の中身も含めてきちんと状況を確認するように計画しておくこと ●契約の内容をもとに確認項目を洗い出してあること	
S31	顧客	顧客のプロジェクト目的は明確か	●プロジェクト計画書に開発側の目的だけでなく、顧客の目的も記載していること ●顧客の目的に関する記述内容について、顧客側の資料をベースにしている、もしくは、顧客によって承認されていること	
S32	技術	方式要件（負荷・性能・信頼性・安全性・保守／運用・移行）と方式設計結果は妥当か	●方式要件と方式設計結果に関するレビューを実施していること ●方式設計レビューのレビュー者が十分なスキルや知識を持っていることを確認すること	

	マネジメントにおけるヒント	対策案
	●口頭による約束では「言った」、「言わない」の問題になる ●文書化された内容について顧客側、関係会社（自社、協力会社含めて）それぞれのステークホルダーが合意している必要がある	●顧客とのコミュニケーションで重要な項目は文書でやり取りする ●口頭による依頼が慣習化している企業もあるので、文書で交わす手続きを慣習化させる努力が必要な場合もある
	●それぞれの会議体がどのようなことを議論し、決定する場であるのかを明確にしておく必要がある ●会議体の設定では、会議開催の頻度や参加者、目的などを事前に計画し、ステークホルダーに周知しておく必要がある。加えて、議事録を取って承認してもらうなどの手続きについても明確にしておく必要がある ●計画だけではなく、実質的にその会議体が稼働していることが重要である ●公式な場では言えないこともあるので、非公式なコミュニケーションパスにも配慮する。ただし、決定事項は会議体の中で確認する必要がある	顧客、協力会社など、プロジェクトの状況を見て必要な会議体を設定し、定例化する
	トラブル対応や調査依頼に対しての中間報告を行い、随時情報を提供する必要がある	社内おける会議体、顧客との会議体を決め、プロジェクト全体の会議体スケジュールを作成する
	●リスクの洗い出しには、ステークホルダーが集まって討議することが有効である ●リスク洗出しの際に効果的な視点としては、次のものがある ・成果の量の増減 ・成果の質の優劣 ・実施タイミングの是非 ・スキルの充足度 ・セキュリティ ・変化対応の柔軟性の有無 ●プロジェクトの状況によってリスクは変化するため、リスク管理表は定期的に見直す必要がある	チーム・メンバーを集めてリスクの洗い出しを行い、洗い出したリスクが発生する確率と発生した場合の影響を判定する
	●契約時に作業進捗・成果物のチェックやレビューのやり方を詰めておく ●確認時にはその報告や成果物の内容チェックまできちんと確認していることが重要。形式チェックだけでは成果物の出来を確認することはできない	協力会社に協力してもらい、作業報告とともに成果物（特に仕様書などのドキュメント）を最新の状態にするように管理する。 特にソースやシステム環境については構成管理チームを作って徹底的に管理するようにする
	●顧客との交渉を円滑に進めるために、顧客がシステム開発を通して、何を実現したいのかを把握しておくことが重要である ●顧客の目的や目標はプロジェクトのスコープに大きく影響する。そのため、顧客から示された目的や目標に対して、思い込みによって過大評価・過小評価することのないよう、プロジェクト・マネージャは顧客のプロジェクト目的について十分に確認し、明確化しておかなければならない ●プロジェクトの目的が明確になっており、顧客側の目的・目標と整合していることが、システム構築プロジェクトを成功させるうえで重要である	●プロジェクトの目的（目指すべきゴール）が明確になっていない場合、まずは顧客のキーパーソンと個別に打ち合わせをして目的・目標を聞き出しておく必要がある ●顧客側のキーパーソンが、明確なゴールを思い描けていない場合も考えられるが、この場合はプロジェクトの最初の段階で、この目的を明確化させる検討を進める必要がある。顧客側での課題認識について意見収集を行い、プロジェクトの活動方針を検討する
	●顧客に責任を持って方式要件を決めてもらい、それに基づいて設計する。ただし、顧客だけでは方式要件は決められないこともあるので、ベンダー側も方式要件の検討を支援する必要がある ●有識者（実装経験がある者や、その製品・技術の中身に詳しい者）に要件と設計結果の妥当性をレビューしてもらう ●方式設計において不安項目がある場合は、プロジェクトの初期段階でプロトタイピングを行うなどの施策を講じること	●技術や製品をよく知っている技術者だけではなく、システム構築・導入の経験者や保守フェーズの担当経験者なども入れて方式検討をするべきである ●新規のWebサービスなどで入力件数と出力予定帳票や更新データ件数などが想定できない場合は、上限を設定し、それを超えたときは、オーバーフローメッセージを出すなどの、トラブル対策について検討する計画をしておくこと ●システム障害時の対応想定として、回復優先順位付けを行う必要がある

No.	知識エリア	チェック項目	評価基準	
S33	組織	プロジェクト体制について、顧客も含めたステークホルダーの責任分担が明確になっており、体制上の不備不足や不安はないか	●プロジェクト体制図が作成されており、顧客側・ベンダー側含めた全体の体制について、想定しているプロジェクトの活動目標を遂行できるメンバー・組織構成になっているかの確認ができていること ●チームごとの責任分担が明確になっていること	
S34	組織	プロジェクト内のメンバー・組織のミッションが明確で、それぞれが自分の責任を意識した行動をしているか	●各メンバー・組織のミッションについて、各者がそれを理解し、合意していること ●プロジェクト体制図が作成できており、各メンバー・組織ごとのミッションが明確になっていること	
S35	基本動作	基本となる動作（コミュニケーションルール、コーディング基準（コメント、ステートメントの書き方、禁止命令語など）が徹底されているか	●「打ち合わせを行うときは議事録をとること」、「設計内容はドキュメント化すること」など、基本的な作法について全メンバーが問題なく認識していること ●セキュリティに対する行動規範が明文化され、徹底されていること	
S36	基本動作	部下の状況を把握しているか	公式、非公式のコミュニケーションを部下と行っていること	
S37	モチベーション	プロジェクト計画に対してメンバーの気持ちがしらけていないか	●プロジェクトの日程計画や実現可能性などに関してメンバーが不信感を持っていないこと ●特にプロジェクト・メンバーや協力会社に対して、プロジェクト計画上の問題点がないか、意見を求めていること	
S38	課題管理	課題管理表などで課題が管理されているか	課題（解決しなければならない既出の問題）の発生日時や課題の内容、重要度、状況、対策予定日がわかるような管理表やそれに類する資料によって課題を随時管理していること	

	マネジメントにおけるヒント	対策案
	想定されている活動計画を遂行するに当たり、以下のような観点でプロジェクト体制を確認すること ●プロジェクト推進上必要な権限を持っている人がいるか ●必要な技術・スキルを持った人や組織が入っているか ●システムを利用する利用者側を取り仕切ることができるメンバーがいるか ●プロジェクトの責任者が明確になっているか ●顧客側とベンダー側とで、体制上のパワーバランス、人数バランスに偏りがないか（質・量でバランスが取れているほうが望ましい）	●分担と責任が明確になっている必要があるが、実施能力がない場合や能力が不足している場合は、外部からの人材補充などにより再構築する ●顧客側責任でプロジェクトが遅れる場合は、納期やシステム開始が遅れることについて合意しておく
	●いつまでに誰が何を行わなければならないかを確定することが重要である ●各メンバーにもミッションの内容を確認してもらい、合意してもらう必要がある。ミッションが不明確な場合、のちのち作業範囲、責任分担について問題になることがある ●各メンバーに対し、そのミッションについて担当可能かどうかを判断してもらっておく必要がある。この判断をしてもらうことによって、プロジェクト参画に対する責任感を高めてもらうことができる	●必要なスキルを明らかにして、作業内容をWBSによって明確に分割しておのおのの作業の分担と責任を明確にする ●作業間の責任は誰の責任作業かを明確に定義する必要がある ●スキルが不足している場合は、外部研修を含む適切な教育を行うか、または協力会社から社員代替要員を確保する
	●チーム全体が「多少手抜きをしても大丈夫」という雰囲気にならないように、決められたことはきちんと守ることを徹底して指示する ●コミュニケーションを良くすることと、「なあなあで済ます」ことは別であり、緊張感を持って仕事に取り組むことが大切である ●先輩が後輩の良いかがみにならなければならないという雰囲気を作ること	●基本動作についての教育的な指導を普段から心がけるようにする ●すべての基本動作徹底が難しい場合は、プロジェクト状況により重点的に徹底する事項を絞る。 例えば、 仕様がふらつく傾向がある→ドキュメント化の徹底 進捗が遅延する傾向がある→進捗報告の数値化と見込み提出の徹底
	●部下から何の報告もない場合は、報告が上げづらい問題を抱えているときもあれば、話しづらい雰囲気などコミュニケーション上の問題があることがある ●定例進捗会議などの会議ではないところででも、コミュニケーションを随時行っていることが大切である ●「何か問題を抱えてはいないか」と声かけをするなど目に見える行動をすること。心配していることを頭で思っているだけでは相手に伝わらない	●直接部下に状況をヒアリングしたり、部下とかかわりのある第三者にヒアリングして部下の状況を多面的に把握する ●権限委譲と丸投げを誤解しないようにする。部下、協力会社に作業を任せるとき、責任まで投げず、共有するようにする
	●プロジェクト責任者よりも、実作業を担うことになるプロジェクト・メンバーや協力会社のほうが、作業工程上の問題や負荷の問題などが見えていることもある。実現可能性について積極的に意見を集めるべきである ●スケジュール、コストのベースラインは、プロジェクト・メンバーが努力すれば、守ることができる水準である ●メンバーの表情や言動を見ることも大切である。作業の質が落ちていると感じたり、粗暴になったりしている場合は、モチベーションが下がっていないか、会話をするべきである	●納得していないメンバーに対して、どの部分が納得できないのかをヒアリングし、問題点を明らかにした上で問題認識を全メンバーで共有化する ●「しらけ」を抱えたままの場合は、チーム全体として著しくモチベーションが下がってしまうリスクがある ●クリティカルパス上の作業には比較的「しらけ」の少ないメンバーを配置する ●作業配分、リソース配置を工夫し直近のマイルストーンをクリアし続けることで、実現可能であることを納得させていく
	●課題管理表の内容が、形式的でないことが必要である ・記載されている内容は本当に課題か ・課題の責任者が明確か ・解決までの経過が記述されているか（目標日変更も含めて） ・解決目標日と完了日に矛盾はないか ・目標日を過ぎている未解決項目はないか ・特定の課題に偏っていないか ●課題に対して顧客側、ベンダー側双方に「解決する」意志を持って取り組んでいるかを常にウォッチすること。「これは顧客側の課題だから」と放置しないこと	顧客側、開発側のキーパーソンを集めて課題を抽出し、一覧表にまとめ、管理する。課題については、管理するだけではなく、ステークホルダー間で共有化し、管理が形骸化しないようにする必要がある。具体的には、 ・議事録に課題がまとめて明記し、次回の会議で必ずフォローする ・ホワイトボードに書き出す、ポストイットを貼り付けるなど、課題を全員で共有するようにする

付録 2. ヒアリングシート

No.	知識エリア	チェック項目	個別のヒアリング要領	
H1	技術	通信制御系・上流工程の成果物の確認	要求仕様、機能仕様書が、他の業務システム並みの文章、一般図表で書かれているだけでは不十分。中流工程以降、開発規模が過大とならないか、考慮漏れは無いかといった視点でヒアリングを実施	
H2	技術	多重アプリケーション・プログラム走行環境におけるデッドロックも想定し詳細設計したか	特に旧資源へのアクセスを行うプロセスは、それだけ資源保留期間が長くなり、多重アプリケーション・プログラム走行環境ではデッドロックに遭遇する確率が高まることにも注意する	
H3	技術	ネットワークは十分な容量設計を行ったか	ネットワークは、異常時パケットの再送などで指数関数的にトラフィックが急増することもあるため、ネットワーク技術に長けたSEの協力を得て、設計する必要がある	
H4	技術	アプリケーション・プログラムの走行性能低下が発生する要因と、その影響を調査・検証しているか	アプリケーション・プログラムの単体走行時性能に加え、アプリケーション・プログラム多重走行環境でアプリケーション・プログラム同士が同時には利用できない共用資源などSRR（Serially Reusable Resources）を調査し、それによる性能低下が無いか検証していること。また、特に高レスポンスが要求されるアプリケーション・プログラムプロセスの実行優先度が高くなるように設計していること	
H5	技術	システムのライフサイクル運用が設計されているか	システムのライフサイクルを通じた運用の仕様が明確になっていること	
H6	技術	市販製品間インターフェースの相性をチェックしたか	システムが使う製品バージョンの組み合わせが連動保証の範囲か、各製品ベンダーに確認すること	
H7	技術	信頼性設計ではハードウェアとソフトウェア両方とも含むシステム全体を俯瞰した設計を行っているか	信頼性制御ソフトの作りこみのみで信頼性を確保しようとしていないこと。また、システム全体の信頼性バランス（ハードウェア、ソフトウェア）を考慮し、信頼性設計の全体が俯瞰できる方式で信頼性設計を行っていること	
H8	技術	システムの過負荷、限界系の設計をしているか	想定可能な過負荷、設計容量を超える資源要求発生などのケースが検証されているかチェックすること	
H9	技術	アプリケーション・プログラム資源確保方式の検証を行っているか	特に、アプリケーション・プログラムに必要な資源確保をアプリケーション・プログラムごとにするか、システム全体で確保すべきか、詳細設計当初に、システム全体に求められる要件から検討・決定しているか、チェックすること	
H10	技術	同一製品体系のソフトのバージョンアップ（特に、制御系システム（オペレーティングシステム・ミドルソフトなど））時は、互換性について制約事項を含めて十分な検討・検証を実施しているか	バージョンに関する検証状況を確認し、その際表面に見えるソフトウェアのバージョンだけしか情報が出てこない場合は、制御系について検討が漏れている可能性を含めて確認を行う	
H11	技術	ユーザー指定ソフトの扱い、制約、問題発生時の処置方法、作業範囲、役割分担などをユーザーとの間で確認しているか	文書（契約書を含む）で記録を残しているか。顧客との間で作業範囲や役割分担を明確にする資料は共有されているか。ユーザーの承認は通っているかを確認する	
H12	技術	●新しい言語や技術を採用する際に適用分野での実績を確認したか ●実績が無い場合、問題に備えたコンティンジェンシー・プランを持っているか	実績確認では口頭でなく、製品マニュアルなど公式文書で確認する	

	評価基準	エビデンス・確認方法	対策案
	●スコープが有限範囲であることを全体を俯瞰しながら確認できる（1枚の状態遷移図に書ききれる）こと ●各状態ですべての内外トリガ（コマンド／オーダー受信など）を考慮した設計となっていること	状態遷移マトリクス （最低でも状態遷移図）	状態遷移表などを用いた網羅性検証、開発スコープが「現実に開発可能な範囲」であることの検証を行う
	●デッドロック対策が明確であること ●対策実施後のデッドロック発生事象を網羅的に列挙し、その影響を評価していること	詳細設計書とそのレビュー証跡	●1つの方法として、デッドロックを起こさないように、システム資源群に対して、アプリケーション・プログラムからアクセス順序を取り決める方式がある ●それが取れない場合、デッドロック発生を許容するが、デッドロック検出～強制解除までの監視時間を短くするチューニング設計を行う方法もある
	技術的根拠を持ってネットワーク設計を行っていること	詳細設計書とそのレビュー証跡	万一の異常時に備え、運用仕様も設計しておくこと
	●SRRの抽出方法が、思いつきではなく、論理的・体系的であること ●SRRへのアクセス頻度を加味した影響を、客観的手法で査定していること ●システム性能要件と実行優先度の設計とが整合していること	詳細設計書とそのレビュー証跡	左記の基準に準じて設計を見直す
	システムのライフサイクルを通じた運用イベントを網羅していること（週次運用、月次運用、年跨り運用、世紀跨り運用、閏年対応運用など）	詳細設計書とそのレビュー証跡	左記により終局までの運用（自動、手動）が正しく実施可能であることを検証すること
	ベンダー確認の他、社内外での実績の情報も確認した上で、相性について判断していること	ベンダーのホワイトペーパー、その他実績情報	連動時に問題が発生するリスクを踏まえ、連動動作確認のスケジュールを組むこと
	システム全体（ハードウェアとソフトウェア両方）を俯瞰した検討がなされているかを確認する	詳細設計書とそのレビュー証跡	システム全体を俯瞰した検討を実施する。これに加え、信頼性設計の範囲は、顧客と合意しておくこと
	CPU、メモリー、データベースなどシステムが利用する資源それぞれについて論理的・体系的な抽出方針に沿って、網羅的に検証が実施されていること	詳細設計書とそのレビュー証跡	要件に合わせた性能設計を実施すること
	アプリケーション・プログラム資源の確保、解放の頻度を、システム全体として検証していること	詳細設計書とそのレビュー証跡	アプリケーション・プログラム資源の確保・解放がアプリケーション・プログラムのアクセスごとに頻繁に行われる場合には、システム立上時点で当該資源を一括確保するように設計すること
	導入される各ソフトのバージョンアップについて検証された資料が存在するか。また、バージョンアップに際しての制約が顧客と共有されているか	●役割／責任分担表 ●契約関係文書	現在把握している情報と、把握されていない情報を明確化し、現状を踏まえてそれらの作業を実施するタイミングや責任分担を調整する
	全体を俯瞰できる情報を元に、資料間の整合性が確保されているか。隙間が発生していないか。曖昧な約束事項が無いかを確認する	●役割／責任分担表 ●契約関係文書	現時点から先の作業分担や責任分担について、明確化する資料を作成し、顧客と調整することで、漏れや意識差異をなくす
	製造工程だけの生産性だけにとらわれず、試験での解析容易性、性能など納品までの全工程を見渡した検証を行う	言語ベンダーの実績レポート	新しい言語を採用する際には、コーディング前に性能条件が厳しいオンラインシステムや、複雑なObject間インターフェースがある制御系ソフトなどへの適用実績を確認する。また、実績が無い場合、問題が発生したら、代替言語で開発するコンティンジェンシー・プランを織り込んで、中流工程をすすめる

No.	知識エリア	チェック項目	個別のヒアリング要領	
H13	顧客	大規模な仕様変更に対して、 ①プロジェクト・マネージャは顧客に対して、リスクを伝えて、対策について折衝したか ②それでも無理があるのであれば、プロジェクト・マネージャの上司／営業にエスカレーションしたか ③顧客が、リスクを知った上で最終的に仕様変更を決行する場合には、悪影響を最小化するよう提案したか	あらかじめ、大規模仕様変更における「大規模」の基準を設け、基準を超える仕様変更が最終的に意思決定する前のタイミングで左記①～③のヒアリングを実施	
H14	顧客	仕様未凍結に対して、 ①プロジェクト・マネージャは顧客に対して、リスクを伝えて、対策について折衝したか ②それでも無理があるのであれば、プロジェクト・マネージャの上司／営業にエスカレーションしたか ③顧客が、リスクを知った上で未凍結のままプロジェクトを進める場合には、悪影響を最小化するよう提案したか	あらかじめ、「未凍結による影響範囲」の基準（原価全体に占める未凍結相当額など）を設け、基準を超える仕様未凍結が存在する場合、左記①～③のヒアリングを実施	
H15	コスト	①システム化の規模見積もりを行っているか ②その際の見積根拠を記録してあるか	●実施した見積もり結果の位置付けがはっきりしていること（予算取り、ベンダー調達の参考情報、顧客との契約） ●見積もり方法、見積もり基準（生産性の想定値）など、算定方法を明確に残していること ●開発規模、体制、生産性、開発期間が整合するか第三者チェックを受けていること ●再見積もりの際に、前回との見積金額差異が、説明できるようになっていること	
H16	コスト	システム化費用において、当初計画からの「振れ」を調整しているか	●当初の計画との差異を費用、工数、機能のそれぞれについて管理しているか ●工数山積みがコスト制約に見合っているか	
H17	スコープ	要件定義を受けた仕様はお客様から承認を得ているか	業務要件、システム機能について、お客様の承認を得ていること	
H18	スコープ	仕様変更ルールがお客様と取り決められており、ルール通り運用されているか （変更の許容範囲、変更の受付から変更反映までのプロセス、仕様変更の承認）	●業務要件、システム機能について、変更管理のベースが存在すること ●エンドユーザー部門含めた仕様変更・追加の場があること ●納期、コストとの関係を考慮してコントロールしているか ●成り行きでシステムのスコープが変わらないようにしているか ●各ステークホルダーに変更管理手続きを徹底させているか	

	評価基準	エビデンス・確認方法	対策案
	①リスクの影響やその軽減策に関し、プロジェクト・マネージャが顧客に対して説明責任を果たしたか ②プロジェクト・マネージャのエスカレーションと上司／営業の判断や行動が実際に行われたか ③顧客が、最終的に仕様変更を決行する場合には、リスクの発生に備えた対策を提案したか	プロジェクト・マネージャ／上司・営業と顧客との折衝議事録	①まずプロジェクト・マネージャは緊急機能追加に見合う、プロジェクト負荷軽減策（既計画機能の納期変更、緊急機能の段階リリースなど）について顧客と折衝する ②それでも無理があるのであれば、プロジェクト・マネージャではなく、プロジェクト・マネージャの上司自らが顧客や顧客の上司にリスクを伝え、顧客の対応を要請する ③リスクを伝えられた顧客が、それでも仕様変更を決行する場合には、ベンダー任せで済ませず、リスクの発生に備えた、コンティンジェンシー・プラン（システム迅速復旧体制など）を自ら用意するように顧客と調整する
	①リスクの影響やその軽減策に関し、プロジェクト・マネージャが顧客に対して説明責任を果たしたか ②プロジェクト・マネージャのエスカレーションと上司／営業の判断や行動が実際に行われたか ③顧客が、最終的にプロジェクトを続行する場合には、リスクの発生に備えた対策を提案したか	プロジェクト・マネージャ／上司・営業と顧客との折衝議事録	①プロジェクト・マネージャは基本部分、他のインターフェースが多い部分のようなサブシステムは、先行着手しない。仕様凍結の遅れ、未実施がもたらすリスクを冷静に顧客に伝え、以降の計画について顧客と折衝する ②それでも無理があるのであれば、プロジェクト・マネージャではなく、プロジェクト・マネージャの上司自らが顧客や顧客の上司にリスクを伝え、顧客の対応を要請する ③リスクを伝えられた顧客が、それでも仕様変更を決行プロジェクトを続行する場合には、ベンダー任せで済ませず、リスクの発生に備えた、コンティンジェンシー・プラン（システム迅速復旧体制工期延長、リリース機能削減など）を自ら用いる
	プロジェクトのシステム化対象範囲について規模見積もりを行っていること。また、その見積もり根拠が明確になっていること 見積もり根拠の例としては、 ●現行ステップ数 ●過去の類似プロジェクト ●機能数やファイル数からの係数見積もり ●複数有識者の意見集約（デルファイ法） などがある	①システム化規模見積もり	●工程の進行に伴い、より精度の高い見積もりを行っていくことを計画に盛込む ●見積もり結果について、前回との差異がどこから発生しているかを、顧客に説明して合意（納得）を得る
	コスト制約に見合った機能削減や請負金額の増額、スコープの変更などの対応を実施していること	①工数山積表 ②収益管理表 ③変更管理票	コスト制約に見合った対応策をあらかじめ決めておく 例えば仕様変更については、顧客と費用や実現機能についての調整を随時できる仕組、関係を築いておく
	スコープがお客様と握られていること	①議事録	承認手続きがない場合は、承認のための手続きを明確にする
	システム対応のスコープそのものが記述されているドキュメントを特定でき、ドキュメントが存在すること ●窓口、取りまとめ者が明確になっていること ●解決期限が明確になっていること ●実際の実施に関して問題がないこと ●変更管理手順が明確になっていること	①変更管理表 ②変更管理ルール	●変更管理手順がない場合は、変更管理についての手続き・手順を明確にする ●変更管理の実施に問題があるかを関係者にヒアリングし、問題がある場合は、どのような問題点があるかを聞き出し、改善する

No.	知識エリア	チェック項目	個別のヒアリング要領	
H19	スコープ	顧客側ステークホルダーが多い場合、仕様決定プロセスは決まっているか	エンドユーザー部門含めた仕様変更・追加の場があることを確認	
H20	スコープ	ベースラインの見直しを伴うスコープ変更への対応策が準備されているか	●スコープ増加時の対応策（機能削減、費用増、スコープ変更）を顧客とあらかじめ合意しておく ●スコープ増の場合は、テストの増大やクリティカルな機能に対する影響度合を確認する	
H21	組織	中流工程における営業のプロジェクト参画状況は、仕様変更頻度などプロジェクトの営業関連課題の大きさと照らして、適切か	●仕様変更、納期変更などに伴い営業として第一人称での対応が必要な価格折衝、見積もり、契約などが放置されていないか確認する ●営業が顧客を定期的に訪問し、顧客とのコミュニケーション・チャネルが良好に維持されていることを確認する	
H22	組織	●中流工程において、プロジェクト・マネージャがプロジェクトで必要とされる能力を持ち、それを十分発揮しているか ●プロジェクト・マネージャの能力を超える課題・リスクがある場合には、組織的な支援が十分あるか	●プロジェクト・マネージャが、システムの難度、対顧客対応の難しさに見合った能力(経験・スキル）を持っているか ●またプロジェクト・マネージャは、その能力をフルに発揮できているか ●プロジェクト・マネージャの上司がプロジェクト・マネージャの抱える課題について、現場訪問などコミュニケーションを蜜にして状況を把握するなど、プロジェクト・マネージャの所属組織に、火を噴く前に支援する文化があるか ●プロジェクト・マネージャの所属組織が、プロジェクト・マネージャの顧客との相性（文化・価値感の差）について、チェックしているか	
H23	組織	ミッション・クリティカルなシステム開発プロジェクトにおける中流工程を開始する場合、アプリケーション・プログラム開発担当だけでなく、システム全体方式の詳細設計・検証ができる技術者が体制に含まれているか	ミッション・クリティカルなシステムでは、様々な性能ボトルネック、信頼性詳細設計上の盲点などが存在する。業務アプリケーション・プログラムからはブラックボックス化された基盤技術（ハードウェア、ネットワーク、OS、データベースなど）の仕組みも分かった上で、個別業務アプリケーション・プログラム開発担当では見落としがちな開発の穴埋めをしてくれる「方式技術者」を配置しているか	
H24	組織	発注者側のステークホルダーとして位置付けられている各組織の責任・権限は明確になって、機能しているか	●発注者側担当者とエンドユーザー部門との責任分担が明確で、両者のコミュニケーション・意思疎通が十分とれているか（これが不十分であることによる仕様追加多発などの恐れが無いか） ●発注側プロジェクト・オーナーが存在し、プロジェクト参画意識が高いか ●また、ステアリングコミッティが必要な場合、それが組織され、実際に機能しているか	

	評価基準	エビデンス・確認方法	対策案
	中流工程でもステアリング会議体制を維持しているか	ステアリング会議議事録	顧客側ステークホルダーが多い場合、ステアリング会議(※)を中流工程でも維持 (※)ステアリング会議 お客様のステークホルダーが多い場合、お客様であるエンドユーザーも含めたメンバーで構成し、プロジェクトマネージャが主催し、機能要件、非機能要件などの仕様について確認、調整、決定する場である
	●スコープ増を定量的に押さえていること ●許容範囲かどうかをコスト、スケジュールの観点から評価していること	①変更管理表 ②議事録	●許容範囲でない場合、どの程度の増があるのかを明確にした上で、顧客との打ち合わせを実施する ●大幅な機能範囲や作業範囲の変更がステークホルダーの合意の下に行われていく。変更にともなうコスト増、納期遅れについても顧客の了解を得る
	●仕様変更要求案件の発生日～クローズ日までの措置状況を評価する ●仕様変更要調整打合せにおける営業の参画状況、営業の顧客訪問頻度を評価する	仕様変更案件管理簿 顧客との価格などの折衝議事録	営業に「受注額」しかノルマを与えられない場合、別の新規案件に注力する方が成績があがるので、受注済みの既存プロジェクトを軽視しがちになる。SI会社として、一括請負契約締結後にも、営業がプロジェクト・フォローを行うようなインセンティブ制度(着地利益責任を営業にも持たせ、仕様変更対応を優先的に行う人事評価など)を設け、既受注プロジェクトのフォローも怠らない仕組みを作るのも一案
	●プロジェクト実行計画書とプロジェクト進捗状況報告書との比較、進捗報告における課題・リスク措置状況、顧客からの評価(営業担当経由で入手)などを総合的に勘案して、プロジェクト・マネージャの実践能力を評価 ●過去の同等環境のプロジェクト経験とそのプロジェクト成否を調べ、プロジェクト・マネージャのスキルを判断。それと実際のプロジェクトの状況とを比較することで、プロジェクト・マネージャ能力の発揮状況を確認 ●プロジェクト・マネージャの顧客との相性(文化・価値感の差)について、チェック(営業担当の力も借りる)	①プロジェクト実行計画書 ②プロジェクト進捗状況報告書	プロジェクト・マネージャの所属組織において、プロジェクト状態の監視、プロジェクト内部のモニタリングを行い、下記の制御を行う ●プロジェクト・マネージャに対して的確なプロジェクト・マネージャ教育を行う ●経験浅いプロジェクト・マネージャに対しては、コーチを付けながら、プロジェクト完遂まで見届けさせる体制を検討する ●プロジェクト計画書を有識者でレビューさせる ●工程の節目などの必要時、PMOによる第三者審査などの支援を要請する ●営業の支援を要請し、プロジェクト・マネージャの対顧客性、ビジネスマンとしての資質が合わない場合、先手を打ったプロジェクト・マネージャ交代などを検討する必要がある
	プロジェクト・マネージャ、業務アプリケーション・プログラム担当、基盤系専門担当に加えて、そのほかの担当責任者が体制図上で明らかになっていること。(その担当者がシステムリソース排他制御のための各アプリケーション・プログラムからの標準アクセス法の提供、システム全体性能設計などのSE・検証できていること)	プロジェクト体制関連図(役割分担表、要員遷移図)	●プロジェクト内で、専属のシステム全体方式技術者を確保することが望ましい ●しかし、そのような優秀な技術者が豊富に存在することは稀であり、この場合には、プロジェクト・マネージャの所属組織にて方式技術者を「方式部門」としてプールし、各プロジェクトが必要とするタイミング(詳細設計レビューなど)で、方式技術者による支援を行う体制をとる
	事前に定義しておくべきこと ●工程別の組織 ●各組織の責任と権限 ●会議体と目的 マネジメント・レベルの会議体も準備し、想定外の問題にも対処可能としておく	①プロジェクト体制関連図(ステークホルダー俯瞰図、プロジェクト推進体制図) ②会議体	●プロジェクト体制関連図の見直しと責任範囲の明確な宣言を受注者側プロジェクト計画書(発注者側プロジェクト計画書)などで文書化する(ように発注者側に提案する) ●特に、プロジェクト・マネージャがプロジェクト・オーナー、エンドユーザーのプロジェクト参画・支援を提案しても実現できない場合、プロジェクト・マネージャは問題のエスカレーションを行う。それを受けた上司は、営業の協力を得て、本来の体制へ迅速な是正措置を行うこと

No.	知識エリア	チェック項目	個別のヒアリング要領	
H25	タイム	実施するべき作業が明確に定義されているか	●作業のリストアップができているか ●リストアップはできているが、その作業で何をするのかが明確にされているか	
H26	タイム	担当の作業負荷や作業の重なり具合、空き具合についてチェックしているか	●管理者が各担当ごとの作業負荷が計画と整合が取れているかチェックしているか ●適切な、進捗指標を使っていること。進捗報告が利用されていること。現物(実物)を見て管理しているか ●終了したことを証明するエビデンスが規定されており、終了の際に実物で確認しているか ●進捗数値の計算方法を決めているか	
H27	タイム	クリティカルパスは明確か	●クリティカルパス候補を特定できているか ●クリティカルパスを認識し作業管理されているか ●変動的なクリティカルパスを、日々モニタできているか ●ボトルネックとなりそうな人員を識別できているか ●作業バランスがとれているか	
H28	タイム	マススケジュールと要員山積みとの整合性は十分か	●当初の計画との差異を見ているか ●作業ピークを計算した山積みになっているか	
H29	タイム	マススケジュールとチーム別スケジュールとの整合性は十分か	●プロジェクト管理ツールを利用するなどして、整合性をとるようにしているか ●複数のベンダーが出したスケジュールの1つに変更があった場合に、全体的なコントロールができ、マススケジュールとの整合性をとっているか ●それぞれのスケジュールに根拠があり、単純な人月計算でスケジュールを作っていないか	
H30	タイム	プロジェクト線表の進み・遅れをどのように捉えて表現しているか(例えば、稲妻線の記述) 捉え方の根拠は明確か	●進捗率の計算方法をあらかじめ決めているか ●進捗率に対応した成果物ができているか	
H31	調達	【オフショア開発を委託する場合】 仕様の記述はオフショア先の理解を得られるレベルか	日本語が堪能なブリッジSEがいるオフショア会社でも、従来の国内外注先と同じ記述レベルの仕様書でプログラム設計～単体試験を発注すると失敗することがある。それを防ぐために仕様書の記述内容・レベルについて規定されており、オフショア先と合意が取れているか確認する	
H32	調達	【オフショア開発を委託する場合】 オフショア先は開発プロセスが統制されているか	個々のプログラマ任せの場合に生じ得る問題を防止するため、コーディング規約、構成管理などの開発標準が存在し、それが順守されていることの確認をしたか、を確認する	

	評価基準	エビデンス・確認方法	対策案
	●WBSを基にスケジュール表が作成されていることが前提となる ●各タスクの作業規模見積がなされて、いつまでに実施すべきタスクなのかが共有できていること ●各タスクの順序だてのつじつまが合っていること	①スケジュール表 ②WBS ③見積もり ④責任分担表(体制表) ⑤打合せ議事録	●キーパーソンを集めて、やるべき作業のリストアップ作業を行う ●リストアップされた作業について、作業内容と成果物を明確にし、作業量の見積もりと作業スケジュールを明確にする ●担当者をアサインする ●新しい領域での最初の開発など、業務ノウハウがない場合は、開発が進むにつれてやらなければならない作業が見えてくるので、新たに明らかになった作業項目を管理対象に加えるとともに、全体を見て、作業分担の偏りがないかなどを確認する
	プロジェクト・マネージャと各担当の双方がチェックしていること	①スケジュール表(進捗管理) ②WBS ③見積もり ④責任分担表	進捗会議などで作業負荷状況を定期的にチェックするようにする。具体的には、開発メンバーを少人数のチームに分割し、チームごとに作業を割り振り、作業分担や作業負荷の調整はチーム内で行う。チーム内で調整できない、影響の大きい項目だけを全体(チーム間)で調整する方法が効果的である
	クリティカルパスに当たる作業項目が洗い出せていること	①スケジュール表 ②WBS	クリティカルパスに遅れがある場合はキーパーソンを集め、早急に問題点の洗い出しを行い、対策を取る。クラッシング、ファーストトラッキングなどの手法がある
	マスタースケジュール作成時に要員山積みを作成して整合性を確認していること	①マスタースケジュール ②要員山積み表	●要員調整計画を見直す ●ピーク時を想定して要員が足りなければ、開発作業と並行して要員を調達する
	マスタースケジュール作成→チーム別スケジュール作成→マスタースケジュール改訂のステップを実施して両者の整合性がとれていること	①変更管理表 ②議事録 ③マスタースケジュール ④チーム別スケジュール	マスタースケジュール作成→チーム別スケジュール作成→マスタースケジュール改訂のステップを実施して両者の整合性を取る。具体的には、マスタースケジュールを理想のスケジュールとし、チーム別スケジュールは現実の積み上げスケジュールと捉え、作業を進めながらチーム別スケジュールをマスタースケジュールに近づけるよう調整する
	進捗率の値が感覚的なものではなく、きちんとした根拠に基づくものであること	①スケジュール表 ②WBS	●進捗の評価方法を形式化／文書化し、進捗報告者に徹底させる ●成果物の作成状況や現場の雰囲気なども把握して、進捗率の報告は感覚的なものになっていないかを確認する ●残りの作業項目を明らかにしたい場合は、進捗率に加えて、残作業工数での管理を追加するとよい
	仕様書の記述内容、記述レベルが規定されていること	●発注仕様書への要求事項(記述レベルなど) ●発注仕様書のサンプル	仕様書の一部をまず作成し、それをオフショア先に提示して、オフショア先が理解の程度、改善すべき点を確認する
	次の確認が取れていること ●オフショア先開発標準がCMMIなど体系的プロセス標準として規定されていること ●それらが順守されていること	●オフショア先選定チェックリスト ●オフショア先選定理由書	●オフショア先選定前に、開発プロセス標準およびそれに基づいて作成したドキュメントを調べる ●標準が不十分な場合、自社コーディング規約を開示するなどして、全体統制がとれるようにする

No.	知識エリア	チェック項目	個別のヒアリング要領	
H33	調達	【パッケージを活用する場合】 製品・ベンダーを含めた品質を判定しているか	●評価基準に基づく選定会議が実施されているか ●選定会議において、採点評価がされているか ●システムの中でのパッケージの利用想定と、パッケージ製品の提供機能が整合していないのに（フィット&ギャップ分析の結果、ギャップが大きいのに）使おうとしていないかを確認しているか	
H34	調達	【協力会社を使う場合】 作業の内製、外部委託の判定を行い、調達作業を決定しているか	●外部調達とした理由、範囲、期間が明確になっているか ●協力会社の力量を十分掴んでいるか ●協力会社の概況を把握しているか ●力量評価に基づいて、状況把握のための定例会議を開催しているか（営業を同席させる、頻度を増やすなど） ●発注する際に約束事項を明確にしているか（例えば、不良が予定以上出た場合に品質向上策を実施する、瑕疵責任をとらせるなど）	
H35	調達	【協力会社を使う場合】 協力会社の進捗・品質に関する中間フォロー実施計画が明確になっているか	●請負契約は成果物さえ納入すればよいと考えていないか ●請負業者の品質方針・品質管理体制などの具体的な品質管理手順について確認を行ったか ●中間時点のチェックを約束・実施しているか	
H36	統合	プロジェクトの全体を俯瞰できているか	全体像を把握することで、各部分の対応の意味も明確になる ●対象システムの構造のほか、関連するシステムには何があるのかが把握できていること ●主要なステークホルダーとして、 ・システムと業務との関連性を把握できている人 ・システムを利用する人 ・財布を握っている人 ・システムを作る人 を把握できていること。関係者の名前だけでなく、各キーパーソンを明確にして、一番効果のある働きかけを行える相手を把握することが重要 ●大日程表の中で、どの部分がクリティカルパスとして重要かが指摘できていること （詳細は、俯瞰の考え方を参照）	
H37	統合	プロジェクトのドミナント・アイテムを把握できているか	ドミナント・アイテムとは、プロジェクトの成否に大きな影響を及ぼす重点項目。それを早期に見つけ出せることが成功のポイント ●俯瞰図に基づいてどのようにドミナント・アイテムを見つけ出したか（その理由など ）が明らかになっている ●体制上の各キーパーソンが何をドミナント・アイテムと捉えているかを自分のこれまでの経験と照らし合わせて把握できているか	
H38	統合	基準を超える大規模な仕様変更・追加要求が発生した場合の、プロジェクト計画も初期の計画策定と同等以上の精度で見積もったか	●あらかじめ取決めた基準を超える大規模な仕様変更の場合、プロジェクトのベースラインの変更が伴うが、上流工程でのヒアリングとは別に、仕様変更のタイミングでもヒアリングを行う ●対象は、その是正個所に閉じず、システム全体への影響をプロジェクト・マネージャが査定しているかどうか、という視点で審査する	

	評価基準	エビデンス・確認方法	対策案
	●使用するパッケージ製品の導入事例の数や、導入先の企業特性などを顧客と比較して、パッケージの品質を適切に判定して選定していること ●パッケージ・ベンダーのサポート品質レベルと、顧客の要求する品質レベルが整合しているかを考慮して選定していること ●技術的観点から評価できる有識者を交えたレビューが実施されていること	①パッケージ活用リスクチェックリスト	●ソリューションとしての的確性、品質、推進体制に関するリスク・マネジメントを事前に行い、リスク対応策を立てる ●実現すべき機能のうち、パッケージによる割合が低く、パッケージに問題があっても、新規開発により納期内に完了させることができる場合は、リスクを受け入れてもよい
	●実施責任者（プロジェクト・マネージャー、プロジェクト・リーダーなど）および有識者を交え、協力会社に依頼する作業が適切な規模、難易度であることを検討していること ●協力会社の要員数、技術・スキルのレベル、これまでの実績、業務ノウハウの保有について判断し、適切な協力会社を選択すること ●作業の外部調達の決定に際して、社内稟議を行っていること ●外部流出をしていけないシステム化範囲を委託していないこと	①協力会社の実績資料 ②協力会社との契約書（受託明細） ③協力会社マネージャとの面談結果	●想定されるリスク対応策を立てる ●作業の内外判定をしないで、従来からの協力会社に発注する場合は、今回のSOWに関する能力を探り、サポートするか、マネジメントをしっかりすれば対応できる場合は、その点に留意してマネジメントする
	進捗会議の定例化や品質基準の明確化についての計画がされており、ステークホルダー間で合意されていること	①進捗管理表 ②委託契約書（提出すべき中間成果物などが記述されていることがある）	●協力会社へのフォローに関して問題がなければよいが、フォローを行っていない状態である場合は、フォローの計画を立てて、実施する ●協力会社との信頼関係が築けている場合は、必要ない
	全体把握（全体俯瞰）の対象となる事項として、以下の３種類を確認する ●開発対象システム ●ステークホルダー（開発体制を含めることもある） ●スケジュール	①システム俯瞰図 ②ステークホルダー俯瞰図（体制図を含む） ③スケジュール俯瞰図	プロジェクトのドミナント・アイテムを把握し、対応するために、俯瞰図を作成する
	システム俯瞰図、ステークホルダー俯瞰図、スケジュール俯瞰図、要員遷移俯瞰図を用いるなどして、ドミナント・アイテムを指摘でき、対策を立てていること	①システム俯瞰図 ②ステークホルダー俯瞰図（体制図を含む） ③スケジュール俯瞰図 ④要員遷移俯瞰図	プロジェクト計画書において、ドミナント・アイテムに関するQCD計画、実施状況を重点的にマネジメントするような計画とする
	●上流工程におけるプロジェクト計画（新規作成）評価と、同程度の評価基準が必要となる ●プロジェクト計画の再策定となるため、新規プロジェクトの計画策定と同様な手続きで承認を得る	●変更要求分に対応するWBS、スケジュール、コスト見積もり ●当初計画でのWBS、スケジュール、コスト見積もり	●Scope：変更による影響の査定範囲がシステム全体となっており、新しいWBSの詳細度が、当初計画と同等以上であること ●Schedule／Cost：新しいScope、外注見積もりなどに基づき、当初計画と比べて同等以上の精度から、新しいスケジュール、コストが見積もられていること ●変更要求のプロジェクト計画と当初のプロジェクト計画に記載されている事項や記述に差がないかの確認をするとともに、お客様や社内権限者の承認を得る

No.	知識エリア	チェック項目	個別のヒアリング要領	
H39	統合	●作業計画、作業要領、規定類が準備されており、それに従って実施し、レビューなどを経て承認を得ているか ●プロジェクト・メンバーに周知されているか	●設計、レビュー、テストなどのプロセスのルール、プログラミングの標準化、構造化等の開発ルールが設定されているかを確認する ●また、開発ルールなどがプロジェクト内で統制されているかを確認する	
H40	統合	プロジェクトの日程、責任範囲、WBSなどがステークホルダ間で合意されているか（プロジェクトのWBSと顧客のWBSを比較した上で調整を行っているか）	●顧客とプロジェクトとの協働作業でやらないといけない ●お互いの作業分担を明確にすることがこの項目の主旨である	
H41	統合	構成管理のルールや手順が明確にされているか	構成管理のルールや手順が文書化されていること	
H42	統合	本番移行計画が明確か （移行計画には、人的移行、システム移行、データ移行などがある）	顧客との責任分担を明確にしていること。移行計画や移行に関する作業についても顧客との合意を得ておくこと	
H43	統合	保守・運用計画および作業手順・ルールを明確にし、レビューを実施しているか、あるいはその計画があるか	●プロジェクトのクロージングで問題になるケースがある ●統合テストより前の段階で確定しているか	
H44	リスク	納期前倒し要請に対して、 ①プロジェクト・マネージャは顧客に対して、リスクを伝えて、対策について折衝したか ②それでも無理があるのであれば、プロジェクト・マネージャの上司／営業にエスカレーションしたか ③顧客が、リスクを知った上で納期前倒しを決行する場合には、悪影響を最小化するよう提案したか	納期前倒しの要請を受けてから、その受理回答を行う前に、左記①～③のヒアリングを実施	
H45	リスク	価格削減要請に対して、 ①プロジェクト・マネージャは顧客に対して、リスクを伝えて、対策について折衝したか ②それでも無理があるのであれば、プロジェクト・マネージャの上司／営業にエスカレーションしたか ③顧客が、リスクを知った上で価格削減を決行する場合には、悪影響を最小化するよう提案したか	価格削減の要請を受けてから、その受理回答を行う前に、左記①～③のヒアリングを実施	
H46	リスク	受発注したシステム開発において、早期に契約を締結しているか	●RFPなど顧客からの要件と、契約内容が一致しているか確認する ●受発注から契約までの期間が長期化すると、リーガルリスクが発生する可能性があること	

	評価基準	エビデンス・確認方法	対策案
	●体制作り、システム計画、保守・運用計画に問題がないかをチェックすること ●あるいは、システム計画、保守・運用計画を作成する作業が予定されていること ●開発ルールが設定されていて、プロジェクト・メンバーに周知されていること	①システム計画書 or 保守・運用計画書 ②開発ルール ③レビュー記録票	●プロジェクト計画書を具現化した中流工程の作業計画、作業要領、規定類を作成する ●プロジェクト・メンバーに対して、開発ルールの説明会を実施する
	作成したWBSがステークホルダーが納得の上で合意されていること。例えば以下のような観点で確認を行う ●現行調査の確認方法 ●要件定義に関するユーザーとの確認方法など	①プロジェクト計画書 ②責任分担表 ③WBS	後で「聞いていない」ということにならないように、関係者を集めて日程と責任範囲について合意を得る会議を開催する
	特にバージョンが輻輳している場合、リリース・モジュールの管理などが手順化されていない場合は、リリースミスやリグレッションなどの問題が発生し、大きな手戻りになることがある	●ルール・手順書 ●リリース管理フロー	●リリース管理の徹底を行う ●リリースミスやリグレッションなどがひどい状態の場合には、まず実施手順を見直し、ミス低減を進めるとともに構成管理チームを別途作り、システム環境やソースを徹底管理する
	ファイルの移行はもちろんのこと、顧客データを移行するときには、顧客データの品質に問題があることが多々ある。顧客の保持しているデータの信憑性をよく確認して移行計画を策定することが重要である	●プロジェクト計画書 ●移行計画書	本番移行計画を立て、有識者や顧客も含めたステークホルダー間でレビューを実施する
	●体制作り、保守・運用計画に問題がないかをチェックすること ●あるいは、保守・運用計画を作成する作業が予定されていること	①保守・運用計画書 ②レビュー記録票	●保守・運用に関する体制の計画を立て、関係者と合意しておく ●作業手順・ルールがない場合は、過去の事例などを参考に整備する
	①リスクの影響やその軽減策に関し、プロジェクト・マネージャが顧客に対して説明責任を果たしたか ②プロジェクト・マネージャのエスカレーションと上司／営業の判断や行動が実際に行われたか ③顧客が、最終的に納期前倒しを決行する場合には、リスクの発生に備えた対策を提案したか	プロジェクト・マネージャ／上司・営業と顧客との折衝議事録	①プロジェクト・マネージャは納期前倒しがもたらすリスクを冷静に顧客に伝え、以降の計画について顧客と折衝する ②それでも無理があるのであれば、プロジェクト・マネージャではなく、プロジェクト・マネージャの上司自らが顧客や顧客の上司にリスクを伝え、顧客の対応を要請する ③リスクを伝えられた顧客が、それでも納期前倒しを決行する場合には、ベンダー任せで済ませず、リスクの発生に備えた、コンティンジェンシー 計画（システム 迅速復旧体制など）を自ら用意するように顧客と調整する
	①リスクの影響やその軽減策に関し、プロジェクト・マネージャが顧客に対して説明責任を果たしたか ②プロジェクト・マネージャのエスカレーションと上司／営業の判断や行動が実際に行われたか ③顧客が、最終的に価格削減を決行する場合には、リスクの発生に備えた対策を提案したか	プロジェクト・マネージャ／上司・営業と顧客との折衝議事録	①プロジェクト・マネージャは価格削減がもたらすリスクを冷静に顧客に伝え、以降の計画について顧客と折衝する ②それでも無理があるのであれば、プロジェクト・マネージャではなく、プロジェクト・マネージャの上司自らが顧客や顧客の上司にリスクを伝え、顧客の対応を要請する ③リスクを伝えられた顧客が、それでも価格削減を決行する場合には、契約上問題がなければ、開発を中止する
	●RFPなど顧客からの要件と、契約内容が一致しているか確認する ●受発注から早期に契約していること	①RFP ②契約書	法務部門を交えて、早期に契約書の妥当性を確認する

No.	知識エリア	チェック項目	個別のヒアリング要領	
H47	リスク	リスク分析を行い、リスク項目を明確にしているか	●リスク管理表により、想定されるリスクの洗い出しが行われているか ●想定されるリスクの洗出しに関して、プロジェクトに関連する有識者からの意見が取り入れられているか ●例えば、顧客との打ち合わせに出席している者でないと意味がわからないような記述になっていないか	
H48	リスク	本番運用時のコンティンジェンシー・プランは、複数の視点（開発部門、運用部門）で設定しているか	品質管理がうまく進んでいることとコンティンジェンシー・プランは別物と想定しているか	
H49	リスク	システム共同化において、システム共同化部分と自社部分のリスク項目を明確にしているか	●リスク管理表により、想定されるリスクの洗い出しが行われているか ●想定されるリスクの洗出しに関して、プロジェクトに関連する有識者からの意見が取り入れられているか ●第1ユーザー、とそれ以降のユーザを交えた会議体が設定され、進捗状況を定期的に確認できるようになっているか	
H50	顧客	RFPなどに顧客からの要件として、何を実現したいかが分かりやすく記述されているか	プロジェクトの位置付けや目的を的確に把握して、共有できることが成功の第1歩（ただし、マイグレーションする以上は、当該システムに対する新たな付加機能は発生するものと認識すべき） ●顧客企業が作成したRFPの場合、プロジェクトの定義、アウトプットの定義などが明確に記載されていること ●コンサルティング会社などが作成したRFPの場合、顧客がその内容を十分理解していること	
H51	顧客	顧客がオーナーシップを持って対応できているか	顧客自身のプロジェクトにおけるコミットメントが明確でなければ、何をすべきか判断できない ●ユーザー部門のコンセンサスを得るための活動がなされているか ●ベンダー側としては、提案や意思決定を促すための努力が行われているか ●決定事項が変更できなくなる工程上のポイントを常に明確にしながらプロジェクトを進めることにより、前提が覆ることへの対応が図られているか	
H52	人的資源	マイグレーションにおいて、ユーザー、ベンダーともに該当システムに熟知した者が投入されているか	●ユーザー、ベンダーともに現行システムのインフラ、アプリケーションに熟知した者がいること ●現行システムのドキュメントが整備されていること	
H53	人的資源	仕様追加変更対応で求められるキーパーソンを獲得できているか	①求められる経験・知識レベルを具体的に把握しているか ②業務知識のキーパーソンを確保できているかを確認する ●社内に人材がいるときは参加してもらう交渉をする ●社内に人材がいないときは、 ・協力会社から調達する ・PJT期間が長い場合は、採用／育成による人材確保 ③②を実施後も必要要員確保できない場合には、そのリスクがもたらす結果を明確にし、上司に報告したか	
H54	人的資源	新しい開発言語に対するスキルを持ったキーパーソンを獲得できているか	●対象プロジェクトのキーとなる開発言語を理解しているか ●該当技術のキーパーソンの経歴・レベルを確認する	

	評価基準	エビデンス・確認方法	対策案
	リスク管理表に以下の内容が記載されていること 想定するリスク ●リスク検知のための確認項目 ●該当のリスクの影響度 ●リスク発生防止のための対応策 ●リスク発生時の対応策とその実施タイミング	①リスク管理表	●リスク管理表上、曖昧な記述や誤解を生むような記述がないかをチェックする。そのような記述があれば、記入者を特定して、真意をヒアリングし、わかりやすく明確な記述にする ●その後、再度レビューを実施し、リスク内容をステークホルダー間で共有する
	複数の視点（開発部門、運用部門）で設定していること	①コンティンジェン・シープラン	ユーザー、ベンダーともにリスク管理を徹底する
	リスク管理表に以下の内容が記載されていること ●想定するリスク ●リスク検知のための確認項目 ●該当のリスクの影響度 ●リスク発生防止のための対応策 ●リスク発生時の対応策とその実施タイミング	①リスク管理表	●リスク管理表上、曖昧な記述や誤解を生むような記述がないかをチェックする。そのような記述があれば、記入者を特定して、真意をヒアリングし、わかりやすく明確な記述にする ●その後、再度レビューを実施し、リスク内容をステークホルダー間で共有する
	以下のような記載事項を的確に把握できているか ●開発範囲 ●契約条件 ●教育訓練 ●検収方法 ●スケジュール	①RFP	●不足分については、内容の不足分をリストアップし、顧客に調査の申し入れを行う ●現行システムの調査作業については、別途協議して再見積もりが必要であることを顧客と合意し、明文化する
	顧客側のプロジェクト推進力が信頼できるかを確認する ●顧客のプロジェクト推進に関する意思決定力が発揮されていること ●決定事項を覆すことが頻繁に起きていないこと	①ステアリングコミッティ ②体制図	●顧客内のプロジェクト・コンセンサスを確認する場を設定するよう要請する ●顧客タスクを文書化し、実行部隊のキーパーソンに徹底するよう要請する ●開発とは別体制の顧客プロジェクト・マネージャ支援契約を提案する ●上記のような対処をしても懸案事項（顧客決断で進めなくてはならないもの）の判断が期限に遅延する場合は、解決策を検討して顧客と合意を得る必要がある
	●熟知した者がいない場合は、相当の調査期間・工数を想定していること ●現行システムのドキュメントが最新に維持されていること	①体制図 ②現行システムのドキュメント	ユーザー、ベンダーともに現行システム開発当時の要員を可能な限り集結させる
	①漠然と「業務分野の経験者」と捉えずに、求められる経験・知識レベルを明確に把握していること ②キーパソンが獲得できないことによるリスクの影響について、上司と意識合わせしていること（顧客からの仕様変更を断るか否か（失敗コストと罰金／信用失墜との見合い）について判断を仰ぐこと）	①必要要員スキル・プロファイル情報 ②リスク管理簿（必要要員確保できない場合）	①要員確保のすべての選択枝を当り、最善策を明確化していること ②プロジェクト・マネージャはリスク管理簿（必要要員確保できない場合）では、プロジェクトのＱＣＤに与える具体的な影響査定結果を上司に確認して貰い、その上で措置と責任の所在を明らかにすること
	プロジェクトにとって新しい開発言語を採用する場合には、当該領域の有識者を獲得できていること	①技術者キャリアシート	●求められるレベルを具体的に把握する ●次に、キーパーソンを確保できているかを確認する ●社内に人材がいるときは、相談やチェックだけでも参加してもらう交渉をする ●社内に人材がいないときは、 ・協力会社から調達する ・理解力のある人物を起用または採用する ・人材不足も想定しコンティンジェンシー・プラン（従来の実績ある言語での製造）をステークホルダーと事前合意を得ておくこと

No.	知識エリア	チェック項目	個別のヒアリング要領	
H55	人的資源	中流工程において、下流工程の試験準備として、問題発生時の切り分け体制、処置方法、作業範囲、ユーザーとの役割分担などが確認されているか	●下流工程での問題発生時の自部門と他部署との役割分担、ミッションがはっきりしているか ●自部門の体制、役割分担が明確になっているか ●アプリケーション・プログラム担当者だけでなく、システム的見識のある担当者が参画し、問題の切り分けなどを行える体制を敷いているか	
H56	品質	【オペレーティングシステム、組み込みソフトの更改を行う場合】 中流工程において、テスト検証作業の手順、方法を計画したうえで、テストの妥当性を確認しているか	●オペレーティングシステム、組み込みソフトの更改では、不具合が発生するとアプリケーション・プログラムと比較して代替手段をとることが困難であること、また、商用システム環境ではユーザーの管理下での問題処理が困難であると認識しているか ●不具合が発生した場合に備えて、コンティンジェンシー・プランを準備しているか	
H57	品質	中流工程において、性能検証計画を明確にしたか	●テスト環境で検証できる性能要件について、障害時運用テスト、過負荷時テスト、移行テスト、運用テストを計画しているか ●テスト環境で検証できない性能要件の場合、中流工程以前に、机上で代替検証しているか	
H58	品質	中流工程において、テスト検証や方法を計画し、確実な品質確保の見通しをたてているか	障害の数の管理だけでなく質の管理、検証を実施しているか	
H59	品質	【アウトソージング受託する場合】 詳細設計レベルのＳＬＡについて、顧客と（再）確認しているか	効果、顧客満足度など総合的に判断したうえで、顧客とサービスレベルについて交渉しているか	
H60	品質	●協力会社との進捗会議や品質基準の明確化についての計画がされているか ●ステークホルダー間で合意されているか	自社にない特定ノウハウをもっている協力会社の場合、任せきりになり失敗するケースがあり、要注意	
H61	コミュニケーション	プロジェクト・マネージャと顧客とのコミュニケーション・信頼関係があるか	複数チャネルから状況をヒアリングする ●プロジェクト・マネージャ本人 ●プロジェクト・マネージャの上司 ●アカウント営業（顧客ＣＳ評価責任者）	
H62	コミュニケーション	プロジェクト・マネージャが顧客や上司／営業への報告は随時行っているか	プロジェクト・マネージャだけでは気づかない問題、措置できない問題もあり、プロジェクト状況についての中間報告を行い、随時情報を提供する必要がある	
H63	コミュニケーション	開発チーム間（協力会社間）とのコミュニケーションに問題はないか	定期会議やファイル共有などの方法を通して、コミュニケーションの促進が図られていること ●会議体が定義され、実施されているか ●情報の共有に関して、形骸的な制限などが存在しないこと	

	評価基準	エビデンス・確認方法	対策案
	●下流工程での問題発生時の役割分担が明確であること ●下流工程での問題発生時の役割に対して適切な人がアサインされていること	①問題切り分け～修正反映までのフロー図 ②問題切り分け～修正反映までの体制図／役割分担表	●取りまとめは誰が責任を持って行うのか、ステークホルダーのそれぞれの位置を明確にする ●上流の検討段階においては、作業を分担するというよりも、一体型の共同検討の形で進める方が効率的な場合もある
	運用テストや実際の運用に入ってから問題が発生しないように、事前にシステム個々の環境固有の条件／環境を反映した、十分なテストが計画されていること	①プロジェクト計画書 ②テスト計画書	●テストの計画が未計画の場合は、計画を立てる ●不具合が発生した場合に備えて、コンティンジェンシー・プラン（旧版への切り戻しなど）を準備する
	運用テストや実際の運用に入ってから性能問題が発生しないように、事前に性能要件テストが計画されていること	①プロジェクト計画書 ②テスト計画書	●性能テストの計画が未計画の場合は、計画を立てる ●性能目標値が明確でない場合は、顧客との打ち合わせを実施して明確にし、合意する ●顧客との打合せでも性能目標が明確に決まらない場合は、SIベンダー側で妥当と思われる目標を設定して、システム性能を設計する
	運用テストや実際の運用に入ってから問題が発生しないように、事前に十分なテストが計画されていること	①プロジェクト計画書 ②テスト計画書	●下流工程に入ってからでは遅く、中流工程にて事前に、テスト計画をを立てること ●品質評価では、定量目標値管理以外に、試験の質が担保されていること（データ環境、試験環境も加味し体系だった試験項目抽出方法など網羅的な試験が計画されていること）
	顧客から見える具体的運用条件について明らかにしておくこと	①契約書、SLA合意書	運用の詳細設計に基づき、顧客からみえる、具体的運用条件（MTTR、MTBFなど）について明らかにしておくこと
	中流工程にて、協力会社からの進捗報告、品質報告の方法について、合意がされていること	①プロジェクト実行計画書 ②協力会社の品質保証計画書 ③協力会社の進捗、品質報告要領	自社にない特定ノウハウをもっている協力会社でも、任せきりにせず、内製チームと同等の品質・進捗管理を行うこと
	プロジェクト・マネージャと顧客がリスクや課題の共有できているか	プロジェクト・マネージャと顧客との折衝議事録、顧客CS評価調書	上司と営業が連携し「プロジェクト・マネージャ―顧客間のコミュニケーション・ギャップ」を早期発見、措置する工夫を行う
	●顧客や上司／営業と会議体を決めて、実際に開催していること ●緊急報告体制を構築していること	プロジェクト週報、月報、マイルストーン報告	●課題やリスク、プロジェクト状況などを報告するための会議体を設定する ●緊急時は、即時連絡する
	●コミュニケーション計画を立案していること ●情報共有の仕方が明確で、それにより必要な計画書、設計書を各担当者がいつでも見られるようになっていること	①会議がどれだけの頻度で行われているかを議事録などから確認 ②（協力会社を使っている場合は、）協力会社へのヒアリング	●コミュニケーションが円滑ではないと判断した場合、適切な会議体を設置する ●担当者間の相性に問題がある場合、1対1でのコミュニケーションには問題が出やすいので、第三者（プロジェクト・マネージャなど）を交えた会議体を定例化する ●協力会社から、さらに一部の作業を別会社に発注している場合、重要事項が実際の作業者まで正しく伝達されないなど、コミュニケーションが滞りがちになる。このような場合は、階層構造をフラットにし、各社からのキーパーソンを集めた会議体を作る

No.	知識エリア	チェック項目	個別のヒアリング要領	
H64	コミュニケーション	作業内容と終了基準についてのメンバーの認識が曖昧ではないか	曖昧さがあると作業の進捗が把握できなかったり、把握したと思っていてもメンバーにより認識の内容が異なっていたりし、プロジェクトとしての統一した進捗把握ができない。品質にも問題が出ることがある。プロジェクト・マネージャの認識が重要	
H65	コミュニケーション	顧客とのレビュー議事録などがあり、承認印をもらうなどして合意(承認)を得ているか	●顧客とのレビュー計画書は作成しているか ●レビュー実施要領は作成できているか ●実施要領などの中で確認方法を取り決めているか ●議事録またはメールでも記録として残しているか	
H66	コミュニケーション	重要な指摘・変更が発生した場合に、その内容が全員に周知徹底されているか	●1つの指摘、変更が他の部分にも影響する可能性がある ●重要な情報についてはメンバーに漏れなくタイムリーに伝達する方法が確立していること	
H67	コミュニケーション	スケジュール・作業内容について顧客との認識(作業レベルの意識)にズレがないか	期限のほか、作業内容、成果物の記述レベルなどについて、具体的な資料を用いて調整しているか	
H68	モチベーション	プロジェクト・メンバーに、このプロジェクトでの個人成長目標を持たせているか	プロジェクトを通して、個人の成長が期待できるということが、話し合われていることが重要。各メンバーが成長するための目標を持っているかどうかは、モチベーションに大きく影響する	
H69	モチベーション	プロジェクトの問題点や作業環境について、メンバー内に不満、要望がでてきていないか	より良い作業環境を望む前向きな意見か、プロジェクトに対する不信感かを区別する。メンバーの要望がプロジェクトの運営に反映できていることが重要	
H70	モチベーション	社員の労務状況(労務時間、作業環境)は悪くないか	遅刻が多い、さっさと帰る、仕事が無いのに残っているなどの問題がある場合には、原因分析を行い、適切な対応策(増員や人材配置の見直しなど)を実施しているか	
H71	モチベーション	人員の離脱が多くないか(心身における体調不良、一身上の都合など)	離脱が多い場合、その原因がプロジェクトへの不信感・危機感ではないか	
H72	基本動作	議事録のほか、各種ドキュメントが正しい日本語で書かれているか	●文書化した内容によって、ステークホルダー間に齟齬を生まないような記述ができているか ●正しい日本語での記述は、プロジェクトの状況を把握する上での前提事項となる。特に議事録などを読む相手のことを考えて、正しい日本語で記述できているかどうかが重要である	
H73	基本動作	プロジェクト内で使われる用語の意味について、顧客およびプロジェクト・メンバーの認識にずれがないようにしているか	使われている用語の意味の取り違えにより設計ミス、作業ミスを起こさない対策をとっているか	

	評価基準	エビデンス・確認方法	対策案
	例えば、詳細設計はどこまでやったら終了か、明確にしているか ●作成すべきドキュメントの完成を宣言できる条件が分かっていること ●プロジェクト・メンバーに終了基準について説明がなされていること ●無計画な検討スケジュールにしていないこと	①作業内容の終了基準を確認	認識が曖昧になっている場合は、どのような状態になれば終了宣言してよいかの基準を明確にし、作業者に伝える。そのためには、次のことを行う必要がある ●作業進捗状況を定量的に測定するための方法を考え、定量的な進捗報告を徹底する ●成果物の作成品質レベルを評価し、作業完了を判断できるキーパーソンを明確にする
	顧客とのやり取りは文書化して記録を残していること ●顧客との各種レビューを実施 ●議事録を作成 ●その議事録について合意を得る	①レビュー計画書 ②レビュー実施要領 ③議事録	●レビュー議事録を作っていない場合は、作成する ●報告会議などで議事録のレビューを慣例化する
	仕様変更、レビュー結果、障害情報などの情報共有の仕組み・やり方が決められていること	①議事録 ②周知徹底の記録（メールなど）	顧客レビューなどによる重要な指摘事項を周知徹底させることを義務付ける
	スケジュール・作業内容について合意を得ていること	①議事録 ②スケジュール表 ③成果物サンプル（記述フォームなど）	●スケジュールの妥当性についてまずは内部で確認・合意し、その後顧客とともにレビューする ●問題がある場合はスケジュールを見直す
	プロジェクト・メンバーと個人成長目標について話し合いができていること	①要員育成目標	プロジェクトとして各メンバーの育成目標を定めて、各メンバーとその内容について合意する
	●プロジェクト・メンバーとプロジェクトの問題点や作業環境などについてコミュニケーションができていること ●メンバーの不満、要望に、適切に対処していること	①メール ②ヒアリング	プロジェクト・マネージャを補佐するスタッフをつけて管理を強化しつつ、不信感を表している人それぞれとコミュニケーションをとる
	労務時間などをもとに判断するとともに、現場に残っているかどうかを目視で確認すること	①勤務管理表 ②職場見学	●労務状況をよくするために、増員、人材配置の見直し、作業環境の改善などを実施する ●労務状況の改善が難しい場合、コミュニケーションを頻繁にとりメンバーとの一体感を作るようにする
	休みや離職の傾向を確認していること	①体制表	●離脱原因がプロジェクトへの不信感・危機感である場合は、プロジェクトの区切り（終わり）の時期について明言してあげることで改善したり、プロジェクト推進に関してプロジェクト一体感や仲間意識を醸成することで、離脱を抑えることができる場合がある ●また、プロジェクト・マネージャを補佐するスタッフをつけて体制を強化する方法も有効である ●さらに、定常的に付き合う協力会社などを作り、離脱が食い止められない場合の要員補充を容易にできるようにすると同時に、キーパーソンの状況を重点的に確認し離脱されないようにしておく
	各種ドキュメントの内容を確認していること	①議事録	議事録や障害票の日本語に問題がある場合は、教育的指導を実施する。また、悪い事例の修正例を公開・情報共有して、注意を促す
	用語集などを作成し、関係者間で共有できていること	①用語集	●用語集一覧を作って顧客およびプロジェクト・メンバーとレビューを実施する ●顧客と長い付き合いがある場合は、必ずしも用語集が必要ではないが、顧客側用語を使うかベンダー側用語を使うか決めておくことは必要である

No.	知識エリア	チェック項目	個別のヒアリング要領	
H74	基本動作	プロジェクトのメンバー内に話しづらい雰囲気があって報告が上がってこないという事象はないか	課題管理や進捗管理など、プロジェクト・マネジメント上の問題が報告されていないことはないか	
H75	基本動作	非公式の場で上下間のコミュニケーションが図られているか	会議の場以外で情報共有を行うことで、通常では表面化しない問題に対応できることがある。特に、メンバーのマインドの問題には有効	
H76	課題管理	課題管理を実施しており、それらの課題について、顧客との合意が取れているか	●課題管理について、顧客との合意ができているか ●顧客側で課題と認識しているのに、開発サイドでは認識していない場合は将来、大きな問題に発展することがある ●課題として共通認識になっていても、重要度や緊急度の認識のずれも問題になる場合がある	
H77	課題管理	管理されている課題の対策予定日が過ぎたものに対して、対処の強化を実施しているか	●対策予定日が大幅に過ぎているものはないか。対策予定日が大幅に過ぎているものは以下のようなプロジェクトの推進に関して問題を含んでいる可能性がある ・担当者がアサインされていない ・担当者の認識がない ・課題が大きすぎて解決策が具体的なところまで落とし込めないまま放置されている ●課題管理の対象にリスクを混入させないこと ●課題の平均解決日数などを活用すると、課題の大きさや意志決定の早さを評価できる	
H78	課題管理	複数の課題解決策に対し、明確な判断基準（メリット／デメリット）のもとに優先順位をつけた上で、解決策を選択しているか	●経験や勘だけに頼った判断をしていないか ●経験や勘だけに頼った判断は大きな判断ミスにつながる。判断には根拠や基準が必要である ●判断の材料となる選択肢についても、十分に検討し、よりよい判断であるかを常に意識しているかをチェックする	

	評価基準	エビデンス・確認方法	対策案
	●進捗、課題管理の定例会議などでのメンバーの発言量が少なくないこと ●メンバーが他のメンバーの発言に対して、活発に意見を出し合っていること	●議事録 ●ヒアリング	●プロジェクト内の雰囲気作りを推進する ●上下の人間関係に問題がある場合には、メンバー変更など体制の問題として取り上げる
	会議以外の場でも報告、相談が行われており、共有できていること	●ヒアリング ●現場確認	プロジェクト内の雰囲気作りを推進する
	顧客と課題管理表のレビューを実施し、その内容および改訂事項について合意を得ていること	①課題管理表	●是正処置を実施していない原因を突き止める ●特に課題内容が大きすぎて、誰が着手するべきか判断ができない場合は、キーパーソンを集めて対応策を検討していく必要がある ●大きな課題か、なかなか合意が取れない場合は、エスカレーションして対策を打つ
	●課題管理表に完了予定日を記入していること ●期限が過ぎた課題に対して、優先順位を上げて対応していること	①課題管理表	●是正処置を実施していない原因を突き止める ●課題が多すぎて対応できていない場合には、優先順位づけしたり、課題内容を一度対策可能なものに置き換える ●特に課題内容が大きすぎて、誰が着手すべきか判断ができない場合はキーパーソンを集めて対応策を検討していく必要がある ●大きな課題か、なかなか合意が取れない場合は、エスカレーションして対策を打つ
	●解決策を複数考え、論理的な比較を行ってより適切な解決策を選択するようにしていること ●解決策が単なる思い込みになっていないこと	①課題管理表	●判断の根拠を明確にした上で判断する。判断の根拠、優先順位を課題管理表に記録として残す ●しかし、明確な判断ができない場合には、スキルを持ったメンバーやプロジェクト外の技術者などとレビューを実施して、解決策を選択する

付録 3. プロジェクトにおける問題事象と対策の事例集

1 仕様変更による「品質問題」

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。

事例における見切りの内容

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。

本来の判断の考え方

・変更計画でも、プロジェクト計画ベースライン見積りについて、初期計画と同じ精度を確保しない限り、リスクも正確にあぶりだされないと判断するべきだった。

対処例

SIベンダー

・まずPMは緊急機能追加に見合う、プロジェクト負荷軽減策（既計画機能の納期変更、緊急機能の段階リリースなど）について顧客と折衝する。
・それでも無理があるのであれば、PMではなく、PMの上司自らが顧客や顧客の上司にリスクを伝え、顧客の対応を要請する。

顧客

・リスクを伝えられた顧客が、それでも仕様変更を決行する場合には、ベンダー任せで済ませず、リスクの発生に備えた、コンティンジェンシー・プラン（システム迅速復旧体制など）を自ら用意する。

顧客とPM/上司/営業の関係を「発注元―業者」の関係ではなく、プロジェクトリスクに関しては、双方が「リスクと闘うパートナー」となるような、リレーションシップを受注～上流工程前に構築しておくこと。

2 仕様変更による「予算超過」

顧客は自ら選定したパッケージ・ソフトウェア導入の前提で予算を組んだ。しかし、検討詳細化に伴い五月雨式に多量のカスタマイズを続けた。当初の費用対効果が得られないほど予算超過となり、仕様変更をストップしシステムが完成。エンドユーザー部門でほとんど使われないシステムとなった。2次開発以降の計画が中止となり、受託開発会社は事業の柱を喪失した。

事例における見切りの内容

明らかに仕様変更によるプロジェクト納期、品質上の問題がなく、追加費用を顧客の担当者が保証してくれたので、発注元の言いなりで仕様変更に応じた。

本来の判断の考え方

・顧客側が主たる責任を負うべき問題だとしても、仕様追加の必要性も確認するべきであった。

対処例

SIベンダー

・上流工程パッケージ・ソフトウェア導入におけるFit&Gap分析の徹底提案。
・仕様追加に対する顧客事業上の必然性の検証。

顧客

・上流工程でのパッケージ・ソフトウェア導入意思決定段階におけるFit&Gap分析の徹底実施。
・パッケージ・ソフトウェア導入の場合、多数の仕様追加も含めた全体での費用対効果の一括検証。

顧客とPM/上司/営業の関係を「発注元―業者」の関係ではなく、双方が「ビジネス・パートナー」となるような、リレーションシップを受注～上流工程にてあらかじめ構築しておくこと。

3 仕様変更による「納期遅延」

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。

事例における見切りの内容

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。

本来の判断の考え方

・PMは上司がどのような性格・状態でも、リスクを正しく説明する責任がある。
・上司/営業は「PM―顧客間」「上司/営業―PM間」のコミュニケーションGapがリスクとなりうると考えるべき。

対処例

SIベンダー
・上流工程にて上司/営業は顧客に合わせたPMのアサインを行う。
・上司はPMとの風通しのよいコミュニケーション環境を構築し、中流工程でも維持する。上司/営業は「PM―顧客間のコミュニケーション・ギャップ」を早期発見し、是正する。

顧客
―

顧客とPM/上司/営業の関係を「発注元―業者間のリスク転嫁の応酬」の関係ではなく、双方が「ビジネス・パートナー」となるような、リレーションシップを受注～上流工程にてあらかじめ構築しておくこと。

4 仕様変更による「システム老化」

五月雨式に仕様・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。

事例における見切りの内容

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る

本来の判断の考え方

・五月雨式な仕様変更が避けられないのであれば、システム老化の責任の所在について明らかにしておく。

対処例

SIベンダー
・上流工程で追加仕様の凍結を五月雨でなく、まとまった単位で一括折衝し決めるよう顧客に提案。
・上流工程で顧客の事業変化要件に合うシステム設計法を提案。

顧客
・上流工程で自社事業環境変化の早期特定と、変化に強いシステム構造、一括仕様変更の検討を主導。

顧客とPM/上司/営業の関係を「発注元―業者間」の関係ではなく、双方が「ビジネス・パートナー」となるような、リレーションシップを受注～上流工程にてあらかじめ構築しておくこと。

5 営業が中流工程以降身を引き仕様変更で「不採算」に

厳しい受注ノルマを課せられる営業は契約が終わると、中流工程以降は、より高い受注が見込める他顧客・他システムの案件を優先し、受注済みプロジェクトからは身を引いた。その後、大量の仕様変更要求が発生し、PMは契約で決めた納期と品質の縛りから防戦したが、費用折衝に関しては営業に比べ顧客の文化に疎く、力もないPMが孤軍奮闘した結果、不採算プロジェクトとなった。またPMの価格交渉負荷が予定よりも大きくなった分、納期・品質でも問題が発生した。

事例における見切りの内容

営業は受注責任と役割分担を見切っていた。営業の目標は受注額であり、受注後はPMが予定原価率を維持する責任者とすれば、問題ないはずだと見切っていた。

本来の判断の考え方

・営業は受注時だけでなく、受注後、プロジェクトが完了するまで価格折衝責任を担うべきとする。

対処例

SIベンダー

・SIベンダーとして、PMの負荷や能力限界も踏まえ、中流工程での価格交渉責任者を明確にする。
・SIベンダーとして営業が受注後もプロジェクトフォローを行う仕組みを作る（例：着地利益率を営業の業績評価として縛るなど）。

—

顧客

—

6 仕様変更の連続で出口が見えないプロジェクト、メンバーが疲弊

顧客は事業環境変化への即応を企業文化としており、仕様・納期の確定後、仕様の追加・削除が次々と発生した。顧客はその分の費用を約束し、納期変更提案も了解してくれたので、すべての変更を受け付けた。出口が見えないプロジェクト・メンバーのモチベーションが下がり、休職・離職者が多発。

事例における見切りの内容

PM側が顧客文化を理解しておらず、仕様変更はあまり無いものと見切って、プロジェクトのScope, 納期をメンバーに説明していた。

本来の判断の考え方

・ゴール（仕様・納期）変更ごとにPMは、メンバーとゴール変更のコンセンサスを取るべきだった。

対処例

SIベンダー

・変更が多い前提で、メンバーからみたプロジェクトのゴール、メンバーの交代計画をプロジェクト計画の中で伝える。
・無理があれば、五月雨式の変更要求をまとめ、顧客に一括変更を行う提案を行う。

—

顧客

—

7 変化即応型の開発をScope変更ルールが無く請負った

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、発注側ー受注側で「言った/言わない」の応酬となった。

事例における見切りの内容

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。

本来の判断の考え方

・双方の将来リスク低減のためにも、変更追加が生じた責任について、前向きな議論をしてから作業着手する。

対処例

SIベンダー
・仕様変更発生時、CCBに従った調整のメリットを顧客の目線で顧客に提案し理解を得る。
・これで、顧客との信頼関係が悪化するようであれば、他にその本質的な原因があると考え、真原因を分析する。

顧客
要件変更に対してCCBを尊重した調整を実行。

顧客とPM/上司/営業の関係を「発注元―業者」の関係ではなく、プロジェクトリスクに関しては、双方が「リスクと闘うパートナー」となるような、リレーションシップを受注～上流工程前に構築しておくこと。その証として、仕様変更ルール/変更管理会議（CCB）設置を双方で合意しておく。

8 変更要求問題で、PMの交渉力、報連相が不足

PM不足の環境で、プロジェクトを任された新任PMが顧客からの信頼を得ようと、変更要求をほとんどすべて受け入れた。これが下流工程でより大きなＱＣＤ問題に発展したが、上司にも相談しようとも考えなかった。

事例における見切りの内容

仕様変更に柔軟に対応することが、顧客からの信頼を得ることにつながると考えた。

本来の判断の考え方

・仕様変更に対するリスクの影響を査定し、顧客、上司などステークホルダーと共有する。
・交渉力の限界では上司にエスカレーションする。

対処例

SIベンダー
上司はPMとの風通しのよいコミュニケーション環境を構築し、中流工程以降でも維持する。上司/営業は「PM―顧客間のコミュニケーション・ギャップ」を早期発見し、是正する。（特に新任PMに対する公式/非公式のコミュニケーションの風通しを良くし、サポートを強化）

顧客
―

―

9 開発中のリスクについてPMが上司にエスカレーションをあきらめた

PMが仕様変更折衝が難航した。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PMを是とし、問題提起報告で同時に解決策が書けないPMを能力不足と評価する傾向があった。PM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。

事例における見切りの内容

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。

本来の判断の考え方

・自分の身の丈を超える問題は、上司のいかんにかかわらず、必ず報告する。

対処例

SIベンダー

・上司はPMとの風通しのよいコミュニケーション環境を構築し、中流工程以降でも維持する。
・上司/営業は「PM―顧客間のコミュニケーション・ギャップ」を早期発見し、是正する。

―

顧客

―

10 PM・顧客間不信頼

前のプロジェクトでは顧客とうまくやってプロジェクトを成功させたPMだが、新規顧客から求められるドキュメントが詳細化し量も増加。PMが過負荷となったところで、中流工程での仕様変更が発生。交渉がギクシャクして難航し、それが工期をさらに厳しくさせ、品質問題も誘発した。顧客は、問題の本質はPMが説明責任を果たせなかったことだと主張した。

事例における見切りの内容

PMは前のプロジェクトでは顧客とうまくやってきたので、これまでの顧客とあうんの呼吸で行ってきたPM流儀がベストであると考え、新しい顧客でもうまくいくと見切った。

本来の判断の考え方

・PMは、これまでの行ってきた流儀（仕様確認、品質説明の詳細度など）を新しい顧客に事前に説明する。顧客の独自の価値観・文化も聞き出した上で、理に適った考えは取り入れる。

対処例

SIベンダー

・初期段階で、上司/営業は顧客の価値観・文化に合わせたPMのアサインを行う。
・上司はPMとの風通しのよいコミュニケーション環境を構築し、中流工程でも維持する。上司/営業は「PM―顧客間のコミュニケーション・ギャップ」を早期発見し、是正する。

―

顧客

―

11 仕様変更を一切受付けないことによる問題

顧客の事業要請上必要な仕様変更であった。確実にシステム開発をやり遂げるが、やや頑固なPMが、できない、できないの一点張りで拒否をし続けた。顧客はPMと長期間折衝の末、PMの上司にPMの交替を要請。新しいPMは仕様変更を実施し、特に品質上問題となることは無かったが、交渉に要した時間だけ顧客システムのサービス開始時期が遅れてしまった。

事例における見切りの内容

旧PMは仕様変更を1件でも受け付けると、一事が万事となると見切った。

本来の判断の考え方

・PMは変更による影響やリスクの客観的な根拠で顧客に説明する。
・PMはビジネスマンとして、顧客要望の必要性を理解し、変更の実施・否について、双方が満足のいく合意点を探る。

対処例

SIベンダー
・PMは変更の影響・リスクを顧客に説明した上で、ビジネスマンとして、双方が満足の行く合意点を探る。
・上司/営業はPMのビジネスマンとしての対顧客性に注目し、問題があればPM交替などの対策を先手で実施する。

顧客
―

顧客とPM/上司/営業の関係を「発注元―業者」の関係ではなく、双方が「ビジネス・パートナー」となるような、リレーションシップを受注～上流工程前に構築しておくこと。

12 複雑な顧客ステークホルダー組織で仕変が多発

ステークホルダーが多くステアリング会議が機能していない状態で仕様追加・変更案件が多発したが、意思決定が遅れ、工期圧迫により、納期・品質問題が発生した。

事例における見切りの内容

ステークホルダーが多かったので仕様凍結までは、ステアリング会議体制があったが、凍結後は余り必要ないと見切った。

本来の判断の考え方

・ステアリング会議は中流工程でも継続すべき。

対処例

SIベンダー
中流工程でのステアリング会議を継続する提案をした。

顧客
中流工程でのステアリング会議を継続した。

中流工程でのステアリング会議の継続。

13 仕様凍結確認せず中流工程に着手

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。

事例における見切りの内容

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。

本来の判断の考え方

- 先行着手できる範囲は、この先どのような仕様変更があっても独立性が高い担保のあるサブシステム部分に限定すべき。
- 基本部分、他のインターフェースが多い部分のようなサブシステムは、先行着手しない。仕様凍結の遅れ、未実施がもたらすリスクを冷静に顧客に伝えるべき。

対処例

SIベンダー
- PMは、費用の制約、納期（段階提供）の制約の下、リスクと凍結Scopeとのトレードオフの議論と意思決定を受けてから着手することを顧客に提案し、仕様凍結時期について折衝する。
- それでも無理があるのであれば、PMではなく、PMの上司自らが顧客や顧客の上司にリスクを伝え、顧客の対応を要請する。

顧客
- リスクを伝えられた顧客が、それでも当初計画を決行する場合には、ベンダー任せで済ませず、リスクの発生に備えた、コンティンジェンシー・プラン（システム迅速復旧体制など）を自ら用意する。

顧客とPM/上司/営業の関係を「発注元―業者」の関係ではなく、プロジェクトリスクに関しては、双方が「リスクと闘うパートナー」となるような、リレーションシップを受注～上流工程前に構築しておくこと。

14 マイグレーションの要件、実質はシステムレベルアップ。仕様追加極大

受注条件は、現行システムのマイグレーションであったが、実態は、性能要件、機能要件の大幅追加であった。現行システムも自社製品であることから、ベンダーが無理難題を引き受けることになった。

事例における見切りの内容

自社テリトリ維持のため、また、顧客満足度向上との認識から、ベンダー負担やむなしと判断した。

本来の判断の考え方

- 過去、ビジネス上の関係が深いとはいえ、本ビジネスを客観的に分析し、受発注すべき。

対処例

SIベンダー
自社テリトリ・ユーザーに束縛されず、冷静に分析する。必要により受注しないことも視野に入れた行動が必要である。

顧客
発注要件を明確にすべき。一見、発注元に有利な案件のように見えるが、受注元に丸投げでは、プロジェクトは失敗する。

過去、ビジネス上の関係が深いとはいえ、本ビジネスを客観的に分析し、受発注すべき。

15 世代交代で顧客、SIベンダーともに熟知した要員不在。マイグレーション仕様固まらず

マイグレーション案件を受注するも、ユーザー、ベンダーともに該当システムに熟知した者がいない。また、ドキュメントもメンテナンスされていない。ユーザー、ベンダーともに相手に依存したままシステム構築に入るが、システム構築容易と判断した上での契約と実態が大幅に乖離。納期大幅遅れ。

事例における見切りの内容

ユーザー、ベンダーともに内容熟知していないために、マイグレーション容易と判断した。

本来の判断の考え方

・システム運用ができていることとシステムの内容熟知とは別と判断すべきであったが、その判断に甘さがあった。また、システム構築をインフラに注力し、アプリケーションを軽視した。

対処例

SIベンダー
営業優先せず、実現性の確実な検証を実施する。

顧客
システム化要件をマイグレーションにこだわらず、再構築を視野に入れ検討する。

・ユーザー、ベンダーともに現行システム開発当時の要員を可能な限り集結し、チェック、レビューを実施すべき。
・受注時、及びシステム開発着手時に第三者によるチェック、レビューを実施すべき。

16 正式要求仕様書無く作業。仕様あいまいにより変更多発

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。

事例における見切りの内容

後付けで要求仕様書を作成し、再度、要求仕様との整合をとった。

本来の判断の考え方

・たとえ短納期といえども、開発プロセスは厳守すべき。

対処例

SIベンダー
あいまいな開発プロセスが、結果的にユーザー、ベンダーともに悪影響を被るということを認識すべき。

顧客
発注元の認識として、ベンダー（受注元）が要求仕様について分かっているとの過大認識は厳禁である。特に、継続開発の場合、要注意である。

短納期という事情をSIベンダー、顧客が認識し要件確定、評価、検証を両者で実施する。

17 当初納期の大幅前倒し要求を受け付けQCD問題に

基幹システムで仕様は決まっていたが、顧客より予定納期が大幅に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。

事例における見切りの内容

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。

本来の判断の考え方

・単純な納期前倒しがもたらすリスクを冷静に顧客に説明する責任を果たすべき。
・その上で、納期前倒しを実現するための方法論（Scope制約、コスト制約などの条件）について、提案し折衝すべきであった。

対処例

SIベンダー
・PMは、納期前倒しを実現するための方法論（Scope制約、コスト制約などの条件）を提案し、この方法の下でのリスクと納期前倒しとのトレードオフの議論と意思決定を顧客に提案し、プロジェクト計画の修正を提案する。
・それでも提案が受けれられないのであれば、PMではなく、PMの上司自らが顧客や顧客の上司にリスクを伝え、顧客の対応を要請する。

顧客
・リスクを伝えられた顧客が、それでも納期前倒しを決行する場合には、ベンダー任せで済ませず、リスクの発生に備えた、コンティンジェンシー・プラン（システム迅速復旧体制など）を自ら用意する。

顧客とPM/上司/営業の関係を「発注元―業者」の関係ではなく、プロジェクトリスクに関しては、双方が「リスクと闘うパートナー」となるような、リレーションシップを受注～上流工程前に構築しておくこと。

18 当初価格の大幅削減要求を受け付けQCD問題に

仕様が決まり仮発注も受けとっていたが、顧客が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Scope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。

事例における見切りの内容

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。

本来の判断の考え方

・PMは上司にプロジェクトの要員稼働削減時のリスクを説明し、顧客への提案内容（会社の利益限界から導かれる限界の削減要員体制）にについて決めて貰うべきだった。
・PMは営業とともに顧客に対して価格（要員）削減がもたらすリスクについての冷静な説明を行う責任も果たすべきだった。

対処例

SIベンダー
・PMは、費用削減を実現する方法論（要員削減の条件）を提案し、リスクと費用削減とのトレードオフの議論と意思決定を顧客に提案する。
・それでも提案が受けれられないのであれば、PMではなく、PMの上司自らが顧客や顧客の上司にリスクを伝え、顧客の対応を要請する。

顧客
・リスクを伝えられた顧客が、それでも費用削減を決行する場合には、ベンダー任せで済ませず、リスクの発生に備えた、コンティンジェンシー・プラン（システム迅速復旧体制など）を自ら用意する。

顧客とPM/上司/営業の関係を「発注元―業者」の関係ではなく、プロジェクトリスクに関しては、双方が「リスクと闘うパートナー」となるような、リレーションシップを受注～上流工程前に構築しておくこと。

19 過剰な責任所在文化による契約締結遅れ、物事のあいまいさ

システム開発を入札で受注した。要求が概括的なものであり、早期に契約を締結し機能範囲を確定しようとするも、ユーザー内部での仕様調整が進まず、また、決定内容への責任回避のため、契約進展せず。その間、度重なる仕様追加、変更が発生し、開発作業が混乱した。

事例における見切りの内容

新規追加、仕様変更の扱いを、当初計画した全作業量とのトレードをそのつど交渉しながら、開発実行。
（プロジェクト・マネージャ疲弊）

本来の判断の考え方

・入札といえども、受発注後、早期に契約を締結し、作業範囲を認定すべき。

対処例

SIベンダー
ユーザー事情を念頭におき、総工数が増えない様、常にコミュニケーションをとる。

顧客
仕様決定に対する責任の重さとともに、システム完成度に対する責任の重要性も認識し、プロジェクトを推進するべき。

SIベンダー、顧客両者で入札、応札条件を厳守する。

20 顧客責任部門、責任者不明

ユーザーで、システム構築責任部門と、費用管理部門が異なるため、機能追加、仕様変更に伴う費用と実現可否が不整合となった。調整工数とスケジュールキープが大変であった。

事例における見切りの内容

費用未調整の増額分については、ベンダーが後日増額交渉するという未解決のまま作業着手した。

本来の判断の考え方

・窓口を1本化し、ユーザー、ベンダーそれぞれ内部調整すべき。

対処例

SIベンダー
ユーザーに窓口一本化と、ユーザー内調整を強く申し入れ、契約時の体制に組み込むべき。

顧客
作業フェーズ（契約、システム構築）で責任窓口を明確にする。

窓口を1本化し、ユーザー、ベンダーそれぞれ内部調整すべき。

21 変えられない既存業務プロセスにパッケージを導入

法人にパッケージを前提にシステムを導入を図るも、同法人では他ライバル法人でパッケージ導入成功例を過剰に意識していた。スクラッチ開発では同法人内の決済が通らないことから入札時は、パッケージ導入とし、それに合わせた安価とした。実態は、法人の業務の独自性から、パッケージの活用に制約があり、結局、カスタマイズ量増加により仕様確定が遅れプロジェクトが混乱した。

事例における見切りの内容

カスタマイズ量の低減に努力するも、結局ベンダー持ち出しでカスタマイズを実施した。

本来の判断の考え方

・パッケージの導入に当たっては、機能、性能、障害、運用、拡張性、制約など実運用を見据えて検証するフェーズを設定する。

対処例

SIベンダー
パッケージ拡販に当たっては、機能（広義）とともに、制約、条件を明確にする。

顧客
パッケージ利用に当たっては、想定している利用形態をシミュレーションし、パッケージの有効性を検証する。そうでないと、カスタマイズが大きくなり、品質面で痛手を被る。

パッケージの導入に当たっては、機能、性能、障害、運用、拡張性、制約など実運用を見据えて検証するフェーズを設定する。

22 システム共同化、再構築で自社独自のテスト不十分。重大障害発生

システム再構築に当たり、システム共同化を選択した。しかし、第一ユーザー向けの開発が遅延し、第二以降ユーザーのテストが不足していた。この結果、重大障害発生した。

事例における見切りの内容

要員増強し、第一ユーザー、第二以降ユーザー両方の対応を実施した。

本来の判断の考え方

・ベンダー、第二以降ユーザーは、テスト実施状況を報告、把握し、あらかじめテスト計画に盛り込む。

対処例

SIベンダー
・システム共同化は、問題発生時、単独システム構築より問題が大きくなるため、リスク管理を十分に行うこと。
・また、早期に第二以降ユーザーにテスト実施状況を報告する。

顧客
・システム共同化は、問題発生時、単独システム構築より問題が大きくなるため、リスク管理を十分に行うこと。
・また、早期に第一ユーザーの状況を把握し、自行のテスト計画に反映させる。

ベンダー、第二以降ユーザーは、テスト実施状況を報告、把握し、あらかじめテスト計画に盛り込む。

23 システム共同化を目指すが、先行ユーザーのサービスインの遅れで納期遅延

システム再構築に当たり、システム共同化を選択。しかし、第一ユーザー向けの開発が遅れ、それに伴い第二以降ユーザーのサービスイン連続遅延。

事例における見切りの内容

移行遅延に伴う費用を支払いサービスイン遅延。また、導入計画（全体・部分）の見直し。

本来の判断の考え方

・第二以降ユーザーのサービスインについては、第一ユーザーのサービス・インを待って決定する旨、あらかじめ契約に盛り込み、計画の見直しを実施。

対処例

SIベンダー
安易にシステム共同化にとらわれ、楽観視せず、システム共同化ゆえのリスクを常に把握する。
（例:要員のアサイン、機能追加、変更など）

顧客
安易にシステム共同化にとらわれ、楽観視せず、システム共同化ゆえのリスクを常に把握する。
（例:要員のアサイン、機能追加、変更など）

第二以降ユーザーのサービスインについては、第一ユーザーのサービスインを待って決定する旨、あらかじめ契約に盛り込み、計画の見直しを実施する。また、システム共同化に当たって、ベンダー主導にならないよう顧客も進捗管理、及びリスク管理を実施する。

24 ハードウェア、OS更改でテスト必要認識不足。重大障害発生

オペレーティングシステムのレベルアップに伴うテスト実施計画で、ユーザーのシステム・インフラに対する重要性認識不足から、テスト不足となり、本番運用後、重大障害発生。

事例における見切りの内容

ベンダー側要員総動員し、休日、深夜に再検証。

本来の判断の考え方

・オペレーティングシステム、組み込みソフトといえども、各システムごと、利用条件が変わる。システム・インフラに不具合があると、運用上代替手段が取り難いことを認識し、検証を十分行うべき。

対処例

SIベンダー
他ユーザー向け導入状況を把握し、検証を徹底する。

顧客
・オペレーティングシステム、組込みソフトの更改に不具合が発生すると、ユーザーの管理下での問題処理が難しいと認識し、徹底した検証をする。
・ベンダーに任せ切りにしない。
・コンティンジェンシー・プランの設定をする。

オペレーティングシステム、組み込みソフトといえども、各システムごと、利用条件が変わる。システム・インフラに不具合があると、運用上代替手段が取り難いことを共通の認識として持ち、十分な検証を実施する。

25 スケジュールが遅れ、業務主体のテストとなり、基盤・運用系不安定

テスト計画が押せ押せになり、業務主体のテスト実施となった。その結果、障害時運用テスト、過負荷時テスト、移行テスト、運用テストなどが不足し、本番運用以降安定せず。

事例における見切りの内容

ベンダー担当者がシステム運用（操作）し、当面切り抜けた。

本来の判断の考え方

・ユーザー、ベンダー両責任者が進捗状況を客観的に捉え、評価環境の増設、要員のアサインなど先を見据えた対策を取るマネジメントが必要。

対処例

SIベンダー
通常、スケジュール遅れの場合、業務テストが優先されることは当然。早い段階から、シュミレータ、疑似アプレリケーション・プログラム、データなどを準備し、計画に組込む。

顧客
システム全体を見据えたプロジェクト管理をする。

ユーザー、ベンダー両責任者が進捗状況を客観的に捉える。
評価環境の増設、要員のアサインなど先を見据えた対策を取る。

26 障害の数の定量的管理にとらわれ過ぎ、品質を誤評価

バグ収束曲線などを検証しながら、品質管理を行っている。サービス・イン直近の滑り込みテストで未解決バグを吸収したが、障害の数の定量的管理にとらわれ過ぎ、バグの内容とテスト過不足の検証が行われなかったため、バグが内在し問題発生。

事例における見切りの内容

ベンダー、ユーザー総動員でシステム再検証。

本来の判断の考え方

・障害の数の管理だけでなく、質の管理、検証を併せて実施する。

対処例

SIベンダー
障害の数の管理だけでなく、質の管理、検証を併せて実施する。

顧客
障害の数の管理だけでなく、質の管理、検証を併せて実施する。

障害の数の管理だけでなく、万一システム運用に影響を与えることを考慮し、質の管理、検証を併せて実施する。

27 システム統合の場合、企業文化の相違によるコンティンジェンシー・プラン不備

本番運用時のコンティンジェンシー・プランに対する意識が異なる。特に開発部門の論理的にプロジェクトを進めているから大きな問題が発生しないだろうという考えと、運用部門の問題が発生したらどう処置するのかという様な意識の違いにより、本番運用前のコンティンジェンシー・プランの質に過不足発生。

事例における見切りの内容

場当たり的な問題処理でしのいだ。

本来の判断の考え方

・ユーザー、ベンダーともにリスク管理を徹底し、たとえ品質管理がうまく進んでいてもコンティンジェンシー・プランの徹底を図る。

対処例

SIベンダー
リスク管理を徹底し、たとえ品質管理がうまく進んでいてもコンティンジェンシー・プランの徹底を図る。

顧客
リスク管理を徹底し、たとえ品質管理がうまく進んでいてもコンティンジェンシー・プランの徹底を図る。

ユーザー、ベンダーともにリスク管理を徹底し、たとえ品質管理がうまく進んでいても、サービス・イン時、仮に問題が発生しても運用でカバーできるよう、コンティンジェンシー・プランの作成、徹底を図る。

28 外部委託のSLA不備

システム運用のアウトソーシングを受託するも、受発注者間でＳＬＡを明確にしておかなかったため、問題発生時混乱し、エンドユーザーに影響を与えた。

事例における見切りの内容

運用を通じてSLAを確立。

本来の判断の考え方

・運用負荷が重いという単純な理由ではなく、効果、顧客満足度など、総合的に判断した上でのビジネス展開を図るべき。

対処例

SIベンダー
運用負荷が重いという単純な理由ではなく、効果、顧客満足度など、総合的に判断した上でのビジネス展開を図る。

顧客
運用負荷が重いという単純な理由で外部委託するのではなく、エンドユーザーに対する運用責任は顧客にあるとの立場で組織を作り、運用する。

運用のルールの見直し、障害時運用の訓練によりSLAの見直し、改善を図る。

29 ソフト開発量の膨張で納期大幅延伸

顧客と合意した通信制御システムの機能仕様書が、他の業務システム並みに書かれているので、そのScopeで詳細設計・製造を行った。しかし、総合テストになると、仕様では想定していなかった通信状態などが判明。設計から試験までやりなおすバグ件数、修正量が多く、納期を大幅に延伸せざるを得なかった。

事例における見切りの内容

上流設計書のレビュー量が、他の一般システム並み以上であったので問題はないと見切った。

本来の判断の考え方

・中流工程には、状態遷移図が少なくとも書ききれる範囲であることを確認できてから進むべきであった。
・そもそも作りきれない規模であれば、そのリスクを顧客に説明責任を果たすべきであった。

対処例

SIベンダー
・PMは、状態遷移表などを用いた網羅性検証、開発Scopeが「現実に開発可能な範囲」であることの検証を行いリスクを明らかにする。リスクがあれば、Scope縮小を提案し、リスクと当初Scope維持とのトレードオフの議論と意思決定を顧客に提案する。
・それでも提案が受けれられないのであれば、PMではなく、PMの上司自らが顧客や顧客の上司にリスクを伝え、顧客の対応を要請する。

顧客
・リスクを伝えられた顧客が、それでも費用削減を決行する場合には、ベンダー任せで済ませず、リスクの発生に備えた、コンティンジェンシー・プラン（システム迅速復旧体制など）を自ら用意する。

顧客とPM/上司/営業の関係を「発注元―業者」の関係ではなく、プロジェクトリスクに関しては、双方が「リスクと闘うパートナー」となるような、リレーションシップを受注～上流工程前に構築しておくこと。

30 性能の異なる新旧資源混在環境でのトラブル

高性能化が進む最新のコンピュータ資源（CPU、記憶装置など）を導入するシステム更改にて、システムの完全移行まで一部性能が極端に低い旧式の資源が残り、新旧資源の混在状態が続くことになった。移行中にオンライン・システムが無応答の状態になった。

事例における見切りの内容

旧資源は、移行期間中の短期間しかアクセスがなく、性能上の着眼点は、新資源納入環境で、目標性能を達成するかという点に絞れば十分だと見切った。

本来の判断の考え方

・旧資源を含むオンライン・多重アプリケーション・プログラム走行環境では、特にデッドロック回避の視点でも、詳細設計をレビューするべきであった。

対処例

SIベンダー
特に旧資源へのアクセスを行うプロセスは、それだけ資源保留期間が長くなり、多重アプリケーション・プログラム走行環境ではデッドロックに遭遇する可能性が高まる。デッドロック回避の視点でも、詳細設計を実施する。

—

顧客
—

31 異常時挙動を想定しない設備設計

IPネットワークを高速LANで構築する際、要求通信トラフィックが通信容量の6〜7割に収まるように考えて設計した。部分障害が起こった際に、ネットワークが無応答状態となった。

事例における見切りの内容

要求通信トラフィックが通信容量以下なので、問題無いと考えた。

本来の判断の考え方

・IPネットワークでの異常時挙動も配慮したネットワーク設計を行うべきであった。

対処例

SIベンダー
・IPネットワークでは、異常時パケットの再送などで指数関数的にトラフィックが急増することもあるため、ネットワーク・ベンダーのSEを得て、設計をする。

—

顧客
—

32 基本設計書の品質不足の問題

基本設計書のユーザーレビューで設計書記述の整合性などに問題があることが判明。詳細設計フェーズに移行できず。要員増を含む体制強化と品質向上のためのルール作りを行い、設計書の全面見直し・修正を実施。工程は2カ月遅れたが、その後の工程でのトラブルはなく無事にカットオーバーを迎えた。

事例における見切りの内容

客先も工期変更（遅延）を認めたうえで、見切りはせず、基に戻ってやり直しを実施。

本来の判断の考え方

・（上記と同じ）

対処例

SIベンダー
・ドキュメントの記述内容・レベルの統一
・ドキュメントレビュールールの設定

—

顧客
—

33 仕様確定遅れによる問題

顧客側の体制が不足しており、仕様確定の遅れやレビューとその回答の遅延が頻発し、詳細設計終了時点で詳細仕様が未確定の部分があった。その後も仕様変更が重なり、テストフェーズでの手戻りが多発した。その結果、カットオーバーが遅延し、また、カットオーバー後の障害が多発した。

事例における見切りの内容

要員不足、工期不足の状態に対して、何とかなるという考えでプロジェクトを続行した。

本来の判断の考え方

・いったんプロジェクトを止め、残課題を洗い出し、その対策と優先順位、対策スケジュールを整理。
・その上で、顧客の体制強化、工期変更を要請する。

対処例

SIベンダー
顧客理由による工程遅延がある場合、適切なエスカレーションをとる。

顧客
仕様確定、レビューのできる要員の投入。

残課題の抽出と認識の共有化を図るための会議を頻度よく行う。

34 アプリケーション・プログラムの単体品質にばかりとらわれ、システム全体の納期・品質が守れず

製造工程にて十分品質を確保したはずのアプリケーション・プログラム群が、総合テストでの多重アプリケーション・プログラム走行試験中にアクセスの排他制御漏れによるデータ破壊などを多数引き起こした。土壇場で詳細設計からのやり直しが必要となり、システムの完成が大きく遅れた。

事例における見切りの内容

システム開発体制には経験豊富なアプリケーション・プログラム開発経験者が十分含まれており、OSやデータベースなどの環境は外部専門ベンダーに構築委託するので問題無いと見切っていた。

本来の判断の考え方

・プロジェクト体制には、ハードウェア、OS、データベース、ネットワークなどの基盤方式を抑えるメンバーを含めるべきだった。その担当者に各アプリケーション・プログラムからリソースへの標準アクセス法などの役割分担とすれば、問題は防げる。

対処例

SIベンダー
・システム開発体制には、基盤方式の基本原理が分かる「アプリケーション・プログラム共通」担当を必ず含めるべき。
・アプリケーション・プログラム共通担当者が各アプリケーション・プログラムからリソースへの標準アクセス法などを提供する。

顧客
—

—

35 業務アプリケーション・プログラム性能にとらわれシステム性能問題を見落とす

厳しい性能が要求された更新・参照系オンライン・システムで、すべてのアプリケーション・プログラムについて、あえてモジュール化をしないなどの対策をとることで高性能化を図る設計を行った。しかし、総合テストでは顧客の予算が削減されたことから、顧客は十分な高負荷テスト環境を用意して貰えずに、サービスに入り、商用システムにてレスポンス、ターンアラウンドが極端に落ちる事象が発生した。

事例における見切りの内容

すべてのアプリケーション・プログラムの個々の単体走行時性能を目標以上に高めたられる見通しが立ったので、十分だと誤認した。

本来の判断の考え方

・アプリケーション・プログラムの単体走行時性能に加え、アプリケーション・プログラム多重走行環境でアプリケーション・プログラム同士が同時には利用できない共用資源などSRR（Serially Reusable Resources）を虱つぶしに調べ上げ、それによる性能低下も見切る必要があった。
・また特に高レスポンスが要求されるアプリケーション・プログラムプロセスの実行優先度が高くなる設計も必要であった。
・以上を行ったとしても、高負荷テスト環境が無いとシステム性能低下のリスクが残るので、顧客にそのリスクを説明するべきであった。

対処例

SIベンダー
・共用資源などSRR（Serially Reusable Resources)の特定と、それによる性能低下も加味したシステム性能を詳細設計段階で検証する。
・また多重走行環境で特に高レスポンスが要求されるアプリケーション・プログラムプロセスの実行優先度は他より高く設計する。
・さらに、PMは、高負荷試験環境が無いことによるリスクを説明し、このリスクと予算削減とのトレードオフの議論と意思決定を顧客に提案する。
・それでも提案が受けれられないのであれば、PMではなく、PMの上司自らが顧客や顧客の上司にリスクを伝え、顧客の対応を要請する。

顧客
・リスクを伝えられた顧客が、それでも納期前倒しを決行する場合には、ベンダー任せで済ませず、リスクの発生に備えた、コンティンジェンシー・プラン（入呼規制運用など）を自ら用意する。

顧客とPM/上司/営業の関係を「発注元―業者」の関係ではなく、プロジェクト・リスクに関しては、双方が「リスクと闘うパートナー」となるような、リレーションシップを受注～上流工程前に構築しておくこと。

36 仕様通りに開発したが運用障害で事業が長時間停止

顧客はシステム/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。

事例における見切りの内容

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。

本来の判断の考え方

・SIベンダーとしては、月跨り、年跨り、閏年跨り時など、システムのライフサイクルを通じた運用についても仕様があいまいであれば、顧客に提案するべきであった。

対処例

SIベンダー
・PMは、システムのライフサイクルを通じた運用についても仕様があいまいであることによるリスクを説明し、このリスクと予算追加とのトレードオフの議論と意思決定を顧客に提案する。
・それでも提案が受けれられないのであれば、PMではなく、PMの上司自らが顧客や顧客の上司にリスクを伝え、顧客の対応を要請する。

顧客
・リスクを伝えられた顧客が、それでも納期前倒しを決行する場合には、ベンダー任せで済ませず、リスクの発生に備えた、コンティンジェンシー・プラン（データ・リカバリ運用の追加など）を自ら用意する。

顧客とPM/上司/営業の関係を「発注元―業者間」の関係ではなく、双方が「ビジネス・パートナー」となるような、リレーションシップを受注～上流工程にてあらかじめ構築しておくこと。

37 システム移行設計不良

24時間運転の大規模オンライン・システム更改に関する移行設計にて、エンドユーザー部門が多いため、顧客は切り戻し仕様まで含めた調整が困難であることから、旧システムには切り戻さず、システム切替作業を続行する移行仕様を決めた。数度にわたる総合テストでの移行リハーサルはうまくいったが、年末年始の移行本番の途中で障害が発生。障害が復旧するまで長時間を要してしまったので、ミッション・クリティカルなシステムのオンライン開始が大幅に遅れ、エンドユーザー部門の事業機会が損なわれてしまった。顧客は、移行中の障害を起こしたベンダーを責めたが、エンドユーザー部門からは顧客も責められることになった。

事例における見切りの内容

顧客が決めた仕様だから、それに愚直に従い、要求された移行上限時間をリハーサルで検証するので、問題はないはずだと見切った。

本来の判断の考え方

・旧システムには切り戻さない移行仕様とすることによるリスクに関する説明責任を果たすべきだった。

対処例

SIベンダー

・PMは、旧システムには切り戻さない移行仕様とすることによるリスクを説明し、このリスクとエンドユーザー部門交渉の労とのトレードオフの議論と意思決定を顧客に提案する。
・それでも提案が受けれられないのであれば、PMではなく、PMの上司自らが顧客や顧客の上司にリスクを伝え、顧客の対応を要請する。

顧客

・リスクを伝えられた顧客が、それでも納期前倒しを決行する場合には、ベンダー任せで済ませず、リスクの発生に備えた、コンティンジェンシー・プラン（エンドユーザー部門の事前了解、入呼規制運用など）を自ら用意する。

顧客とPM/上司/営業の関係を「発注元―業者」の関係ではなく、プロジェクト・リスクに関しては、双方が「リスクと闘うパートナー」となるような、リレーションシップを受注～上流工程前に構築しておくこと。

38 製品間インターフェース保証の見落とし

実績が十分にある市販のOS、データベース、ミドルウェア、業務パッケージ・ソフトウェアでシステムを構築したが、サービス開始後、システムが時々停止するなど、動作が不安定となった。

事例における見切りの内容

実績が十分にある市販製品を使うのだから、問題ないと見切った。

本来の判断の考え方

・市販製品間における相性についても検証するべきだった。

対処例

SIベンダー

すべての市販製品ベンダーにシステムが使う市販製品群のバージョンでの組み合わせが連動保証の範囲か確認すること。

顧客

―

―

39 有名ソフトを未検証のまま使用し不具合発生

業界で名の知れたソフトウェアを適用規模、性能条件、機能制約などの詳細を未検証のまま使用し、総合テスト以降不具合発生。

事例における見切りの内容

業務プログラム、ハードウェアなどを変更、増強し対処した。

本来の判断の考え方

・カタログ・ベースにとらわれず、実機にて確認、検証すべき。

対処例

SIベンダー
カタログ・ベースにとらわれず、実機にて確認、検証すべき。

顧客
ソフトウェア設計のレビュー時、採用ソフトの使用実績を確認、検証する。

構築するソフトの採用に当たっては、ベンダー、顧客両者で検証、確認を行い、リスクを共有認識する。

40 営業主導で決めたソフトで開発し要員不足、クレーム、不具合発生

開発、保守体制弱小なベンチャー企業が開発したパッケージ・ソフトウェアを営業主導で採用し、システム構築したが、パッケージ・ソフトウェアに対する要求要件が拡大した。つど、カスタマイズするものの、開発、保守要件に体制が追い付かず、問題発生。

事例における見切りの内容

パッケージ・ソフトウェアを採用したSIベンダーが、工数、費用を投入し開発、保守を行うこととなった。

本来の判断の考え方

・外部のパッケージ・ソフトウェアを活用する場合は、該当システムの将来まで見込んで決定すべき。
・また、著作権などの法的の扱いも明確にしておくべき。

対処例

SIベンダー
パッケージ提案は、性善説になりがちで、問題、制約が伝わりにくい。より厳密に検証、提案すること。

顧客
パッケージの提案は、一般的に良い事柄のみ出る。よって、使用条件を詳細に検討し、実現可否を検証すること。

外部のパッケージ・ソフトウェアを活用する場合は、該当システムの将来まで見込んで決定すべき。また、著作権などの法的の扱いも明確にしておくべき。

41 パッケージ採用と言いながらカスタマイズ量過多（実質固有ソフト）

中小規模のユーザーとの契約時、現行システムをパッケージ適用により実現する旨の約束をした。しかし、現行システムをパッケージにてシステム構築しても、運用的に困難。また、ユーザー、及びベンダーの体制も小規模であり、機能実現、納期、費用などで大問題発生、以降のビジネスに影響を与えた。

事例における見切りの内容

実現機能、納期、費用などを再交渉。結局、当初の計画を縮し、システムを実現。以後、ビジネス上の付き合い消滅。

本来の判断の考え方

・ユーザー
ベンダーに丸投げしないこと。
・ベンダー
中小規模ユーザーといえどもパッケージの適用が容易と判断しないこと。

対処例

SIベンダー
・中小規模ユーザーにパッケージを容易に押し付けないこと。
・パッケージ適用検証は慎重に行う必要あり。

顧客
小規模システムといえども、ベンダーに丸投げしないこと。

・ユーザー
ベンダーに丸投げしないこと。
・ベンダー
中小規模ユーザーといえどもパッケージの適用が容易と判断しないこと。

42 パッケージ機能の認識、期待感のギャップ大。規模、工数、費用、納期大幅ズレ

パッケージの詳細機能について、ベンダーと顧客でギャップがあり、力関係から顧客の言い分をすべてパッケージに組み込まざるを得なくなった。また、費用についても、本来パッケージで備えるべきものか、カスタマイズかで揉めた。

事例における見切りの内容

つど、カスタマイズ機能、標準機能かを調整しながら、システム構築。費用はベンダー持ち出し大。

本来の判断の考え方

・パッケージ適用に当たっては、ユーザーのエンド部門を交え、パッケージの内容を長時間かけ、詳細検証すべき。

対処例

SIベンダー
パッケージ適用に当たっては、契約時ユーザーとの検証を行い、確定することも取り決める。

顧客
パッケージ適用に当たっては、ユーザーのエンド部門を交え、パッケージの内容を長時間かけ、詳細検証すべき。

安易にパッケージ提案、採用せず、他での問題事例を参考に詳細検証する。

43 特定顧客で未完パッケージを完成させる計画が頓挫

パッケージ適用を前提に、システム構築したが、商談がカタログベースで進行し、かつ該当ユーザーをベースにパッケージ開発する旨約束しているため、実質、個別パッケージとなった。

事例における見切りの内容

パッケージ開発の方針をあきらめ、個別開発とした。他ユーザーへのパッケージとしての適用は、流用という概念のみとした。また、費用は、パッケージ費用売上げに対し、増加した分をすべてベンダー負担。

本来の判断の考え方

ユーザー、ベンダー間で事前に
・パッケージの共同開発
・パッケージの適用検証
・パッケージの適用
などを明確にしておき、実現不可能な計画を立てない。

対処例

SIベンダー
・あらかじめ、パッケージの開発方針を明確にしておく。
・仮に該当ユーザーをベースにパッケージ開発するならば、開発、検証期間を十分に確保すること。

顧客
安易にパッケージの提案を受け入れず、使用条件とパッケージの充足状況、将来計画、費用の扱いをベンダーと取り決める。

ユーザー、ベンダー間で事前に
・パッケージの共同開発
・パッケージの適用検証
・パッケージの適用
などを明確にしておき、実現不可能な計画を立てない。

44 システム全体としての信頼性設計の不備

顧客の強い要請により、システムの個別コンポーネントの故障に備え、2重・3重の冗長構成をとったものの、アプリケーション・プログラム含めたシステム全体のリカバリ仕様が複雑となった。開発しきれないほどの製造規模・テスト項目数となった。結局、出荷後、異常時にはソフトバグによる、長期障害が頻発した。

事例における見切りの内容

システムの個別コンポーネントの故障に備え、2重・3重の冗長構成をとったので、要求される信頼性が確保できると考えた。

本来の判断の考え方

・ハードウェアの信頼性は冗長構成などで向上するが、ソフトによる信頼性向上機能は開発中にバグを虱つぶしに見つけ出すしか方法は無い。この限界について顧客に対する説明責任を果たすべきだった。

対処例

SIベンダー
・PM はシステム全体の信頼性バランス（ハードウェア、ソフトウェア）を考慮し、ソフトウェア制御方式の全貌が俯瞰できる範囲にソフトウェアの Scope を縮小することを顧客に提案をするべき。
（PM は、このソフトウェア仕様とする）
・それでも提案が受けれられないのであれば、PMではなく、PMの上司自らが顧客や顧客の上司にリスクを伝え、顧客の対応を要請する。

顧客
リスクを伝えられた顧客が、それでも納期前倒しを決行する場合には、ベンダー任せで済ませず、リスクの発生に備えた、コンティンジェンシー・プラン（障害時ソフトウェアによらない運用・手動による対処など）を自ら用意する。

顧客とPM/上司/営業の関係を「発注元―業者」の関係ではなく、プロジェクト・リスクに関しては、双方が「リスクと闘うパートナー」となるような、リレーションシップを受注～上流工程前に構築しておくこと。

45 過負荷・限界系の見落とし

顧客要求性能を超える処理能力が出るような設計の見通しが得られた。しかし、総合過負荷テストでシステムからの応答が何時までも返ってこなかった。

事例における見切りの内容

目標性能達成の見通しが得られたので、問題ないと見切った。

本来の判断の考え方

・資源を限界まで使った場合のシステムの挙動という観点での設計も行うべきであった。

対処例

SIベンダー
起こり得るメモリーなどの過負荷、設計容量を超える資源要求発生などのケースをすべて洗い出した設計・製造のレビューを行うこと。

顧客
―

―

46 方式性能ボトルネックの見落とし

オンライン中に頻繁に起動/終了するアプリケーション・プログラムについて単電文処理で目標を超える見通しを詳細設計で付けたが、総合テストでのアプリケーション・プログラム多重走行時、レスポンスが悪化。すべてのアプリケーション・プログラムが、起動/終了時に資源を確保/開放する設計にしていた。

事例における見切りの内容

アプリケーション・プログラム単体性能が十分であり、問題ないと見切った。

本来の判断の考え方

・アプリケーション・プログラムに必要な資源確保をアプリケーション・プログラムごとにするか、システム全体で確保すべきか、詳細設計当初に検討すべきであった。

対処例

SIベンダー	顧客
アプリケーション・プログラムに必要な資源確保をアプリケーション・プログラムごとにするか、システム全体で確保すべきか、詳細設計当初に、システム全体に求められる要件から検討・決定すべきであった。	—

—

47 最新ソフト、バージョンを検証不足のまま採用。不具合多発

オペレーティングシステム・ミドルソフトなどをバージョンアップしたが、細かい点で互換性に制約が発生。制約問題がないだろうという認識で計画を立てているため、検証体制不足、問題発生時の体制不備、検証期間不足などにより不具合、及びプロジェクト計画が混乱。

事例における見切りの内容

本番開始時期を延期した。

本来の判断の考え方

・同一製品体系のソフトでも互換性（パッチ適用も含む）については、特に念入りに検証すべく、プロジェクト計画を立てる。

対処例

SIベンダー	顧客
制御系システムは、ベンダーの責任を負う所が大である。表面的な検証結果にとらわれず、過去発生した問題、互換性を中心に慎重な検証を実施する。	同一ベンダーでのバージョンアップを安易に考えず、納期遅延などを考慮し、コンティンジェンシー・プランを設定する。

同一製品体系のソフトでも互換性（パッチ適用も含む）については、特に念入りに検証すべく、プロジェクト計画を立てる。

48 自前ソフト以外の検証、確認認識が薄く、後工程で問題発生

ユーザー指定の流通ソフトを組み込み、システムを構築するが、使用経験、及び社内に対応部門が無く、検証の勘所がつかめず、スケジュールが遅延した。

事例における見切りの内容

問題発生つど、ユーザー経由で開発ベンダーに問い合わせ処置を行った。

本来の判断の考え方

・指定ソフトの扱い、制約、問題発生時の処置方法などを事前にユーザーと取り決めておく。

対処例

SIベンダー	顧客
安易にユーザー指定の流用ソフトを使用せず、事前にシステム構築のための準備、検証を行う。	指定ソフトをシステム構築条件にする場合、ＳＩベンダーに任せきりにせず、自ら検証する。

指定ソフトの扱い、制約、問題発生時の処置方法などを事前にユーザーと取り決めておく。

49 アプリケーション・プログラムの生産性で言語を選定したが、性能問題がネックとなった

生産性が高いといわれるオブジェクト指向系の新しい言語が流行ってきたので、顧客から、その採用を根拠に、オンライン・システムのアプリケーション・プログラム開発費削減を要求された。テストで予想以上に性能が出なかったことに加え、オブジェクトのネストが深いところでのトラブルが発生すると解析が困難となったので、開発途中に既存言語で大規模な作り直しをすることになった。

事例における見切りの内容

従来言語に比べ生産性が高くコーディングができた実績を確認したので、採用してもよいと考えた。

本来の判断の考え方

・新言語の適用分野での実績を確認し、実績が無い場合、問題が発生したら、代替言語で開発する予備費を積んで費用見積りを行うべきであった。

対処例

SIベンダー

・オブジェクト指向系の新しい言語を採用する際には、コーディング前に性能条件が厳しいオンライン・システムや、複雑なオブジェクト間インターフェースがある制御系ソフトなどへの適用実績を確認する。
・実績が無い場合、問題が発生したら、代替言語で開発するコンティンジェンシー・プランを織り込んで、中流工程をすすめる。

顧客

—

—

50 開発ルール未確定のため保守性、流用性が悪い

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。

事例における見切りの内容

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。

本来の判断の考え方

・設計、レビュー、テストなどのプロセスのルールに加え、プログラム作成ルールを定め、全員に徹底しておく。

対処例

SIベンダー

特に、プログラム製造を海外に発注する場合、ルールを明確にするとともに必ず事前説明をする。

顧客

—

—

51 新しい言語の熟練者不足でプログラム完成度が低い

ユーザーから、将来性の観点から新しい言語を採用する指定あり。しかし、新しい言語に熟知した技術者が不足し、やむを得ずスキルの低いメンバーで開発したが、結果的に品質の低いプログラムが出来上がった。

事例における見切りの内容

つど、新しい言語経験者を集め、そのメンバーを中心にレビューを実施し、品質の悪いプログラムの改修を行った。

本来の判断の考え方

・要員アサインのポイントは、要員数より、質が重要。あらかじめ条件を明確にした上で人選する。
・該当プロジェクトを通して、経験を積ませる計画を立てることは厳禁である。

対処例

SIベンダー
要員不足なら、外部の要員をリーダーにした体制を作る。

顧客
—

—

52 客観的な生産性指標が少なく、工数、費用にインパクト大

ユーザー、ベンダーともに、客観的で納得感のある生産性を持ち合わせていない。スケジュール、工数、費用を交渉するに当って、無理な生産性で計画を立て、帳尻を合わせるが結果的には実現不可となった。

事例における見切りの内容

ベンダー側が、要員の増強、費用を負担した。

本来の判断の考え方

・生産性は、希望的観測値になり易いため、事前に事例、難易度、リスクなどを踏まえて確定する。

対処例

SIベンダー
生産性は、希望的観測値になり易いため、事前に事例、難易度、リスクなどを踏まえて確定する。

顧客
—

—

53 担当者にシステム的認識欠落、障害時の切り分け、情報採取の仕組み不足

アプリケーション・プログラム担当者にシステム的認識がなく、性能、アプリケーション・プログラムに問題が発生した時の切り分け、情報採取などの配慮がされていないため、結合テスト、総合テストに手間取った。

事例における見切りの内容

システム系担当者が、アプリケーション・プログラムに手を加え対応した。

本来の判断の考え方

・各種ルールの制定に当って、表面的なもの以外に効率化など内容的なものを組み込んでおく。

対処例

SIベンダー
・あらかじめプログラミング・ルールに効率化のための施策を組み込んでおく。
・ツールを共有化し、関係者が容易に利用可能とする。

顧客
—

—

54 オフショアで品質と納期遅延の問題（発注側問題）

日本語が堪能なブリッジSEがいるオフショア会社なので、従来の国内外注先と同じ仕様精度でプログラム設計～単体テストを発注した。忠実で才能あるプログラマは予定より早く納品したが、結合テスト以降、インターフェース・バグなどが多発。システム全体では納期を守れなかった。

事例における見切りの内容

「日本語が堪能」であることにより「日本の慣習・文化に熟知」と判断した。

本来の判断の考え方

・インドなどは米国に近い低コンテクスト社会（日本のソフト開発会社のようには仕様にない「行間を読んでくれる」力や文化は期待できない）と判断すべき。

対処例

SIベンダー	顧客
・オフショア先会社は低コンテクスト文化にあるとの前提の下で、誰が見ても解釈が変わらない仕様にした上で発注する。	—
—	

55 オフショアで品質と納期遅延の問題（受注側問題）

製造工程をオフショア会社に一括発注した。プログラマ個々人の才能はあり、予定より早く納品したが、コーディング規約など全体統制力に乏しく、後続工程で品質、納期遅延問題が発生した。

事例における見切りの内容

プログラマ個々人の才能はあり、技術的には大きな問題はないと見切った。

本来の判断の考え方

・コーディング規約、構成管理など全体統制を行う仕組みがあるかという点でも検証する。

対処例

SIベンダー	顧客
・オフショア先の開発プロセス標準を選定前に調べる。 ・不明確の場合、自社コーディング規約を開示するなどして、全体統制がとれるようにする。	—
—	

56 外注会社責任者は担当者に任せ切り

協力会社に作業を発注したが、責任者は内部に深入りせず、担当者に任せ切り。進捗指摘、問題指摘などは、実質発注元が行わざるを得ない状況であった。

事例における見切りの内容

下請法などを考慮しつつも、発注元が協力会社責任者に問題指摘し対処した。

本来の判断の考え方

・協力会社に発注する際、発注先責任者を組み入れた体制、役割を明確にしておく。

対処例

SIベンダー	顧客
契約前に発注先の評価を行い、仮にスキルがあっても、管理面で不安な場合は、契約形態を請負から派遣に変更する。	—
—	

57 特定ノウハウについて外注依存。SIベンダーのコントロール不可

システム構築に当って、特定のノウハウを所有している協力会社に発注せざるを得ない。しかし、発注元にスキルが無いため、プロジェクトとしてのコントロールが行き届かない。

事例における見切りの内容

一般的なプロジェクト管理を行い、実質、該当協力会社に依存していた。責任者との情報交換を密にし、ビジネス的な観点での付き合いを強化していた。

本来の判断の考え方

・たとえ特定のノウハウを所有している協力会社といえども、ビジネスとして毅然とした対処をする。

対処例

SIベンダー
細かい契約を締結し、場合によりペナルティ条項を盛り込み、一方的に協力会社ベースにならないようあらかじめ考慮する。

顧客
—

—

58 行動規範順守が困難。機密情報漏えいの問題

情報漏えい防止に当って、契約者の取り交わし、PCの搬入出の届け、個人PCの使用禁止策などを講じるも、進捗上、重要データを個人持ち帰りで作業をし、結果的に情報漏えい、またはその危険にさらされるケースがある。本人問題認識はあるものの、進捗遅れ挽回という片一方の大義名分を意識しているため、問題が複雑であった。

事例における見切りの内容

・該当協力会社とは取り引き停止とした。
・ユーザーへの損害賠償を行った。
・管理強化した。

本来の判断の考え方

・常に、プロジェクト・メンバーに留意する旨、指示する。また、進捗報告時、担当者が追い込まれるあまり、不正行為に走る恐れがないように配慮する。

対処例

SIベンダー
特に重要プロジェクトでは、入出門監視装置の設置、入出門時の手荷物検査を実施する。

顧客
—

—

付録 4. 中流工程分析ツール

	ヒアリングシート	
① 上流工程完了結果の要件機能検証、確認	H1,H4,H5,H6,H7,H12,H14,H15,H16,H17,H18,H19, H20,H21,H22,H23,H24,H26,H31,H32,H33,H34, H37,H38,H39,H40,H42,H45,H46,H48,H50,H51, H52,H53,H54,H59,H60,H61,H62,H69,H70,H71	
●契約書の条項と要件定義書（仕様書）をレビューし、確認したか ・RFP、提案の視点 ・受注の視点 ・法令の視点	H1,H4,H5,H6,H7,H11,H12,H13,H15,H16,H20,H22, H23,H24,H26,H28,H31,H32,H34,H35,H37,H38, H39,H40,H42,H45,H46,H47,H48,H49,H50,H51, H52,H53,H60,H61,H62	
●システム設計書（方式仕様書）をレビューし実現性を検証したか ・システム化の視点 ・技術の視点 ・ユーザーの視点	H1,H2,H3,H4,H5,H6,H7,H8,H10,H12,H14,H15,H20, H23,H24,H27,H33,H35,H36,H37,H38,H46,H50, H51,H52,H53,H54,H60,H61,H64,H65,H66	
●プロジェクト計画書をレビューし、妥当性、リスクを検証したか ・人的資源の視点 ・タイムの視点 ・品質の視点 ・コミュニケーションの視点	H1,H4,H6,H12,H14,H15,H16,H17,H18,H19,H20,H21, H22,H23,H24,H25,H26,H27,H28,H29,H31,H32, H33,H34,H38,H39,H40,H41,H42,H43,H44,H45, H46,H47,H48,H49,H50,H51,H52,H53,H54,H60, H61,H62,H63,H64,H65,H66,H69,H70,H71	
② 上流工程結果を受けた上流工程の見直し	H1,H5,H6,H11,H12,H13,H14,H15,H16,H17,H18,H19, H21,H22,H23,H28,H31,H32,H33,H34,H35,H36, H37,H38,H39,H40,H41,H42,H45,H46,H49,H50, H52,H53,H60,H61	
●契約書と要件定義書の見直しをお客様に要求したか ・要件定義の再実施 ・契約の見直し	H1,H5,H6,H11,H12,H14,H15,H16,H17,H18,H19,H21, H22,H28,H31,H32,H33,H34,H35,H37,H38,H39,H40, H42,H45,H47,H48,H49,H50,H52,H53,H60,H61	
●システム設計書の見直し交渉をお客様に行ったか ・システム設計の再実施	H2,H3,H5,H6,H7,H8,H10,H11,H12,H15,H16,H23, H27,H35,H36,H37,H38,H39,H53,H60,H61,H64, H65,H66	
③ 上流工程結果の実現性（ユーザー要件、外部制約）の確認、検証	H3,H4,H10,H11,H12,H15,H21,H22,H33,H39,H45, H52,H54,H60,H61,H62	
●コスト（費用）の制約を満たせるか	H45	
●ハードウェアやパッケージソフト（物と情報）の制約はクリアできるか	H10,H11,H33	
●要員のスキルや人数（人）の制約で問題が生じないか	H4,H12,H15,H22,H39,H52,H54,H60,H61	
●役割、権限、責任など（組織体制）の社内的な制約での影響はないか 【発注側、受注側ともに】	H20,H21,H24,H46,H51,H62	
●工期（時間）の制約は大丈夫か	H44	
●開発環境、テスト環境は整備されているか		
④ 上流工程結果並びに仕様・納期・費用変更を受けたプロジェクト計画の見直し	H1,H14,H15,H19,H21,H22,H33,H38,H39,H44,H45, H52,H53,H61,H62,H69,H70,H71	
●プロジェクトに影響する内容を確認し、プロジェクト計画書の見直しを実施したか ・人的資源の視点 ・タイムの視点 ・品質の視点 ・コミュニケーションの視点 ・コストの視点	H1,H14,H15,H19,H21,H22,H38,H39,H45,H52,H53, H61,H62,H69,H70,H71	
●完了した工程、計画差異に合わせたリスクの見直しは実施しているか ・リスクの視点	H33,H44	
⑤ 上流工程完了結果の結合テスト、総合テストへの情報提供	H47,H49,H56,H57	
●RFP、提案のレベルで再検討された内容の反映を実施しているか	H49	
●要件定義、システム設計の結果確定した情報の反映を実施しているか	H47,H56,H57	

	測定分析データー覧	(①～⑭が対処できなかったため発生した) 事例	ヒアリングシートから引用した 自動マッピングの結果
	Te1,Te2,K1,K2,K3,R1,R2,R3,Q10,Q11,Q12, Q13,Q14,Q15,Q16,Q17,Q18,Q19,T8	J2,J3,J4,J5,J6,J7,J8,J9,J10,J11,J12,J13,J14, J15,J16,J18,J19,J20,J21,J27,J28,J29, J32,J33,J35,J36,J37,J39,J40,J41,J42,J43, J44,J49,J50,J51,J52,J54,J55,J56,J57,J58	H1,H4,H5,H6,H7,H12,H14,H15,H16,H17, H18,H19,H20,H21,H22,H24,H26,H31, H32,H33,H34,H39,H40,H42,H45,H46, H48,H50,H51,H52,H53,H54,H59,H60, H61,H62,H69,H70,H71
	K1,K2,K3,O3	J3,J8,J10,J11,J14,J15,J16,J18,J19,J39,J40, J41,J42,J43,J44,J49,J57,J58	H4,H7,H12,H15,H20,H22,H24,H26,H33, H39,H45,H46,H50,H51,H52,H53,H60, H61,H62
	K1,K2,K3	J13,J14,J15,J16,J19,J21,J29,J39,J40,J41, J42,J44,J49,J51,J57	H1,H4,H7,H12,H14,H15,H20,H24,H33,H46, H50,H51,H52,H54,H60
	Te1,Te2,K1,K2,K3,M1,M2,Cm1,Cm2,Cm3, Cm4,H1,H2,H3,H4,H5,H6	J3,J4,J5,J6,J7,J8,J9,J10,J11,J12,J13,J14, J15,J16,J19,J20,J21,J27,J29,J32,J33,J35, J37,J42,J49,J50,J51,J52,J54,J55,J56,J57, J58	H1,H4,H6,H12,H14,H15,H16,H17,H18,H19, H20,H21,H22,H24,H26,H31,H32,H33, H34,H39,H40,H42,H46,H48,H50,H51, H52,H53,H54,H60,H61,H62,H69,H70, H71
	Te1,K1,K2,K3,M1,M2,R1,R2,R3,H6,H7,H8, Q10,Q11,Q12,Q13,Q14,Q15,Q16,Q17,Q18,Q19, T8	J2,J5,J8,J13,J14,J15,J16,J18,J32	H14,H15,H21,H22,H33,H39,H45,H50,H52
	K1,K2,K3,R3	J5,J13,J14,J15,J16,J18	H14,H15,H21,H45,H50,H52
	K1,K2,K3	J32	H39
	Te1,Te2,K1,K2,K3,M1,M2,H1,H2,H3,H4,H5, H6,C1,T7,T8,T10,T11,T12	J2,J5,J8,J9,J10,J11,J15,J18,J31,J39,J40, J41,J42,J43,J47,J48,J49,J51,J57	H3,H4,H10,H11,H12,H15,H21,H22,H33, H39,H45,H52,H54,H60,H61,H62
	K1,K2,K3,C1	J18	H45
	K1,K2,K3	J2,J39,J40,J41,J42,J43,J47,J48	H10,H11,H33
	Te1,Te2,K1,K2,K3,M1,M2,H1,H2,H3,H4,H5, H6	J8,J10,J11,J15,J49,J51,J57	H4,H12,H15,H22,H39,H52,H54,H60,H61
	Te1,Te2,T10	J5,J9,J19	H20,H21,H24,H46, H51,H62
	K1,K2,K3,T7,T8	J17	H44
	T11,T12		
	Te2,K1,K2,K3,R1,R2,R3,Cm1,Cm2,Cm3, Cm4,H1,H2,H3,H4,H5,H6,C1,T9	J1,J2,J3,J5,J6,J8,J9,J10,J11,J12,J13,J15,J17, J18,J32	H1,H14,H15,H19,H21,H22,H33,H38,H39, H44,H45,H52,H53,H61,H62,H69,H70,H71
	Te2,K1,K2,K3,Cm1,Cm2,Cm3,Cm4,H1,H2, H3,H4,H5,H6,C1,T9	J1,J3,J5,J6,J8,J9,J10,J11,J12,J13,J15,J18, J32	H1,H14,H15,H19,H21,H22,H38,H39,H45, H52,H53,H61,H62,H69,H70,H71
	K1,K2,K3,R1,R2,R3	J2,J17	H33,H44
	K1,K2,K3,Cm1,Cm2,Cm3,Cm4,H6,T8	J22,J23,J24,J25	H47,H49,H56,H57
	K1,K2,K3	J22,J23	H49
	K1,K2,K3	J24,J25	H47,H56,H57

	ヒアリングシート	
⑥ ソフトウェア設計の実施	H1,H4,H6,H8,H9,H12,H17,H18,H23,H24,H31,H33,H39,H50,H55	
●上流工程の不備をエスカレーションしているか	H1,H23,H24,H33,H39,H50	
●設計にあたって必要な基準は明確になっているか ・標準化の基準 ・構造化の基準 ・命名規則	H4,H6,H8,H9,H12,H17,H18,H23,H31,H39,H55	
⑦ ソフトウェア設計時の要件機能検証、確認	H1,H7,H13,H20,H24,H33,H38,H39,H44,H46,H50,H51,H53,H61,H62	
●ソフトウェア設計時に要件からのトレーサビリティを確認しているか ・要件定義、システム設計の再実施	H1,H7,H13,H19,H24,H33,H38,H39,H44,H50	
●ソフトウェア設計時に見積条件との差異は発生していないか ・契約の見直し	H1,H12,H13,H15,H20,H24,H33,H38,H39,H40,H45,H46,H51,H52	
⑧ OS、DB、NW、言語、組み込みソフト、流通ソフトなどの制約条件検証、確認	H1,H2,H3,H4,H5,H6,H7,H8,H9,H10,H11,H12,H15,H23,H33,H42,H47,H52,H55,H56	
●機能、性能、インターフェースの確認。特に商用システムでの実績を検証しているか	H1,H3,H4,H6,H8,H9,H10,H11,H12,H33,H47,H56	
●異常、輻輳、障害、不整合時のインターフェース、動作の確認、およびテスト、運用ドキュメントへの反映を実施しているか	H2,H3,H4,H5,H7,H55	
●異バージョンとの互換性、排他制御／SRR見落とし、システム制限値、システム変更時の影響範囲の確認、および保守、維持ドキュメントへの反映を実施しているか	H10,H15,H23,H42,H52	
⑨ ソフトウェア設計結果の結合テスト、総合テストへの情報提供	H3,H5,H6,H13,H38,H49	
●RFP、提案に関わるレベルまで再検討された内容の反映を実施したテスト・ケースになっている	H49	
●要件定義、システム設計まで遡って変更された内容の反映を実施したテスト・ケースになっている	H3,H6	
●ソフトウェア設計の結果確定した情報の反映を実施したテスト・ケースになっている		
⑩ プログラミング、単体テストの実施	H58	
●コードインスペクションは実施しているか ・仕様との整合性確認 ・コーディング規約順守状況確認 ・わかりやすいプログラムであることの確認		
●単体テスト実施内容の確認を実施しているか ・単体テスト計画 ・テスト・ケース		
●単体テスト時の障害発生状況の確認と分析を実施しているか	H58	
●問題、課題の発生状況と内容の確認と分析を実施しているか		
⑪ ソフトウェア設計の見直し	H17,H18	
●単体テスト結果のプログラム変更、修正のソフトウェア設計書への反映を実施しているか ・プログラムと設計書の整合性維持の観点		
●単体テストの結果判明した課題を反映するための、ソフトウェア設計のやり直しを実施しているか		

	測定分析データー覧	(①～⑭が対処できなかったため発生した) 事例	ヒアリングシートから引用した 自動マッピングの結果
	K1,K2,K3,O1,O2,Q1,Q17,Q18,Q19,T6,S1,S2,S3,S4	J2,J4,J16,J29,J32,J33,J34,J45,J46,J49,J50,J53,J54	H1,H4,H6,H8,H9,H12,H17,H18,H23,H24,H31,H33,H39,H50,H55
	K1,K2,K3,O1,Q17,Q18,Q19,T6,S1,S2,S3,S4	J2,J16,J29,J32,J33,J34	H1,H23,H24,H33,H39,H50
	K1,K2,K3,O1,O2,Q1	J4,J34,J45,J46,J49,J50,J53,J54	H4,H6,H8,H9,H12,H17,H18,H23,H31,H39,H55
	K1,K2,K3,O1,O2,O4,R1,R2,R3,Cm1,H1,H2,H3,H4,H5,Q1,Q2,Q3,Q4,Q5,Q6,Q7,Q8,Q9,Q11,Q12,Q13,Q14,Q15,Q16,Q17,Q18,Q19,T6,T8,S1,S2,S3,S4	J1,J2,J3,J17,J16,J19,J20,J21,J29,J32,J33,J40,J41,J42,J43,J44	H1,H7,H13,H20,H24,H33,H38,H39,H44,H46,H50,H51,H53,H61,H62
	K1,K2,K3,O1,O2,O4,Q1,Q2,Q3,Q4,Q5,Q6,Q7,Q8,Q9,Q11,Q12,Q13,Q14,Q15,Q16,Q17,Q18,Q19,T6,T8	J1,J2,J10,J12,J14,J16,J17,J29,J32,J33,J40,J41,J42,J44	H1,H7,H13,H19,H24,H33,H38,H39,H44,H50
	K1,K2,K3,O1,S1,S2,S3,S4K1,K2,K3,O1,S1,S2,S3,S4	J1,J7,J13,J15,17,J18,J19,J20,J21,J29,J42,J43,J49	H1,H12,H13,H15,H20,H24,H33,H38,H39,H40,H45,H46,H51,H52
	K1,K2,K3,O1,H1,H2,H3,H4,H5,H6,H7,T6,T7,T8,Q3,Q10,Q16,Q18	J2,J15,J24,J29,J30,J31,J34,J35,J36,J37,J38,J39,J40,J41,J42,J43,J44,J45,J46,J47,J48,J49,J53	H1,H2,H3,H4,H5,H6,H7,H8,H9,H10,H11,H12,H15,H23,H33,H42,H47,H52,H55,H56
	K1,K2,K3,O1,H1,H2,H3,H4,H5,H6,H7,T6,T7,T8,Q3,Q10,Q16,Q18	J2,J24,J29,J31,J35,J38,J39,J40,J41,J42,J43,J45,J46,J47,J48,J49	H1,H3,H4,H6,H8,H9,H10,H11,H12,H33,H47,H56
	K1,K2,K3,O1,H1,H2,H3,H4,H5,H6,H7,T6,T7,T8,Q3,Q16,Q18	J30,J31,J36,J44,J53	H2,H3,H4,H5,H7,H55
	K1,K2,K3,O1,H1,H2,H3,H4,H5,H6,H7,T6,T7,T8,Q3,Q16,Q18	J15,J34,J37,J47	H10,H15,H23,H42,H52
	K1,K2,K3,O1,O2,O4,O5,Cm1,Cm2,Cm3,Cm4,H1,H2,H3,H4,H5,H6,H7,T8	J1,J22,J31,J36,J38	H3,H5,H6,H13,H38,H49
	K1,K2,K3	J22	H49
	K1,K2,K3	J31,J38	H3,H6
	K1,K2,K3		
	K1,K2,K3,O2,O4,M1,M2,R1,R2,R3,Cm1,Cm2,Cm3,Cm4,H6,H7,H8,H9,H10,Q20,Q21,Q22,Q23,Q24,Q25,Q26,Q27,Q28,Q29,Q30,Q31,Q32,T9,T10,T11,T12	J26	H58
		J26	H58
	K1,K2,K3		
	K1,K2,K3,O3,O4,R1,R2,R3,Cm1,Cm2,Cm3,Cm4,H1,H2,H3,H4,H5,H6,H7,H8,Q1,Q2,Q3,Q4,Q5,Q6,Q7,Q8,Q9,T8,S1,S2,S3,S4	J4	H17,H18
	K1,K2,K3		
	K1,K2,K3		

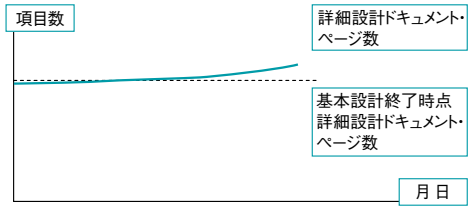
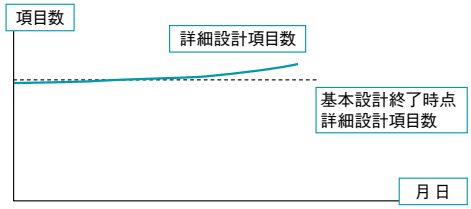
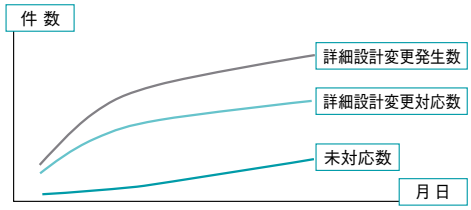
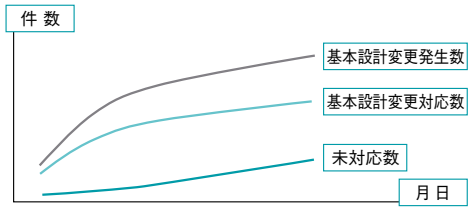
	ヒアリングシート	
⑫ 単体テスト結果の結合テスト、総合テストへの情報提供	H58	
●単体テストの結果変更された仕様、プログラムの情報を反映しているか	H58	
●単体テスト結果修正が完了していない仕様、プログラムの情報を反映しているか	H58	
⑬ 結合テスト、総合テストのシナリオ作成	H4,H5,H6,H33,H47,H48,H55,H56,H57	
●上流工程で確定した情報を反映したシナリオ作成を実施しているか ・要件定義で確定した情報 ・システム設計で確定した情報 ・ソフトウェア設計で確定した情報 ・単体テストの結果確定した情報	H4,H6,H33	
●ユーザーの視点を取り込むための、ユーザーの参画をお客様に要求したか	H5	
●ユーザーの視点で確認するためのシナリオ作成（操作性など）を実施したか	H5	
●性能面（負荷テストを含む）を確認するためのシナリオ作成を実施したか	H4,H57	
●コンティンジェンシー・プラン対応のシナリオ作成を実施したか	H47,H48,H56	
●運用を想定したシナリオの作成を実施したか	H5,H47,H56,H57	
●結合テスト、総合テストでの評価基準の作成を実施したか		
●結合テスト環境は、発生した障害を迅速に解析できることを含めてテストに十分なものが準備されているか	H55	
⑭ ソフトウェア設計、プログラミング、単体テスト状況に合わせたプロジェクト計画の見直し	H7,H24,H49	
●プロジェクトに影響する内容を確認し、プロジェクト計画書の見直しを実施したか ・スケジュール ・体制 ・設備 ・費用 ・プロジェクト運用ルール ・開発標準（ネーミン・グルール）	H7,H24,H49	
●見直し後のプロジェクト計画のレビュー・承認を実施したか	H7,H24,H49	
●計画の見直し結果をプロジェクト・メンバーへ周知したか ・プロジェクト運用ルール ・開発標準	H24	

	測定分析データー覧	(①～⑭が対処できなかったため発生した) 事例	ヒアリングシートから引用した 自動マッピングの結果
		J26	H58
		J26	H58
		J26	H58
	O2,H7,Q33,Q34,Q35,Q36,Q37,Q38,T12	J24,J25,J27,J35,J36,J38,J39,J53	H4,H5,H6,H33,H47,H48,H55,H56,H57
	Q33,Q34,Q35,Q36,Q37,Q38	J35,J38,J39	H4,H6,H33
	Q33,Q34,Q35,Q36,Q37,Q38	J36	H5
	Q33,Q34,Q35,Q36,Q37,Q38	J36	H5
	Q33,Q34,Q35,Q36,Q37,Q38	J25,J35	H4,H57
	Q33,Q34,Q35,Q36,Q37,Q38	J24,J27	H47,H48,H56
	Q33,Q34,Q35,Q36,Q37,Q38	J24,J25,J36	H5,H47,H56,H57
	Q33,Q34,Q35,Q36,Q37,Q38		
	Q33,Q34,T12	J53	H55
	Te2,K1,K2,K3,M1,M2,R1,R2,R3,Cm1, Cm2,Cm3,Cm4,C1,T9,Q4,Q10,Q11,Q16, Q28,Q29,Q30,Q31,S3	J22,J23,J33,J44	H7,H24,H49
	Te2,K1,K2,K3,M1,M2,R1,R2,R3,Cm1, Cm2,Cm3,Cm4,C1,T9,Q4,Q10,Q11,Q16, Q28,Q29,Q30,Q31,S3	J22,J23,J33,J44	H7,H24,H49
		J22,J23,J33,J44	H7,H24,H49
	Cm1,Cm2,Cm3	J33	H24

付録 5. 測定分析データー覧表

測定分析データー覧表（スコープ）

項目番号	知識エリア（主）	知識エリア（関連）	測定の目的	重点項目	利用者	
S1	スコープ		確定した機能の詳細設計の規模と変動を把握する （以下２項目をグループ化）		PM、 PMO	
S2	スコープ				PM、 PMO	
S3	スコープ		設計変更の面から ●詳細設計変更の規模と変動を把握する ●詳細設計変更への対応状況を把握する		PM、 PMO	
S4	スコープ		設計変更の面から ●基本設計変更の規模と変動を把握する ●基本設計変更への対応状況を把握する		PM、 PMO	

	導出尺度	収集者	導出尺度の見方、分かること
	●基本設計終了時点の詳細設計ドキュメント・ページ数 ●詳細設計ドキュメント・ページ数の推移 	詳細設計者	●基本設計終了時点に計画したドキュメント・ページ数よりも増加している場合、詳細以降の工程の作業工数、期間が増加し、 全体工期、コストを守れなくなる ●詳細設計ドキュメント・ページ数が急激に増加あるいは減少した場合、機能が変わった可能性がある。機能の確認が必要
	●基本設計終了時点の詳細設計項目数 ●詳細設計項目数の推移 	詳細設計者	●基本設計終了時点に計画した詳細設計項目数よりも増加している場合、詳細以降の工程の作業工数、期間が増加し、 全体工期、コストを守れなくなる ●詳細設計項目数が急激に増加あるいは減少した場合、機能が変わった可能性がある。機能の確認が必要
	●詳細設計変更発生数の推移 ●詳細設計変更対応数の推移 ●未対応数の推移 未対応数＝詳細設計変更発生数累計 －詳細設計変更対応数累計 	詳細設計者	●詳細設計変更発生数が多い、あるいは増加傾向にある場合、基本設計で機能が確定されていない可能性がある ●詳細設計変更発生数が急激に増加した場合、機能が変わった可能性がある。機能の確認が必要 ●詳細設計変更対応数が多いあるいは増加傾向にある場合、対応負荷・費用について顧客と合意が取れていない可能性がある ●未対応数が多い、増加傾向にある場合、その対応に工数と期間がとられ、詳細設計の工期を守れなくなる
	●基本設計変更発生数の推移 ●基本設計変更対応数の推移 ●未対応数の推移 未対応数＝基本設計変更発生数累計 －基本設計変更対応数累計 	基本設計者	●基本設計変更発生数が多い、あるいは増加傾向にある場合、要件定義が明確でない可能性がある ●基本設計変更発生数が急激に増加した場合、要件が変わった可能性がある。要件の確認が必要 ●基本設計変更対応数が多いあるいは増加傾向にある場合、対応負荷・費用について顧客と合意が取れていない可能性がある ●未対応数が多い、増加傾向にある場合、その対応（機能確定）に工数と期間が取られ、詳細設計に移れず工期を守れなくなる

測定分析データー覧表（タイム）

項目番号	知識エリア（主）	知識エリア（関連）	測定の目的	重点項目	利用者	
T1	タイム		詳細設計作業の進捗を把握し、計画との差異を確認する（以下４項目をグループ化）		PM、PMO	
T2	タイム				PM、PMO	
T3	タイム				PM、PMO	
T4	タイム				PM、PMO	
T5	タイム		詳細設計レビュー作業の進捗を把握し、計画との差異を確認する	★	PM、PMO	
T6	タイム	品質	詳細設計レビュー指摘件数を把握し、未対応数が残っていないか確認する		PM、PMO	
T7	タイム	品質	詳細設計変更に伴う詳細設計書改編作業の進捗を把握し、改編遅れがないか確認する		PM、PMO	

	導出尺度	収集者	導出尺度の見方、分かること
	詳細設計ドキュメント・ページ作成の推移	詳細設計者	●詳細設計の進捗が計画より遅れている場合、詳細設計を加速することにより、詳細設計が不十分となる可能性があり、単体テスト以降の工程において設計の見直しが発生して手戻りが生じ、コストの増加や、さらなるスケジュール遅れにつながるリスクがある ●詳細設計の進捗が計画より早い場合、新規システムの場合や担当者の経験が少ない場合などにおいては、設計すべき機能を見落としている可能性があり、単体テスト以降の工程で、設計漏れが発生したり、再検討を行ったりすることにより、品質の低下やスケジュール遅れにつながるリスクがある
	作成した詳細設計項目数の推移	詳細設計者	
	詳細設計ドキュメント作成の進捗率 進捗率＝完成したドキュメント・ページ数／計画ドキュメント・ページ数	詳細設計者	
	詳細設計項目の進捗率 進捗率＝設計完成した項目数／計画項目数	詳細設計者	
	詳細設計書のレビュー進捗率 ＝レビュー実績回数／レビュー計画回数 ＝レビュー実績時間／レビュー計画時間	詳細設計者	●詳細設計のレビューの回数や時間が計画より少ない場合、レビューが不十分な可能性があり、詳細設計の品質問題のため、単体テスト以降の工程において手戻りが発生し、コスト増加やスケジュール遅れにつながるリスクがある ●詳細設計のレビューの回数や時間が計画より多い場合、詳細設計の品質が元々悪く、レビューを行っても十分改善されていない可能性があり、単体テスト以降の工程で、手戻りが発生し、コスト増加やスケジュール遅れにつながるリスクがある ●いずれの場合も、レビューの状況（手法、かけた時間、参加した人、検出された誤りと修正、完了時の状況など）を確認すべきである
	●詳細設計書レビュー時指摘件数の推移 ●詳細設計書レビュー時指摘対応数の推移 ●詳細設計書レビュー時指摘未対応数の推移 未対応数＝指摘件数累計ー指摘対応数累計	詳細設計者	●詳細設計のレビューの回数や時間が計画より少ない場合、レビューが不十分な可能性があり、詳細設計の品質問題のため、手戻りが発生し、コスト増加やスケジュール遅れにつながるリスクがある ●詳細設計のレビューの回数や時間が計画より多い場合、詳細設計の品質が元々悪く、レビューを行っても十分改善されていない可能性があり、手戻りが発生し、コスト増加やスケジュール遅れにつながるリスクがある ●いずれの場合も、レビューの状況（手法、かけた時間、参加した人、検出された誤りと修正、完了時の状況など）を確認すべきである
	●詳細設計変更発生数の推移 ●詳細設計変更に伴う改編頁数の推移 ●詳細設計変更に伴う詳細設計書改編完了数の推移 ●改編未完数 改編未完数＝詳細設計変更発生数の累計ー改編完了数	詳細設計者	改編未完数がゼロに収束しなければ、スケジュール遅れにつながり、また、見切って次工程に移れば、品質の低下や手戻りにつながるリスクがある

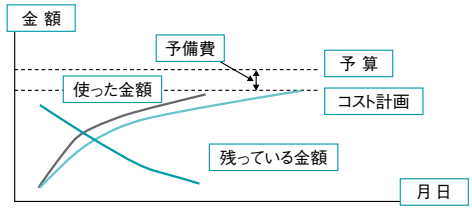
測定分析データー覧表(タイム)

項目番号	知識エリア(主)	知識エリア(関連)	測定の目的	重点項目	利用者	
T8	タイム	品質	基本設計変更に伴う基本設計書改編作業の進捗を把握し、改編遅れがないか確認する		PM、PMO	
T9	タイム		●クリティカル・パスの進捗を把握し、全体工程への影響がないか確認する ●クリティカル・パスが明確になっているかも見る		PM、PMO	
T10	タイム		マイルストーン達成の責任者が明確になっているか、的確に責任者がアサインされているかを見る		PM、PMO	
T11	タイム		プロジェクト推進に必要な環境が用意されているか確認する		PM、PMO	
T12	タイム				PM、PMO	

測定分析データー覧表(コスト)

項目番号	知識エリア(主)	知識エリア(関連)	測定の目的	重点項目	利用者	
C1	コスト		●コストの消化状況を見る ●コストが計画を越えていないかを見る ●当初計画にない対応を取る場合に、予算内で取れるコストを見る		PM、PMO	

	導出尺度	収集者	導出尺度の見方、分かること
	●基本設計変更発生数の推移 ●基本設計変更に伴う改編頁数の推移 ●基本設計変更に伴う基本設計書改編完了数の推移 ●改編未完数 改編未完数＝機能変更発生数の累計－改編完了数	詳細設計者	改編未完数がゼロに収束しなければ、スケジュール遅れにつながり、また、見切って次工程に移れば、品質の低下や手戻りにつながるリスクがある
	●クリティカル・パス明記の有無 ●クリティカル・パスにある詳細設計書、基本設計書作成の上記に記載の推移、進捗率	PM	●クリティカル・パスが明確になっていなければ、全体工程のスケジュール遅れの把握に手間取り、見逃して、対策が後手に回るリスクがある ●クリティカル・パスの進捗が計画より遅れると、プロジェクト全体のスケジュールが遅れ、工程確保のために人員や工数が増大するリスクがある
	●責任者のアサインの有無 （組織上の責任の取れる立場にある人のアサイン）	PM	●アサインされていないと、進捗監視、問題解決が遅れ、スケジュールが遅れるリスクがある。また、一貫性のある意思決定が行われず、品質が悪化するリスクがある
	開発環境の数 開発環境：ソフトウェア設計やプログラミングを行う環境	PM	開発環境数が少ないと、プログラム改修のための開発環境の調整・設定変更に時間がかかり、改修速度を遅らせている可能性がある
	テスト環境の数 テスト環境：ソフトウェア・テストを行う環境	PM	テスト環境数が少ないと、改修したプログラムのテストのための環境の調整・設定変更に時間がかかり、改修速度を遅らせている可能性がある

	導出尺度	収集者	導出尺度の見方、分かること
	●予算 ●使った金額 ●残っている金額 残っている金額＝予算－使った金額  	PM	●使った金額がコスト計画より多い場合、問題が発生し、当初計画外の対応を実施している可能性がある。あるいは計画が甘い可能性がある。計画を見直す必要がある ●使った金額がコスト計画より少ない場合、計画通りに要員・その他リソースの調達・投入ができていない可能性がある。あるいは計画が甘い可能性がある。計画を見直す必要がある ●当初の計画にない対応に必要な額が予備費内に収まらない場合、エスカレーションして追加予算の承認をとることが必要

測定分析データー覧表(品質)

項目番号	知識エリア(主)	知識エリア(関連)	測定の目的	重点項目	利用者	
Q1	品質		【保守性・移植性】 詳細設計プロセスにおける標準順守率を把握する		PM、 PMO	
Q2	品質		【信頼性】 詳細設計書レビューが有効に機能しているか把握する	★	PM、 PMO	
Q3	品質		【信頼性】 詳細設計書レビュー結果への対処が確実に行われているか把握する	★	PM、 PMO	
Q4	品質		【信頼性】 詳細設計書レビューの指摘事項への個別の対応だけでなく、共通的あるいは基本的な対策を実施しているかを確認する		PM、 PMO	
Q5	品質		【保守性・移植性】 詳細設計プロセスの品質を把握する		PM、 PMO	
Q6	品質		【保守性・移植性】 詳細設計プロセスの品質を把握する		PM、 PMO	
Q7	品質		【保守性・移植性】 詳細設計プロセスの保守能力を評価する		PM、 PMO	
Q8	品質		【信頼性】 詳細設計書の信頼性を把握する	★	PM、 PMO	
Q9	品質		【信頼性】 確定度合いに問題のある要件を特定する		PM、 PMO	

	導出尺度	収集者	導出尺度の見方、分かること
	詳細設計プロセスでの標準順守率 　標準順守率＝順守している標準項目数／順守すべき標準項目数 ここでの標準とは ●詳細設計書レビュー計画書 ●詳細設計書レビュー基準 ●詳細設計書フォーマット ●詳細設計書記入要領	詳細設計者 もしくは レビュー担当者	標準順守率が低い場合、要件の検討が不十分で必要な要件が網羅されていないなど、詳細設計書の品質が低い可能性がある
	●詳細設計書・レビュー時の指摘事項数 ●ドキュメントページ数当たりの指摘事項数 　＝レビュー時の指摘事項数／ページ数 ●ファンクション・ポイント数当たりの指摘事項数 　＝レビュー時の指摘事項数／ファンクション・ポイント数 レビューでは、標準に順守できていないことも指摘する	PM もしくは レビュー担当者	指摘事項総数、ドキュメント・ページ数当たりの指摘数あるいはファンクション・ポイント数当たりの指摘数が少ない場合、レビューが不十分で詳細設計書の品質が低い可能性がある
	レビューに基づいた ●詳細設計書の指摘事項修正数 ●詳細設計書の指摘事項未修正数 　未修正数＝レビュー時の指摘事項数累計 　　－指摘事項修正数累計 ●詳細設計書の指摘事項対応率 　対応率＝指摘事項修正数累計／レビュー時の指摘事項数累計	PM もしくは レビュー担当者	修正数が少ない、対応率が低い場合を詳細設計書の品質が低い
	詳細設計書の ●指摘事項分析の有無 ●指摘事項分析結果による対策実施の有無	PM	詳細設計書の指摘事項分析および対策の実施がされていない場合、詳細設計書の品質が低い可能性がある
	詳細設計工程での手戻り工数の分布 （平均、分散、個別、総計）	詳細設計者 もしくは レビュー担当者	●手戻り工数が多い場合、レビューの仕方が悪く、詳細設計書の品質が低い可能性がある ●レビュー方法を見直すことなどにより、効率の良いレビューとなり、生産性の向上が期待できる
	詳細設計工程での手戻り回数の分布 （平均、分散、個別、総計）	詳細設計者 もしくは レビュー担当者	●手戻り工数が多い場合、レビューの仕方が悪く、詳細設計書の品質が低い可能性がある ●レビュー方法を見直すことなどにより、効率の良いレビューとなり、生産性の向上が期待できる
	詳細設計書の修復までの時間 　修復までの時間＝レビュー時指摘事項作成日時 　　－レビュー時指摘事項対応日時	PM もしくは レビュー担当者	●修復までの時間が長い場合、詳細設計者のスキルレベルが低い可能性がある ●他の作業と競合するなど、担当者の作業負荷が高くなっている可能性がある
	ドキュメント・レビュー回数 　ドキュメント・レビュー回数＝チーム内レビュー回数 　＋プロジェクト・レビュー回数 　＋ユーザー・レビュー回数	PM もしくは レビュー担当者	●レビュー回数が少ない場合、詳細設計書の品質が低い可能性がある ●設計の複雑率、難易率を考慮してレビュー回数を確認する必要がある
	詳細設計工程での手戻り発生したコンポーネント群 ●コンポーネントごとの修正回数	詳細設計者	●修正回数の多い要件は、目的が明確でない、あるいは目的に適っていない可能性がある ●他の条件によって容易に変更される可能性も考えられるため、詳細設計工程終了後も注意しておく必要がある

測定分析データー覧表(品質)

項目番号	知識エリア(主)	知識エリア(関連)	測定の目的	重点項目	利用者	
Q10	品質	スコープ タイム	【効率性】 システムの目標となる性能値が設計されているかを把握する		PM、 PMO	
Q11	品質	スコープ	【機能性】 システムに要求された機能の実現率合いを把握する		PM、 PMO	
Q12	品質		【保守性・移植性】 基本設計プロセスにおける標準順守率を把握する (技術標準に準拠する標準項目数を基に評価する)		PM、 PMO	
Q13	品質	技術	【保守性・移植性】 基本設計書の信頼性を標準適合の率合いから把握する (設計内容が技術標準に準拠する成果物数を基に評価する)		PM、 PMO	
Q14	品質	スコープ タイム	【信頼性】 システムの目標となる障害回復時間が設計されているかを把握する		PM、 PMO	
Q15	品質	スコープ タイム	【機能性】 システムの目標となるセキュリテイの設計がされているかを把握する		PM、 PMO	

	導出尺度	収集者	導出尺度の見方、分かること
	性能設計進捗率 ●タスク進率率 = 完了したタスク数／タスク数 ●トランザクション設計進捗率＝設計されたトランザクション数／トランザクション数 （注）「設計されたトランザクション」には次に示す数値が具体化されていること •トランザクション別処理速率の見積値 •トランザクション別処理能力の見積値 •トランザクション別リソース使用率の見積値 •バッチ処理単位単位の制限時間 •バッチ処理単位の見積処理時間	基本設計者	●進捗率が低い場合、性能設計が十分行われていない可能性がある ●一部の性能設計が漏れた場合、それが原因でシステム全体としての性能要件を満たせなくなる可能性がある
	システム機能実現率 実現率＝実現システム機能数／要求システム機能数	①要件定義者 ②基本設計者	実現率が1未満の時は、要求機能が実現できない可能性がある。要求元との機能レベルでの調整が必要になる
	基本設計プロセスでの標準順守率 標準順守率＝順守している標準項目数／順守すべき標準の数 ここでの標準とは ●基本設計書レビュー計画書 ●基本設計書レビュー基準 ●基本設計書フォーマット ●基本設計書記入要領	基本設計者 もしくは レビュー担当者	標準順守率が低い場合、検討が不十分で基本設計書の品質が低い可能性がある
	基本設計書の標準適合率 適合率＝標準適合成果物数／成果物数	設計者 もしくはレビュー担当者	標準適合率が低い場合、設計内容が不十分であるなど、基本設計書の品質が低い可能性がある
	信頼性設計進捗率 進捗率＝設計された障害パターン数／設計すべき障害パターン数 （注）「設計された障害パターン」には次に示す数値が具体化されていること •障害回復対象機器数 •障害回復パターン数 •障害回復パターン別回復対象リソース数 •障害回復パターン別回復見積時間 •リソース別バックアップ見積時間 •リソース別バックアップ・タイミング •交代系機器数 •交代系機器別切り替え見積時間	基本設計者	●進捗率が低い場合、信頼性設計が根拠無く行われている可能性がある ●進捗率が1未満の場合、設計されていない障害パターンの発生によって、システムの信頼性が損なわれる可能性がある
	セキュリティ設計進捗率 進捗率＝設計されたセキュリティ機能数／設計すべきセキュリティ機能数 （注）「設計されたセキュリティ機能」には次に示す数値が具体化されていること •利用組織数 •利用組織別権限の階層数 •見積利用者数 •利用者の追加、削除、訂正の見積数 •システム化する機能数 •他システムとのインターフェース数	基本設計者	●進捗率が低い場合、セキュリティ設計が根拠無く行われている可能性がある ●進捗率が1未満の場合、システムにセキュリティ上の問題が発生する可能性がある

測定分析データー覧表(品質)

項目番号	知識エリア(主)	知識エリア(関連)	測定の目的	重点項目	利用者	
Q16	品質	スコープ タイム	【機能性】 システムに要求された機能の設計がされているかを把握する		PM、 PMO	
Q17	品質		【信頼性】 基本設計書レビューが有効に機能しているか把握する		PM、 PMO	
Q18	品質	タイム	【信頼性】 基本設計書レビュー結果への対処が確実に行われているか把握する		PM、 PMO	
Q19	品質	タイム	【信頼性】 基本設計書レビューの指摘事項への個別の対応だけでなく、共通的あるいは基本的な対策を実施しているかを確認する		PM、 PMO	
Q20	品質		【信頼性】 単体テスト計画書はレビューされているかを確認する		PM、 PMO	
Q21	品質		【信頼性】 単体テスト計画書はレビューされているかを確認する		PM、 PMO	
Q22	品質		【信頼性】 単体テスト・ケースはレビューされているかを確認する		PM、 PMO	
Q23	品質		【信頼性】 単体テスト・ケースはレビューされているかを確認する		PM、 PMO	

	導出尺度	収集者	導出尺度の見方、分かること
	●画面設計進捗率 　進捗率＝設計された画面数／画面数 ●帳票設計進捗率 　進捗率＝設計された帳票数／帳票数 ●インターフェース設計進捗率 　進捗率＝設計されたインターフェース数／総インターフェース数 ●機能設計進捗率 　進捗率＝設計された機能数／機能数 ●データ項目設計進捗率 　進捗率＝設計されたデータ項目数／データ項目数(注1) ●データベース設計進捗率 　進捗率＝設計されたテーブル数／テーブル数(注2) （注1）画面、帳票、インターフェースに現れるデータ項目 （注2）テーブルにはインデックスの設計、ビューの設計などの定義も含む	基本設計者	進捗率が低い場合、必要な設計対象が設計されていない可能性がある （ページ数での管理の裏付けとなる）
	●基本設計書レビュー時の指摘事項数 ●ドキュメント・ページ数当たりの指摘事項数 　＝レビュー時の指摘事項数／ページ数 ●ファンクション・ポイント数当たりの指摘事項数 　＝レビュー時の指摘事項数／ファンクション・ポイント数	PM もしくは レビュー担当者	指摘事項総数、ドキュメント・ページ数当たりの指摘数あるいはファンクション・ポイント数当たりの指摘数が少ない場合、レビューが不十分で基本設計書の品質が低い可能性がある
	レビューに基づいた ●基本設計書の指摘事項修正数 ●基本設計書の指摘事項未修正数 　未修正数＝レビュー時の指摘事項数累計－指摘事項修正数累計 ●基本設計書の指摘事項対応率 　対応率＝指摘事項修正数累計／レビュー時の指摘事項数累計	PM もしくは レビュー担当者	●修正数が少ない、対応率が低い場合は基本設計書の品質が低い
	基本設計書の ●指摘事項分析の有無 ●指摘事項分析結果による対策実施率 　対策実施率＝対策実施数／対策実施対象数	PM	基本設計書の指摘事項分析および対策の実施がされていない場合、基本設計書の品質が低い可能性がある
	単体テスト計画書のレビュー回数	PM	テスト計画書のレビューがされていない、あるいは回数が少ない場合、テストの網羅性が低い可能性がある （テスト網羅度が低い場合、信頼性保証できない）
	単体テスト計画のレビュー時指摘数	PM	●レビュー時の指摘数が少ない場合、レビュー不足の可能性がある ●レビュー時の指摘数が多いあるいは修正残数が多い場合、テストの網羅性が低い可能性がある
	単体テスト・ケースのレビュー回数	PM	●テスト・ケースのレビューがされていない、あるいは回数が少ない場合、テストの網羅性が不十分の可能性がある。また、異常系のテスト・ケース漏れの可能性がある ●テスト・ケースのレビューが実施されていないで、テスト・ケース作成完了とされている場合、進捗率の値が感覚的に報告されている可能性がある
	単体テスト・ケースのレビュー時指摘数	PM	●レビュー時の指摘数が少ない場合、レビュー不足の可能性がある ●レビュー時の指摘数が多いあるいは修正残数が多い場合、テスト・ケースの検討不十分あるいは重要性の認識があまい可能性がある

測定分析データー覧表(品質)

項目番号	知識エリア(主)	知識エリア(関連)	測定の目的	重点項目	利用者	
Q24	品質		【信頼性】 単体テスト・データはレビューされているかを確認する		PM、 PMO	
Q25	品質		【信頼性】 単体テスト・データはレビューされているかを確認する		PM、 PMO	
Q26	品質		【信頼性】 単体テスト・ケースは品質確認のために十分な数であるか確認する		PM、 PMO	
Q27	品質		【信頼性】 プログラムの品質を把握する		PM、 PMO	
Q28	品質		【信頼性】 プログラムの品質を把握する		PM、 PMO	
Q29	品質		【信頼性】 プログラムの品質を把握する		PM、 PMO	
Q30	品質		【信頼性】 プログラムの品質を把握する		PM、 PMO	
Q31	品質		【信頼性】 プログラムの品質を把握する		PM、 PMO	
Q32	品質		【信頼性】 プログラムの品質を把握する		PM、 PMO	

	導出尺度	収集者	導出尺度の見方、分かること
	単体テスト・データのレビュー回数	PM	テスト・データのレビューがされていない、あるいは回数が少ない場合、テスト・データがテスト内容に対して妥当でない可能性がある
	単体テスト・データのレビュー時指摘数	PM	●レビュー時の指摘数が少ない場合、レビュー不足の可能性がある ●レビュー時の指摘数が多いあるいは修正残数が多い場合、テスト・データの検討不十分あるいは重要性の認識があまい可能性がある
	単体テスト・ケースの密度	PM	●テスト・ケース密度が、基準値と比較して低い場合、テスト網羅度が不足し品質保証できない可能性がある
	手戻り単体テスト回数	PM	手戻り回数が多い場合、 ●リリース・ミスが多い、あるいは構成管理が不十分である可能性がある ●プログラム修正後の単体テストが不十分である可能性がある
	単体テスト検出バグ(現象)数	PM	●検出バグ数累計、欠陥密度、重要度別バグ発生比率からプログラム品質の程度が分かる ●発生推移から飽和数、飽和時期の概略が分かる。 ●バグ検出数の推移と修正完了数の推移から全数修正完了時期が見込める ●バグ修正完了数の推移からバグ修正のパワー不足、次工程のスケジュールへの影響が分かる ●バグ発生数に比べてプログラム修正数が異常に少ない場合、プログラム変更管理がされていない。あるいはプログラム変更内容のレビューがされていない可能性がある ●バグ票との対応をみて、バグごとのプログラム修正数が、0のものが多い場合、プログラム変更管理がされていない。あるいはプログラム変更内容のレビューがされていない可能性がある
	単体テスト検出バグ数(現象)の分析実施回数	PM	バグ分析が実施されていない、あるいは回数の少ない場合、バグの除去が行われていないか、不適切である可能性が高い (前提:あるバグが報告された場合、まずそのバグの分析を行った上で必要となる変更作業を実施するべきである)
	MTBF	PM	クリティカルな機能のMTBFが短いと品質が悪い(触れば直ぐ不良が発生する)
	MTTR(平均修正日数)	PM	MTTRが長い場合、 ●プログラム構造が悪く、修正に時間がかかっている可能性がある。すなわち設計が悪い、あるいは設計されていない(危険な状況)可能性が高い ●あるいは、開発チームの対応力不足の可能性がある
	コード・クローン	PM	●計画されたコード・クローンが少ない場合、カスタマイズされている部分が多く、品質が悪い可能性がある ●計画されていないコード・クローン量が多い場合は、プログラムの品質が悪いことが分かる ●問題があったときのコード・クローンによる影響範囲が分かる

測定分析データー覧表(品質)

項目番号	知識エリア(主)	知識エリア(関連)	測定の目的	重点項目	利用者	
Q33	品質		【信頼性】 結合テスト計画書はレビューされているかを確認する		PM、 PMO	
Q34	品質		【信頼性】 結合テスト計画書はレビューされているかを確認する		PM、 PMO	
Q35	品質		【信頼性】 結合テスト・ケースはレビューされているかを確認する		PM、 PMO	
Q36	品質		【信頼性】 結合テスト・ケースはレビューされているかを確認する		PM、 PMO	
Q37	品質		【信頼性】 結合テスト・データはレビューされているかを確認する		PM、 PMO	
Q38	品質		【信頼性】 結合テスト・データはレビューされているかを確認する		PM、 PMO	

	導出尺度	収集者	導出尺度の見方、分かること
	結合テスト計画書のレビュー回数	PM	テスト計画書のレビューがされていない、あるいは回数が少ない場合、テストの網羅性が低い可能性がある（テスト網羅度が低い場合、信頼性保証できない）
	結合テスト計画書のレビュー時指摘数	PM	●レビュー時の指摘数が少ない場合、レビュー不足の可能性がある ●レビュー時の指摘数が多いあるいは修正残数が多い場合、テストの網羅性が低い可能性がある
	結合テスト・ケースのレビュー回数	PM	●テスト・ケースのレビューがされていない、あるいは回数が少ない場合、テストの網羅性が不十分の可能性がある。また、異常系のテスト・ケース漏れの可能性がある ●テスト・ケースのレビューが実施されていないで、テスト・ケース作成完了とされている場合、進捗率の値が感覚的に報告されている可能性がある
	結合テスト・ケースのレビュー時指摘数	PM	●レビュー時の指摘数が少ない場合、レビュー不足の可能性がある ●レビュー時の指摘数が多いあるいは修正残数が多い場合、テスト・ケースの検討不十分あるいは重要性の認識があまい可能性がある
	結合テスト・データのレビュー回数	PM	テスト・データのレビューがされていない、あるいは回数が少ない場合、テスト・データがテスト内容に対して妥当でない可能性がある
	結合テスト・データのレビュー時指摘数	PM	●レビュー時の指摘数が少ない場合、レビュー不足の可能性がある ●レビュー時の指摘数が多いあるいは修正残数が多い場合、テスト・データの検討不十分あるいは重要性の認識があまい可能性がある

測定分析データー覧表(人的資源)

項目番号	知識エリア(主)	知識エリア(関連)	測定の目的	重点項目	利用者	
H1	人的資源		詳細設計工程における体制の強弱を測定する		PM、PMO	
H2	人的資源		詳細設計工程における体制の強弱を測定する		PM、PMO	
H3	人的資源		詳細設計工程における体制の強弱を測定する		PM、PMO	
H4	人的資源		詳細設計工程における体制の強弱を測定する		PM、PMO	
H5	人的資源		詳細設計工程における体制の強弱を測定する		PM、PMO	
H6	人的資源		プロジェクト管理体制の強弱を測定する		PMO	
H7	人的資源				PMO	
H8	人的資源				PMO	
H9	人的資源				PMO	
H10	人的資源				PMO	

	導出尺度	収集者	導出尺度の見方、分かること
	詳細設計工程における、詳細設計業務経験者率 詳細設計経験者数／詳細設計メンバー数	PM	詳細設計メンバーの経験、見識の深さが推定できる。経験者が不足している場合、詳細設計が遅延したり、不十分なまま次工程へ進んでしまう恐れがある
	詳細設計工程における、対象業務分野（小売業や保険業など）の経験者率 対象業務分野経験者数／詳細設計メンバー数	PM	詳細設計メンバーの対象業務に対する経験、見識の深さが推定できる。経験者が不足している場合、詳細設計が遅延したり、不十分なまま次工程へ進んでしまう恐れがある
	詳細設計工程における、対象プラットフォーム経験者率 対象プラットフォーム経験者数／詳細設計メンバー数	PM	詳細設計メンバーのプラットフォームに対する経験、見識の深さが推定できる。経験者が不足している場合、詳細設計が誤る恐れがある
	詳細設計メンバーのプロジェクト間兼務率 Σ紐付け率／詳細設計メンバー数	PM	詳細設計メンバーの兼務の状況が分かる。組織上人数がいても、能力が高くても、兼務率が高いと当該プロジェクトに対する十分な業務ができない
	詳細設計メンバーのプロジェクト間兼務率 1ー専任者数／詳細設計メンバー数	PM	詳細設計メンバーの兼務の状況が分かる。組織上人数がいても、能力が高くても、兼務率が高いと当該プロジェクトに対する十分な業務ができない
	プロジェクト・マネージャのＩＴＳＳレベル※ プロジェクト管理チーム・メンバーの人数、ＩＴＳＳレベル※ ※ITSSレベルとは、ITスキル標準（R）における職種「プロジェクトマネジメント」の中で、担当するシステムに該当する専門分野のレベルのことである	PM	プロジェクトの規模に較べて、ＰＭのＩＴＳＳレベルが低い、あるいはプロジェクト管理チーム・メンバー数が少ない、ＩＴＳＳレベルが低い場合、プロジェクト管理が十分になされてない可能性がある
	品質管理担当者の（プロジェクト内）専任者数	PM	品質管理の責任者が少ない、あるいは専任者がいない場合、品質管理、品質保証活動がが十分に行われず、バグ発生状況の把握不十分、バグ分析不足のため、バグ修正対応が不十分な可能性がある
	構成管理者（ライブラリアン）の（プロジェクト内）専任者数	PM	構成管理者が専任されていない場合、テスト中、頻繁に改修するプログラム、ドキュメントの構成管理ができていない可能性がある。その場合、改修ミス、テスト環境へのリリースミスを起こし、テストを進めることができなくなり進捗を遅らせることになる
	リリース管理者の（プロジェクト内）専任者数	PM	リリース管理者が専任されていない場合、改修プログラムのリリース・スケジュール、リリース事由が管理されていない可能性がある。その場合、テスト環境へのリースミスを起こし、テストを進めることができなくなり進捗を遅らせることになる
	テスト・開発環境管理者の（プロジェクト内）専任者数	PM	●テスト環境管理の専任者が少ないあるいは専任者がいない場合、テスト環境の利用スケジュール調整が不十分で、テスト環境の取り合いがそのつど発生し、その調整に時間を要し、テストの進捗を遅らせている可能性がある。また、テスト環境切り替えが計画通りできなくて、次のテストが実施できないため進捗を遅らせている可能性がある ●開発環境管理の専任者が少ないあるいは専任者がいない場合、開発環境の構成管理が不十分となり、プログラム修正ミスを起こしている可能性がある

測定分析データー覧表(コミュニケーション)

項目番号	知識エリア(主)	知識エリア(関連)	測定の目的	重点項目	利用者	
Cm1	コミュニケーション		各委託先、部門と十分にコミュニケーションが取れているか測定		PM、PMO	
Cm2	コミュニケーション		要員ごとの会議の出席状況より、コミュニケーション度合い(コミュニケーション不足の要員がいないかなど)を測定する		PM、PMO	
Cm3	コミュニケーション		会議開催ごとの出席状況より、コミュニケーション度合い(出席予定数に比べて、出席者数が少なくないかなど)を測定する		PM、PMO	
Cm4	コミュニケーション		組織、グループ、チーム間に跨る課題がないか否か測定する		PM、PMO	

測定分析データー覧表(リスク)

項目番号	知識エリア(主)	知識エリア(関連)	測定の目的	重点項目	利用者	
R1	リスク		リスク全体をスポットで測定し、トップマネジメントに活用		PM、PMO	
R2	リスク		リスクの時系列変化を測定し、問題の早期解決を図る(リスク監視コントロール)		PM、PMO	
R3	リスク		リスクの要因を具体的に測定する		PM、PMO	

測定分析データー覧表(モチベーション)

項目番号	知識エリア(主)	知識エリア(関連)	測定の目的	重点項目	利用者	
M1	モチベーション		要員ごとの予定労働時間に対する労働時間など勤務状況により、(極端な労働時間の増加など)モチベーションに影響を与える要因を測定する		PM、PMO	
M2	モチベーション		要員ごとの現業に対するモチベーションを測定する		PM	

	導出尺度	収集者	導出尺度の見方、分かること
	複数委託先(協力会社)部門の体制 ●階層数 ●会社数(部門数) ●開発拠点数(チームごとの作業場所数) ●報告書の提出数	PM、 PMO	数値が悪い場合、以下の事項が懸念される ①情報の共有が図れない、および問題点が職制に上らないため、問題発点の指摘、方針の徹底ができず、設計品質が低下する可能性がある ②設計品質の低下により、組織への依頼、納期、費用、システム品質の悪化、さらにメンバーのモチベーション低下が懸念される
	会議出席率1=(要員ごとの)会議出席回数／会議開催回数	PM、 PMO	
	会議出席率2=(会議開催ごとの)会議出席者数／会議出席予定者数	PM、 PMO	
	各組織、グループ、チーム間の課題の数と解決期間	PM、 PMO	

	導出尺度	収集者	導出尺度の見方、分かること
	重要度、発生確率、緊急度、対応策有無のマトリックス	PM、 PMO	発生しているリスクが大きい場合、プロジェクトが失敗する可能性がある
	一定期間における各リスクの発生度合い	PM、 PMO	
	リスク要因とリスク度	PM、 PMO	

	導出尺度	収集者	導出尺度の見方、分かること
	(要員ごとの) 労働時間(残業時間)／予定労働時間(予定残業時間)	PM、 PMO	①プロジェクト・メンバーの疲弊感、不達成感により、プロジェクト・ルールを順守しなくなり、チームワークが乱れる恐れあり ②報告、問題提起が消極的になり、問題がプロジェクトの中で潜在化する。その結果、プロジェクト失敗の恐れあり ③退社、転職などの人的流動で、ノウハウ流出し、知的財産の損失の恐れあり
	(要員ごとの) ●進捗 ●課題、問題と解決期間 ●報告書などの提出率	PM、 PMO	

測定分析データー覧表(組織)

項目番号	知識エリア(主)	知識エリア(関連)	測定の目的	重点項目	利用者	導出尺度	
01	組織	品質	詳細設計書が組織としてレビューされているか測定する		PM、PMO	詳細設計書のレビュー達成率 ＝詳細設計書のレビュー実施回数／詳細設計書のレビュー計画回数 詳細設計書のメンバー参加率 ＝詳細設計書の参加実績者数／詳細設計書の参加計画者数	
02	組織		組織で定めたプロセス標準に基づいて作業を行っているか測定する		PM、PMO	プロセス順守率 ＝実践しているプラクティス数／全プロセスエリア内で定義されたプラクティス数 ※カバーすべきプロセスエリアは組織やプロジェクトによって異なる	
03	組織		組織として効率的に作業を行っているか測定する		PM、PMO	生産性の推移 ＝作成ドキュメント頁数／作成工数	
04	組織	品質	組織として構成管理が標準通り行われているか測定する		PM、PMO	ドキュメントのチェックイン／アウト回数の推移	
05	組織		作業環境の充足度からプロジェクトの置かれている環境を把握する		PM、PMO	作業環境に対して組織標準で定められた要件数とそのうちで充足された要件の数	

	ベース尺度(左の導出尺度を求めるための測定データ)	ベース尺度ID	尺度の収集元・方法	収集者	導出尺度の見方、分かること
	①詳細設計書のレビュー実績回数 ②詳細設計書のレビュー計画回数 ③詳細設計書のレビュー参加実績者数 ④詳細設計書のレビュー参加計画者数	①BO001 ②BO002 ③BO003 ④BO004	詳細設計書のレビュー時に確認する	PM	詳細設計書が組織として計画通りにレビューされていない場合、プロジェクトの体制、スケジュールに問題がある可能性がある
	①実践しているプラクティス数 ②全プロセスエリア内で定義されたプラクティス数	①BO005 ②BO006	定期的に(月1回など)	PM	組織で定めたプロセスを順守できていない場合、重要な作業の見落とし、無駄な作業の実施、成果物管理の失敗、品質管理の失敗、進捗状況把握の失敗…などのリスクが発生する
	①作成ドキュメント頁数 ②作成工数	①BO007 ②BO008	当該工程完了時	PM	生産性の推移を監視することにより、プロジェクトの疲弊を検知できる ①生産性が最初からあがらない場合は初期要求に問題ある可能性がある ②生産性が低下する場合、無理な要求の追加や要求のいつまでも明確なっていない可能性がある
	①ドキュメント・チェックイン回数 ②ドキュメント・チェックアウト回数	①BO009 ②BO010	定期的(週次)に、構成管理システム(CVSなど)から①、②を収集する	PM	①チェックイン、アウトが工程終盤にのみ行われている場合、構成管理が標準通りに行われていない可能性がある ②工程終盤で特定のドキュメントのチェックイン、アウトの回数が他のドキュメントに較べてチェックイン、アウトの回数が多い場合、要件、機能の変更が多く発生し、要件、機能が確定していない可能性がある
	①作業環境要件数 ②作業環境要件充足数	①BO011 ②BO012	プロジェクト立ち上げ時および基本設計終了時点	PM	充足度が低い場合、要求への対応が十分でなく、メンバーのモチベーション低下につながる可能性がある

測定分析データー覧表(課題管理)

項目番号	知識エリア(主)	知識エリア(関連)	測定の目的	重点項目	利用者	
K1	課題管理		課題発生数、対応数を測定し、課題完了状況を把握する		PM、PMO	
K2	課題管理		未解決課題の影響範囲を測定・推定する		PM、PMO	
K3	課題管理		未解決課題の緊急度を測定・推定する		PM、PMO	

測定分析データー覧表(技術)

項目番号	知識エリア(主)	知識エリア(関連)	測定の目的	重点項目	利用者	
Te1	技術	品質	キャパシティ・プラニング実施の有無を測定する		PM、PMO	
Te2	技術		開発に必要な技術に対する要員ごと、または組織として利用経験を観ることにより、利用技術の新規性を測定する		PM、PMO	

	導出尺度	ベース尺度 (左の導出尺度を求めるための測定データ)	導出尺度の見方、分かること
	●課題発生数の推移 ●課題対応数の推移、 ●未解決課題数の推移 未解決課題数＝課題発生数累計－課題対応数累計	①課題発生数 ②課題対応数	●未解決課題数が増加傾向にある場合は、工期遅延のリスクが増加する ●課題が増加しているにもかかわらず、対応数の累計に変化がない場合、課題への対応が進んでいないということなので、解決の困難な課題の存在や、課題対応の適切な担当割ができていない可能性がある
	影響度別未解決課題数 ＝影響度別課題発生数累計－影響度別課題対応数累計	①影響度別課題発生数 ②影響度別課題対応数	影響度の高い未解決課題がある場合、工期遅延のリスクが増加する
	緊急度別未解決課題数 ＝緊急度別課題発生数累計－緊急度別課題対応数累計	①緊急度別課題発生数 ②緊急度別課題対応数	緊急度の高い未解決課題がある場合、工期遅延のリスクが増加する

	導出尺度	収集者	導出尺度の見方、分かること
	キャパシティ・プラニング実施の有無	PM	●キャパシティ・プランニングが実施されていない場合、著しい工期遅延、製品品質の低下の可能性がある ●キャパシティ・プラニングを実施する要員や支援部署がプロジェクトにアサインされていない可能性がある
	利用経験の有無	PM	利用経験なしの場合、著しい工期遅延、製品品質の低下の可能性がある

参考文献

〈全　般〉
［1］　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 ,“ITプロジェクトの「見える化」上流工程編”,日経BP社, 2007年5月.
［2］　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 ,“ITプロジェクトの「見える化」下流工程編”,日経BP社, 2006年6月.
［3］　長岡良蔵,“プロジェクト「見える化」とは—下流工程編—”,SEC journal,No.6, p28, 2006年5月.
［4］　樋口 登,“ITプロジェクトの「見える化」—上流工程編—”,SEC journal, No.10, p18, 2007年5月.
［5］　長岡良蔵,“プロジェクト「見える化」—下流工程編—”, SEC Forum 2006予稿, 2006年6月.
［6］　長岡良蔵,“プロジェクト「見える化」—上流工程編—”, SEC Forum 2007予稿, 2007年6月.
［7］　長岡良蔵,“プロジェクト「見える化」—上流工程編—”, 2007年度PM学会予稿, 2007年9月.
〈第2章〉
［8］　ISO/IEC 12207 “Software Life Cycle Processes”, 1997.
［9］　IEEE/EIA 12207.2, “ Guide for ISO/IEC 12207, Standard for Information Technology-Software life cycle processes -Implementation Considerations”.
［10］ JIS X 0160 : ソフトウェアライフサイクルプロセス(追補1および2), 2007年10月.
［11］ 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 ,“共通フレーム2007 ～経営者,業務部門が参画するシステム開発および取引のために～ ”,オーム社, 2007年10月.
〈第3章〉
［12］ Ohtaka, Kasahara, “Risk Management and Business Performance”, IPMA, June.2004
［13］ Ohtaka, “P2SM and Business Performance”, ProMAC2004, IPMA2004, June.2004 October.2004.
［14］ Project Management Institute,“A guide to the Project Management Body of Knowledge. 3rd edition.”, Project Management Institute, 2004.
［15］ 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 ,“ソフトウェアエンジニアリングの実践～先進ソフトウェア開発プロジェクトの記録～ ”, 日経BP社, 2007年11月.
［16］ 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 ,“プロセス改善ナビゲーションガイド～なぜなに編”,オーム社, 2007年3月.
［17］ 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 ,“プロセス改善ナビゲーションガイド～プロセス診断活用編”,オーム社, 2007年4月.
〈第4章〉
［18］ 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 ,“ソフトウェア改良開発見積りガイドブック～既存システムがある場合の開発～ ”, オーム社, 2007年10月.
［19］ 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 ,“ソフトウェア開発見積りガイドブック～ IT顧客とベンダーにおける定量的見積りの実現～ ”,オーム社, 2007年3月.
［20］ 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 ,“経営者が参画する要求品質の確保～超上流から攻めるIT化の勘どころ～ ”, オーム社, 2007年7月.

執筆者（敬称略・五十音順）

プロジェクト見える化部会委員（◎：主査、○：WGリーダー）

秋山 雅俊	株式会社NTTデータユニバーシティ
飯田 元	奈良先端科学技術大学院大学
大川 繁喜	日本電気株式会社
大高 浩	株式会社DTS
亀田 康雄	NTTソフトウェア株式会社
川原 雅宏	株式会社アイネス
木根 秀隆	日立ソフトウェアエンジニアリング株式会社
栗田 存	株式会社クロスリンク・コンサルティング
○桑原 秀昌	TIS株式会社
香村 求	株式会社システムSWAT
芝田 晃	三菱電機株式会社
長岡 満夫	IPA/SEC（株式会社NTTデータ）
◎長岡 良蔵	IPA/SEC（新日鉄ソリューションズ株式会社）
○西川 廣	新日鉄ソリューションズ株式会社
拝原 正人	株式会社クロスリンク・コンサルティング
樋口 登	IPA/SEC（日本電気株式会社）
福地 豊	株式会社日立製作所
水野 真澄	株式会社日立システムアンドサービス
神谷 芳樹	IPA/SEC
吉川 宏幸	IPA/SEC（T＆D情報システム株式会社）

レビュー協力

葭谷 努	TIS株式会社
荒井 勝	松下電器産業株式会社
大鹿 幸一郎	株式会社日本総合研究所
後藤 雅史	インテック株式会社
渡辺 道広	NTTソフトウェア株式会社

ITプロジェクトの「見える化」 中流工程編

2008年10月6日	初版第1刷発行
著作／監修	独立行政法人 情報処理推進機構（IPA） ソフトウェア・エンジニアリング・センター（SEC）
発行人	中島 久弥
発行	日経BP社
発売	日経BP出版センター 〒108-8646　東京都港区白金1-17-3
表紙デザイン	成田 美由喜（日経BPクリエーティブ）
制作	日経BPクリエーティブ
印刷・製本	大日本印刷

ISBN978-4-8222-6230-3

日経BP社
Nikkei Business Publications,Inc.
〒108-8646　東京都港区白金1-17-3