

品質データが出る前に診断する

—マスタスケジュールから品質管理欠陥を読み解く 5 つの診断軸

ソフトウェア品質シンポジウム 2026 経験論文

上田 昌平

第 1 章 はじめに

1.1 問題の所在

ソフトウェア品質管理に関する議論の多くは、「品質データが出てからどう対応するか」に集中してきた。レビュー指摘密度、テスト密度、バグ密度といった定量指標は、いずれも工程実行後に得られるデータであり、これらを用いた品質判定・改善活動は、いわば症状への対症療法に過ぎない。

しかし、大規模 SI 案件における品質問題の本質は、個別バグの数や密度ではなく、品質管理プロセスの構造的不在にある。品質を作り込み、評価し、判定する仕組みが計画に組み込まれていなければ、いかに精緻な品質指標を持ち出しても事後対応の域を出ない。病魔は Day 1 から潜んでいる。

1.2 本論文の主張と独自性

本論文は、品質管理の構造的欠陥が、品質データが出る前にマスタスケジュールから診断できることを主張する。マスタスケジュールはプロジェクト計画の中心成果物であり、品質管理プロセスが構造として埋め込まれているか否かは計画段階で判別可能である。筆者は 20 年超にわたる大規模 SI 実務経験から 5 つの典型的な構造欠陥パターンを抽出し、これを「5 つの診断軸」として体系化した。

既存の品質管理研究との差別化ポイントは 3 点である。第一に、プロジェクト外部からの構造診断という視座である。ここでの「外部」とは組織の内外ではなく、対象プロジェクトの利害から独立した構造的視点を指す（6.2 節で詳述）。第二に、計画段階での介入である。マスタスケジュールという計画成果物を診断対象とし、品質データの発生を待たずに介入を可能にする。第三に、5 軸の連鎖関係の提示である。上流の意思決定の質が下流の構造として現れる連鎖の理解が、診断と介入の優先順位付けを可能にする。

1.3 用語の定義

マスタスケジュール：プロジェクト全体の工程・タスク・期間・依存関係を表現した計画成果物。WBS をガントチャート等の時間軸表現で可視化したもの。

マネジメントスケジュール：マスタスケジュール内に組み込まれる、PM・PMO・品質管理担当が実施するマネジメント活動（品質評価、ゲートレビュー、進捗評価、意思決定タスク等）の計画。多くの現場ではそもそも存在しない。

構造診断：定量データではなく、計画成果物の構造（タスクの存在・配置・順序関係）から品質管理体制の健全性を判定する診断手法。

第 2 章 マスタスケジュールは「物語」である

2.1 スケジュール=物語という視座

マスタスケジュールは単なるタスクリストではなく、プロジェクトが勝利に至るストーリーの可視化物である。優れたマスタスケジュールには、Vision（何を、いつ、どのような状態でリリースするかという最終的な勝利の姿）、Plot（勝利に至る論理的かつ具体的なステップ）、Voice（ステークホルダー間の合意と統合、各関係者の役割・責任・依存関係の反映）の 3 要素が織り込まれている。これら 3 要素が織り込まれた物語として読めない計画は、その時点で構造が破綻しており、必ず構造的欠陥が存在する。

ストーリーの歪みを読み解く 5 つの診断軸

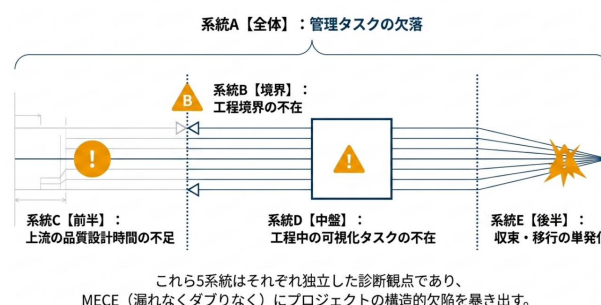


図 2-1 マスタスケジュールと 5 つの診断軸の位置づけ

2.2 物語の歪みから診断軸へ

「物語が読めない」という状態は主観ではなく、3 つの典型形態として現れる。第一に、根拠なき次工程への移行（前工程の品質を証明する仕組みなしに次工程が始まる）。第二に、意思決定タスクの空白（テスト戦略・移行戦略・サブシステム結合戦略等の重要な意思決定が「決める作業」としてタスク化されていない）。第三に、完了定義の曖昧さ（「終わった」の判定基準が事前に定義されず、完了報告が形骸化する）。

これらの歪みは、より構造的に 5 つの診断軸として分類できる。第 3 章で詳述する 5 軸は、「物語が読めない」という主観的判断を、再現可能な診断基準へと変換する装置である。

第 3 章 5 つの診断軸

3.1 5 軸の俯瞰

マスタスケジュールから読み取れる構造的欠陥を、5 つの診断軸（軸 A～E）として提示する（表 3-1）。各軸について定義・スケジュール上の見え方・中核となる病理を示した後、3.7 節で診断チェックポイントを一覧化し、3.8 節で 5 軸の関係構造を論じる。

軸	軸の名称	中核となる病理
A	管理タスクの欠落	マネジメントを単なる「会議体運営」と誤認
B	工程境界の不在	前工程の品質を無批判に信じ切る
C	上流の品質設計時間の不足	「決める」ことを工程として認識していない
D	工程中の可視化タスクの不在	進捗を「タスク完了」と捉えている
E	収束プロセスの段階性欠如	「収束は一発で決まる」という致命的錯覚

表 3-1 5 軸の俯瞰

3.2 軸 A：管理タスクの欠落

定義：PM・PMO・品質管理担当が実施すべきマネジメントタスクがスケジュール上に存在せず、開発タスク（作業系）だけで構成されている状態。
スケジュール上の見え方：ガントチャートの PM/PMO 行が空白、または単発イベント（キックオフ・報告会）のみ。「品質評価」「ゲートレビュー」「進捗評価」等のタスクが見当たらず、マネジメント層の関与が「会議出席」としてしか描かれていない。

病理：マネジメントとは会議体の設定ではなく、意思決定を駆動する戦略策定タスクの設計である。テスト計画書や移行計画書を「テスト直前に作る作業」と捉えている現場では、この軸で構造的欠陥が検出される。

3.3 軸 B：工程境界の不在

定義：工程と工程の間に Go/No-Go 判定が描かれておらず、前工程の品質を担保せずに次工程へ流れる構造。
スケジュール上の見え方：要件定義→基本設計→詳細設計→実装が滑らかに接続され、境界に判定マイルストーンが置かれていない。工程が重なって並走し、前工程の終了基準が曖昧。

病理：品質の証明なき次工程への移行は、前工程への盲信に過ぎない。スケジュール遵守が品質より優先される構造が固定化し、No-Go 判定を出せる仕組みが存在しない。Conditional Go という中間判定がないため、判定が「全 Go」に偏る。

3.4 軸 C：上流の品質設計時間の不足

定義：要件定義・基本設計工程の期間が短すぎる、もしくはレビュー・修正のサイクルが組み込まれておらず、下流の品質を作り込むべき上流に時間が確保されていない状態。
スケジュール上の見え方：設計工程に対し実装・テスト工程の比率が極端に高い。「設計」が単一バーで描かれ、ドラフト→レビュー→FIX の内部構造が見えない。

病理：「作る」ことだけを工程とみなし、「決める」ことを工程として認識していない。意思決定のリードタイムが計画に組み込まれず、下流工程に入ってから「あれを決めていなかった」が頻発する。

3.5 軸 D：工程中の可視化タスクの不在

定義：工程の途中で進捗・品質を可視化するタスクが描かれておらず、工程末尾の完了報告まで状態が分からない構造。
スケジュール上の見え方：各工程内に中間チェックポイントが置かれていない。週次・月次の品質指標モニタリングがタスク化されておらず、「いま何が起きているか」を把握する仕組みがない。

病理：完了報告を待つ構造である。継続的な可視化と検証ポイントが物理的に存在しないため、工程末尾でしか実態が見えず、対策を打つタイミングを失う。「完了報告で初めて遅れが発覚する」が常態化する。

3.6 軸 E：収束プロセスの段階性欠如

定義：複数の並列展開された開発単位を一度に収束させようとする、あるいは収束イベントを単発で済ませようとする

る構造。本来段階的に統合・収束すべきプロセスが圧縮されている。 **スケジュール上の見え方**：時間軸方向の単発化（テスト工程が1サイクルのみ、移行リハーサルが1回のみ、リリース直前にしか収束タスクがない）と、並列軸方向の単発化（複数サブシステムのビッグバン結合、結合順序（疎→密）の未設計、チーム横断統合タスクの不在）として現れる。

病理：単発化の根本原因は、収束プロセスを「作業」ではなく「イベント」として捉えていることにある。巨大システムは、時間軸・並列軸双方での段階的な構造的収束なしには統合できない。

3.7 診断チェックポイントの一覧

各軸を、実際のマスタスケジュールのどの箇所からどう判定するかを、確認観点として表 3-2 に集約する。診断は本表を用いたチェックリスト形式で実施でき、診断者個人の経験や勘に依存しない。

軸	確認観点（マスタスケジュール上でのチェックポイント）
A：管理タスクの欠落	・PM/PMO 行に継続的な管理タスクが描かれているか ・テスト計画・移行計画・品質管理計画書の策定タスクが上流工程内（要件定義～基本設計）にスケジュール化されているか ・個別計画書の策定タイミング・承認プロセスが明示されているか ・工程ごとの品質評価タスクが計画されているか ・上位レビュー（部長・経営層）のタイミングが事前定義されているか
B：工程境界の不在	・各工程の終了に判定マイルストーンが設置されているか ・Go/Conditional Go/No-Go の3段階判定が定義されているか ・判定基準（プロダクト品質×プロセス品質）が事前に明示されているか ・不合格時の戻り工程・再判定タイミングが描かれているか ・レビューの独立性（PJ 外部の目）が確保されているか
C：上流の品質設計時間の不足	・設計工程内にレビュー期間・修正期間が明示的に確保されているか ・設計工程の長さが下流工程との比率で妥当か ・テスト戦略・移行戦略が上流工程で検討されているか ・サブシステム間の結合戦略（順序・粒度）が上流の設計で意思決定されているか ・レビュー指摘密度のベンチマーク評価タイミングが組み込まれているか
D：工程中の可視化タスクの不在	・工程内に中間進捗評価タスク（30%/50%/70%地点等）が設置されているか ・品質指標（レビュー指摘密度、バグ密度等）の定期モニタリングが計画されているか ・課題管理・リスク管理のレビュータイミングが描かれているか ・EVM 等の定量指標による進捗評価が工程内に組み込まれているか ・予兆検知から対策までのリードタイムが見込まれているか
E：収束プロセスの段階性欠如	・テスト工程が複数サイクルで計画されているか ・移行リハーサルが複数回（本番想定で2～3回）計画されているか ・サブシステム結合が段階的（疎結合→密結合の順）に設計されているか ・結合の各段階で品質評価タスクが組み込まれているか ・チーム間の統合インターフェース確認タスクが上流～中流に分散しているか ・リリース前の安定化期間が確保されているか

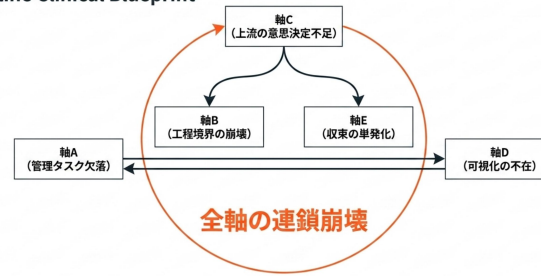
表 3-2 5 軸の診断チェックポイント一覧

3.8 5 軸の関係構造

5 軸は独立ではなく連鎖関係を持つ。軸 C（上流の品質設計時間不足）は軸 B（工程境界の不在）と軸 E（収束の単発化）として下流に現れ、軸 A（管理タスクの欠落）は軸 D（可視化の不在）として現れる。軸 C と軸 A が重なると全軸の連鎖崩壊に至る。本フレームの中核命題は「上流の意思決定の質が、必然的に下流のスケジュール構造として露呈する」というものである。

特に軸 C が軸 E に現れる関係は中核的洞察である。サブシステム結合戦略の欠如は、設計工程での意思決定不足が下流のスケジュール構造に露呈した症例として読める。逆に軸 E の構造を観察することで、軸 C の意思決定の質を遡って評価できる。この連鎖の理解は介入優先順位の決定に直結する。下流の症状（軸 E）への対症療法ではなく、上流の根本原因（軸 C）への構造介入が、結果的に最も効率的な処方となる場合が多い。

Pristine Clinical Blueprint



5つの軸は独立していない。上流の意思決定の質が、物理的かつ必然的に下流の構造的ジオメトリとして露呈する。

© Hontobooks Ltd

図 3-1 5 軸の連鎖構造（上流の意思決定が下流の構造として露呈する）

第 4 章 処方パッケージ

4.1 処方の全体観

5 つの診断軸には、それぞれ対応する処方が存在する。本論文は計 15 の処方を提示する（表 4-1）。これらは単発のテクニックではなく、マネジメントスケジュールに恒久的に組み込むべき構造的コンポーネントとして設計されている。

番号	処方名	主対応軸	フォーカス
1	マネジメントスケジュールの新規策定	A	★
2	個別計画書（テスト/移行/品質管理）の上流策定タスク化	A	
3	PM/PMO 役割分担の明確化（検討系/作業系の分離）	A	
4	中間品質可視化マイルストーンの設置	D	★
5	品質指標の定期モニタリング体制構築	D	
6	課題・リスク管理の定期レビュー設計	D	
7	ゲートレビュー2 層構造の導入	B	★
8	上流意思決定タスクの独立スケジュール化	C	★
9	Go/Conditional Go/No-Go 判定基準の事前定義	B	
10	設計工程内のレビュー・修正サイクル組み込み	C	
11	レビューアの独立性確保（PJ 外部レビュー）	B	
12	サブシステム結合の段階設計（疎→密の順序設計）	E	★
13	テスト工程の複数サイクル化	E	
14	移行リハーサル of 複数回実施	E	★
15	リリース前安定化期間の確保	E	

表 4-1 15 処方の一覧と軸対応

このうち介入効果が特に大きい 6 つをフォーカス処方（★）とし、3 グループに整理する。G1「管理と可視化の修復」（処方 1・4、軸 A・D、全工程貫通レイヤー）、G2「上流設計と境界の構築」（処方 7・8、軸 B・C、上流レイヤー）、G3「段階的収束の設計」（処方 12・14、軸 E、下流レイヤー）である。3 グループは介入レイヤーに対応しており、診断結果に応じてどのレイヤーから着手するかを決定できる。

4.2 フォーカス処方の要点

処方 1（マネジメントスケジュールの新規策定）：マネジメントタスクをスケジュール上の独立行として可視化し、PM・PMO の活動を「会議出席」から「計画されたタスク群」へ転換する。「会議で十分」という抵抗には、マネジメント未定義による下流での手戻りコストを示して対処する。

処方 4（中間品質可視化マイルストーンの設置）：各工程の 30%/50%/70%地点等に中間品質評価を設置し、工程末尾まで状態が見えない構造を解消する。最初の中間評価で隠れた遅延・品質課題が表面化し、早期対策が工程末尾の混乱を減らす。

処方 7（ゲートレビュー2 層構造の導入）：Layer 1（PJ 内ゲート：PM/PMO/TL による品質確認）と Layer 2（マネジメントレビュー：組織的承認）の 2 層で、Go/Conditional Go/No-Go の 3 段階判定を制度化する。Conditional Go という中間判定が「全 Go 偏重」と「No-Go を出せない構造」の双方を解消する。

処方 8（上流意思決定タスクの独立スケジュール化）：テスト戦略・移行戦略・サブシステム結合戦略等の意思決定

を、独立したタスク群として要件定義～基本設計工程に組み込む。下流での「あれを決めていなかった」を大幅に減少させる。

処方 12（サブシステム結合の段階設計）：結合を疎結合→密結合の順に段階化し、各段階に品質評価タスクと戻り基準を組み込む。ビッグバン結合での障害切り分けコストを回避し、障害をサブシステム単位で切り分け可能にする。

処方 14（移行リハーサルの複数回実施）：移行リハーサルを最低 2 回（理想は 3 回）計画し、手順検証→時間制約下での実施可能性検証→本番想定 of 全工程通し、と段階的に熟成させる。本番での障害発生率を大幅に低下させる。

4.3 処方の段階的導入

15 処方の一斉導入は現実的ではなく、診断結果に応じた段階導入が必要である。一般には、他の処方の前提条件となる G1（管理と可視化の修復）を第一段階とし、介入時点が上流工程内であれば G2 を、下流工程に入っていれば G3 を優先する。下流での介入では G2 は次フェーズへの教訓として記録する。ただしこの順序は一般論であり、実案件では診断結果と介入時点の状況に応じて柔軟に組み替える。

第 5 章 適用事例

5.1 案件概要

本フレームを適用した案件は、大規模金融系基幹システムのリプレースプロジェクトである。匿名化のため詳細な業種・組織は伏せる（表 5-1）。

項目	内容
業種・性格	金融系基幹システム。オンプレミス旧システムからパッケージベース新システムへのリプレース。カスタム開発比率 60%超
規模・期間	約 1,000 人月、約 20 億円。当初 2 年計画、延長で約 3 年
体制	顧客（事業会社）、システム子会社、現行ベンダ、切替先ベンダ、周辺システムベンダの多層構造
筆者の関与	外部 IT マネジメント・コンサルタント（火消し依頼にて参画）。筆者は週 2 日、コンサルメンバ 3 名は常駐、関与期間約 1 年
介入時点	当初予定のシステムテスト工程が停滞し、リスケジュール直後
ミッション	上流ドキュメント品質の確認と FIX、システムテストのリプラン、リリース遅延を 1 年以内に圧縮

表 5-1 案件プロフィール

5.2 介入時点の診断結果

介入時点で 5 軸診断を実施した結果、5 軸すべてで構造的欠陥が検出された（表 5-2）。判定凡例：○（十分）／△（内容不十分）／×（項目自体が存在しない）。

軸	判定	主な所見
A：管理タスクの欠落	×	マネジメントタスクが一切描かれておらず、ゲートレビューも不在。テスト計画書・移行計画書の策定タイミングも不明確。
B：工程境界の不在	×	儀式的なゲートのイベントは存在したが判定基準が曖昧で、実態は上位者への形式承認。プロダクト品質・プロセス品質の評価は行われていなかった。
C：上流の品質設計時間の不足	×	パッケージ活用を理由に要件定義と基本設計が重複し、実質は設計時間の圧縮。レビューサイクルは未設計で、一発レビューで次工程へ進む想定。
D：工程中の可視化タスクの不在	×	進捗・品質の定量可視化が弱く「順調」「遅延」といった定性報告が中心。工程末尾近くで「終わらない」ことが判明するパターンが常態化。
E：収束プロセスの段階性欠如	×	実態は段階結合だがスケジュール上は単一の結合イベント。移行リハーサル環境をシステムテストに活用する統合設計も欠落。

表 5-2 5 軸診断結果サマリ

特に注目すべきは軸 C と軸 E の連鎖である。「パッケージ活用だから設計は軽くていい」という前提の下で上流工程が圧縮され、その結果、下流の統合・収束が単発イベント化するという典型的連鎖が観察された。3.8 節の中核命題が本案件で実証されたかたちである。



図 5-1 介入時点の 5 軸診断結果（5 軸すべてで構造的欠陥を検出）

5.3 診断から介入への第一手：バックスケジューリング

診断結果を処方実施へ繋ぐ第一の介入手段は、リリース期限（動かせない制約）を起点に、リリース前に完了すべきマイルストーン、その前提となる工程完了時期、さらに前提となる中間マイルストーンへと後方から制約を逆算し、現行計画との不整合を洗い出すバックスケジューリング分析であった。

この分析により、現行のマスタスケジュールが「形式的には描かれているが、後方からの制約を満たしていない」ことが組織横断で可視化された。後方から必要タイミングを明確化する過程で、マネジメントスケジュールの必要性が自然に浮かび上がり、処方 1 導入への合意形成を容易にした。バックスケジューリングは 15 処方には含まれないが、抽象的な診断結果（「軸 A に構造的欠陥がある」等）を「計画がこのままでは期限を守れない」という具体的な計画問題へと翻訳する橋渡しであり、この翻訳が組織横断の合意形成と処方導入の意思決定を加速する。開発ベンダから表明された「元々の契約と進め方が変わる」という抵抗に対しても、計画見直しが契約変更の問題ではなく契約履行のための必要措置であることの合意形成に、この翻訳機能が有効に働いた。

5.4 導入した処方と効果

処方導入に先立ち、PM と PMO の役割再設計を行った。PM が実質的に設計リーダーを兼務し「先を見据える」動きができない構造に対し、PM は現時点の設計・実装に関する意思決定に集中し、PMO（筆者らコンサルチーム）がバックスケジューリング、マネジメントスケジュール策定、ゲートレビュー設計等の「将来への備え」に注力する分担とした。これは処方 3 の実践例であり、後続する処方導入の前提条件となった。

処方 1（マネジメントスケジュールの新規策定）：バックスケジューリング分析を踏まえ、品質評価・ゲートレビュー準備・意思決定タスク等を工程ごとに配置した。本案件ではマネジメントスケジュールの存在自体がゼロからの創出であったため、導入効果は顕著に現れた。

処方 12（サブシステム結合の段階設計）：実態として行われていた段階的結合をスケジュール上の明示的な段階として可視化し、各段階での品質評価ポイントを定義してビッグバン結合を回避した。スケジュール表現と実態の乖離解消が、関係者間の共通認識形成に重要であった。

処方 4・7・8・14 についても部分的に導入し、明確に「やらなかった」処方はなく 5 軸すべてへの介入を試みた。効果として、定量面ではリリース遅延を 1 年以内に圧縮し当初ミッションを達成した。定性面では、感覚的な品質議論（「順調」「大丈夫そう」）から事前定義された指標に基づく品質確認への転換が組織内に定着し、関係者間の共通認識形成を容易にした。

第 6 章 考察と結論

6.1 考察：事後対応から構造設計へ

本フレームは、品質管理を事後対応から構造設計へとパラダイムシフトさせる実践知である。品質指標は症状を観測する道具であり、構造を診断する道具ではない。診断対象を「品質データ」から「マスタスケジュールの構造」に移すことで、介入タイミングを工程実行後から計画段階へ前倒しする。

本フレームは PMBOK に代わるものではなく、補完する実践知である。PMBOK が「何をすべきか（What）」を体系的に教えるのに対し、本フレームは「それが構造的に機能しているか」を診断するツールであり、両者は相互補完関係にある。

6.2 適用条件と限界

本フレームの有効性は次の 3 条件下で確認されている。第一に大規模 SI 案件（100 人月超、複数チーム・複数ベンダ、複数工程）であること。第二に計画段階または上流工程での介入であること。下流工程からの介入では軸 C への完全な構造介入は困難であり、本事例でも部分的対応に留まった。第三に、プロジェクトの利害から独立した視点からの診断であること。

第三の条件である「外部性」の本質は、組織の内外ではなく、対象プロジェクトの利害からの独立性と、計画を構造として読む視点にある。プロジェクト内部の PM・PMO は自プロジェクトの計画書を客観視することが構造的に困難であり、PM 本人による自己診断は構造的盲点が残る。一方で、この役割は外部コンサルタントに限らず、横串組織の品質管理部門、全社 PMO、他部門の経験豊富な PM 等、独立性を確保できる立場であれば組織内でも担い得る。ただし実効性の確保には、外部視点を受け入れる組織文化と権威づけ（本部長オーダー等）が伴うことが望ましい。適用条件を外れる場合の留意点も整理しておく。小規模・単一チーム案件では 5 軸のフル診断はオーバースペックであり、軸 A・軸 B に絞った簡易診断が現実的である。アジャイル文脈では「物語」の単位が異なるため直接適用はできず、イテレーション計画への翻訳が今後の検討課題である。また本事例は危機介入での適用だったが、定常的な計画レビューやゲートレビューにおける観点集としての予防的利用も可能である。多ステークホルダー構造では、診断結果の合意形成自体に時間を要する点に留意が必要である。

6.3 品質指標運用に関する実践知

フレーム適用を通じて、品質指標運用に関する実践知を得た。従来のベンチマーク運用（上限値・下限値による判定）は、組織に品質指標の実績蓄積がなければ実践的な意味をほとんど持たない。このような組織では、まず「数値を保有する状態を作る」ことが優先される。機能・設計者・規模・件数・推移といった複数次元のデータを、いつのタイミングで取得するかを事前に計画する。これは処方 1 と処方 4 の延長として実装可能である。ベンチマーク判定は、データが蓄積された後のフェーズに移行する。

事例案件で観察された「感覚から数値への議論の転換」は、この「数値の保有」を確立したことの結果でもある。実績蓄積のない組織における実効的なアプローチとして、本実践知はフレームを補完する。

6.4 今後の展望と結論

今後の展開として、第一に診断の半自動化がある。構想としては 2 段階を想定している。まず表 3-2 の確認観点をチェックリストとして形式知化し、診断の標準化と診断者育成に用いる段階。次いで、スケジュールデータ（タスクの名称・配置・依存関係）を機械的に解析し、5 軸ごとの構造的欠陥候補を自動抽出する段階である。最終判断は人間の診断者が行うが、初期スクリーニングの自動化により診断の適用コストを下げられる。第二に、複数組織・複数案件への適用蓄積による再現可能性の本格的検証と適用条件の精緻化、第三に PMO 人材育成手法への展開である。

本論文は、品質管理の構造的欠陥がマスタスケジュールから事前診断可能であることを主張し、5 つの診断軸と 15 の処方を体系化して提示した。大規模金融系基幹システムリプレース案件への適用を通じて有効性を確認し、診断→翻訳（バックスケジュールリング）→処方の 3 段プロセスと、実績蓄積のない組織における品質指標運用の実践知を提示した。品質を「データが出てから対応する」のではなく「構造で設計する」ためのツールとして、本フレームが品質管理実務に貢献することを期待する。

参考文献

- [1] PMI. PMBOK Guide 第 7 版.
- [2] SQuBOK 策定部会編. ソフトウェア品質知識体系ガイド. オーム社.
- [3] 情報処理推進機構. ソフトウェア開発分析データ集 2022.
- [4] 日本情報システム・ユーザー協会. ソフトウェア・メトリクス調査 2025.
- [5] 三菱電機. PJ 失敗 15 事例×51 問題事象の分析. SQiP 2025 (A2-2) .
- [6] QC コンパス論文. テーラリング不全の先行研究. SQiP 2025 (A3-3) .