

欠陥構造の可視化によるトラブルの早期発見と未然防止

～欠陥を中心に置くことで可能となるQAのコト作り～

Visualizing Defective Structure for Early Detection and Prevention of Troubles

～Koto making QA made possible by placing around the defect～

主査：細川 宣啓（日本アイ・ビー・エム株式会社）

副主査：永田 敦（ソニー株式会社）

リーダー：早崎 伸二（株式会社 リンクレア）

研究員：木下 良介（株式会社 リンクレア）

研究概要

我々の研究の目的は「何故トラブルはなくなるのか？なぜ組織は同じ失敗を繰り返すのか？」という問いの答えを見付け出し、トラブルの早期発見と未然防止の仕組み作りに対して貢献することである。この目的を達成するための手段として、「問題構造MAP」と名付けた可視化フレームを考案し、問題視すべきことと、トラブルを引き起こす欠陥構造の可視化を行った。そして、この可視化フレームを用いて過去のトラブル情報の見える化を行い、「プロジェクト健康診断問診票」という名前で組織内への展開を行った。その結果、QAが取組める五つのコト作り（品質向上施策）が可能となり、プロジェクトの自然治癒力と品質監査品質が高まることを確認できた。このことは、トラブル情報を組織内に展開する仕組みを構築することが「欠陥を中心に置いた品質向上施策を可能とし、トラブルの早期発見と未然防止に繋がる」ということを示したと考える。

Abstract

The purpose of our research is to find the answer of "Why troubles don't get lost? Why do organizations repeat the same mistakes as past trouble projects?" It must be contributed to the mechanism about early detection and prevention of trouble projects. To achieve this, first, we purpose the visible framework named "Problem Structure Map", followed by visualization of the defect structure. Next, to visualize and socialize the past trouble project information, we expanded the framework into issue named "Project Checkup Questionnaire". Finally, we could confirm that this makes possible to make five quality assurances experiences (Quality Improvement Measure) which leads to increase self-recovering ability and improves quality of projects. This trouble framework and checkup questionnaire enables projects to establish a system to be deployed with its quality improvement measures around the defects.

1. はじめに

本論文ではエンタープライズ系でソフトウェア開発や保守を請負う（契約形態ではない）場合に発生するトラブルに論点を当てる。そのため、成果物に埋め込まれる問題（以降、成果物問題と略記する）だけでなく、プロジェクトで発生するトラブルに繋がる問題（以降、プロジェクト問題と略記する）についても取り扱う。また、対象とするトラブルは図1に示す範囲とし、不良を意味する用語として「障害」は現象表現、「欠陥」は原因表現という定義で使用する。

- | | |
|------------|-----------------|
| ➤ メンバー離脱 | ➤ 本番誤操作 |
| ➤ 取引停止 | ➤ 品質起因での無償業務代行 |
| ➤ プロジェクト中断 | ➤ Goalが見えない障害対応 |
| ➤ コスト超過 | ➤ 品質強化要請 |
| ➤ 納期遅延 | ➤ 本番システム障害 |

図1. トラブルプロジェクトの定義

我々は、飯塚悦功。旧東京大学名誉教授が著書「原因分析」^[1]で問題解決において重要

第7分科会（RCAグループ）

だと述べられている「問題を問題と感じる感受性・意識を持つこと」と「問題発生の構造を明らかにすること」が、トラブル発生の問題を解き明かすために必要な本質的な課題であると捉えて検討を開始した。その検討の中で、まず以下のことを行なった。

- ・トラブルの構造の表現として、
「プロジェクト問題の捉え方」と「成果物問題の因果関係」を構造で表現
- ・「見逃してはいけない失敗の種類」と「根本原因の種類」を具体化

その成果として、欠陥構造を可視化するフレームを考案し、「問題構造MAP」と名付けた。

次に、過去のトラブル情報をこのフレームに当てはめて文章表現することで、問題視すべき事象と根本原因の可視化を行ない、その結果（プロジェクト健康診断問診票と命名）を組織内へ展開することで、以下の「品質向上施策（コト作り）」を実施可能とした。

- ①品質意識の底上げ
- ②根本原因分析の定着化による成果物欠陥混入の未然防止
- ③プロジェクト内での自己改善を推進（自然治癒力の向上）
- ④品質監査者（以降QAと略記する）が行う監査品質の向上
- ⑤新たなトラブル情報を組織内展開する仕組みの構築

そして、上記「コト作り」がトラブルの早期発見と未然予防に繋がることを確認した。

本論文の貢献は、「トラブル情報を組織内に展開（移転）するための欠陥構造を具体化し、QAが取組み可能な五つのコト作り（上記①～⑤）を考案したこと」であると考ええる。

尚、当論文の構成は、2章にて問題を提起し、3章にて考案した欠陥構造の考え方を示し、4章にてその活用例を示す。その上で、5章にて可能となる品質向上施策について解説し、そして、6章で展開結果の評価を述べ、7章にて展望を論じた上で、8章にてまとめを行う。

2. 考案にあたり問題として捉えたこと

本番障害は誰もが問題視し発注元は原因の説明と再発防止を求めて来る。しかし、同じ障害が単体テスト段階で検出された場合は、問題視せずに個人の責任範疇と考えて原因分析や予防策を打たない。また、問題視したとしても根本原因まで追究せずに障害対応等を問題対応だと思い込み、再発する問題の対処に追われているプロジェクトも散見される。このような成果物問題をプロジェクト問題に置換えても同様であり、トラブル時と平常時では同じ事象に対しての捉え方が異なると考える。また同じく、問題の事後対応に追われている状況が少なくない。どちらの問題であれ、問題を問題として認識することがトラブル減少へ向けての出発点であり、根本原因の究明が必要である。さらに、有効な対応策を打つためには、「プロジェクトが自力で改善する力を付けること」が望ましい姿であると考えるが、QAが、現場の自力改善の向上まで踏み込んでいないのが現状である。

次に、QAが問題と認識して指摘した内容に対して、PLが是正しないケースについて考察する。このケースは、QA指摘内容をPLが問題として認識しない場合であり、指摘内容は無視され改善は行われない。この状況は、指摘内容の重症度や放置するとどのような事態が引き起こされるかについての認識が異なるために引き起こされると考える。

また、トラブルが起こるたびに報告書は作成されるが、活用出来ていない状況がある。この状況はトラブル情報を知識化していないがために流通できない状況であると認識する。

これらの状況を鑑み、我々は、「以下の五つをQAが問題視すべきこと」と捉えた。

- (1) 問題として捉えるべきことを問題として認識しない結果、何の対処もなされない
- (2) 問題視した事についても根本原因を解決せずにいるため、同じ問題が再発する
- (3) プロジェクトの自己治癒力の向上に対してQAは何の手も打てていない
- (4) プロジェクトの品質監査に於いて、QA指摘の内容が無視され是正されない
- (5) 同じ失敗を繰り返さないための仕組みを構築していないがために、組織内で起こったトラブルから失敗を学べず、プロジェクトは失敗を繰り返す

3. 考案した可視化フレーム「問題構造MAP」

我々は、問題視すべきことの可視化を構造と共に表現可能な可視化フレームを考案し、「問題構造MAP」と命名した（図2参照）。以降でこの可視化フレームを解説する。

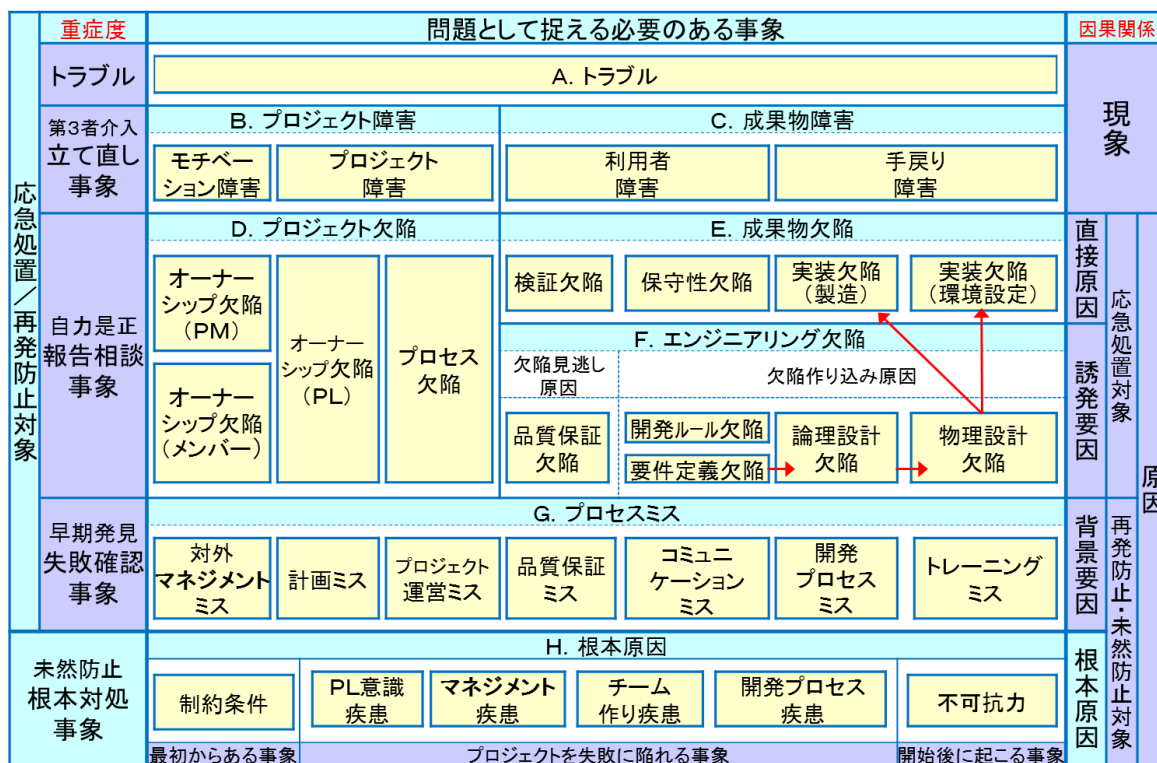


図2. 問題構造MAP

3.1 問題として認識すべきことを具体化してID化

問題として捉える必要のある事象（以降トラブル事象と表現する）を、個人差のある感受性に頼らない形で当事者が問題認識する方法の解決策として、トラブル事象を具体的に理解出来る表現で文章にし、当フレームの該当箇所へマッピングすることを想定した。また、識別を容易にするためのID化も想定した（付録4，5へ掲載した活用例を参照）。

3.2 トラブル情報の可視化方法

我々の主張は、問題は時間の経過とともに事象を変えるのだから、ある瞬間の事象だけを伝えるのではなく移り変わっていく事象の全てを伝えることで、初めて相手に伝わるという考え方である。拠って、トラブル情報の可視化方法として、プロジェクト問題は重症度が進むにつれて変わる事象を全て文章化し、成果物問題は因果関係のある事象を全て文章化することとした。

(1) 問題構造の捉え方

トラブル事象は図3に示す構造で捉えた。トラブルの元となる根本原因が存在すると何らかの失敗（プロセスミス）を引き起こす。そして、この失敗を放置するとプロジェクト問題か、または成果物問題を作り込み、その問題が悪化するとト

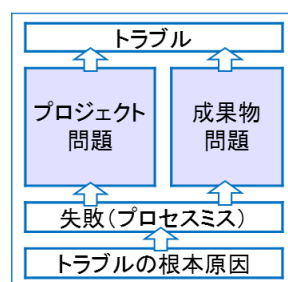


図3. 問題構造の概念

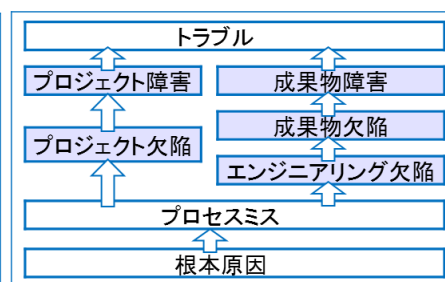


図4. 問題構造の具体化

ラブルへと繋がるという捉え方である。尚、プロジェクト問題と成果物問題は異なる問題構造として捉えた（図4を参照）ため、別々に論じることとする。

第7分科会（RCAグループ）

(2) プロジェクト問題の可視化

プロジェクト問題は、ヒューマンに関する事象とそれ以外の事象を切り分けた上で、重症度が進むにつれて変化する状態を可視化することとした（図2の「B.プロジェクト障害」と「D.プロジェクト欠陥」欄を参照）。具体的な重症度の可視化としては、立て直しの主体がPLのままで良いかと、報告相談の要否と、そして事象確認の要否という3つの観点で捉えることとした。責任感の強いPL程、問題を認識した場合に自分で解決しようとする。その結果、対処が後手に回り問題を更に悪化させてしまうことがある。また、相談さえしていれば回避できた事態や、早い段階でトラブルの芽に気付いていれば簡単に対処できた事例もある。このような事態になることを防止するために、重症度の見出しは行動に結び付く表現とした（図2の重症度の列を参照）。

さらに、階層表現を取ったことにより、今起きている事象を放置するとどのような結果を引き起こすかを視覚的に把握できるようになり「何をすべきか」という思考に繋がることも考えた。図5に具体例を示す。

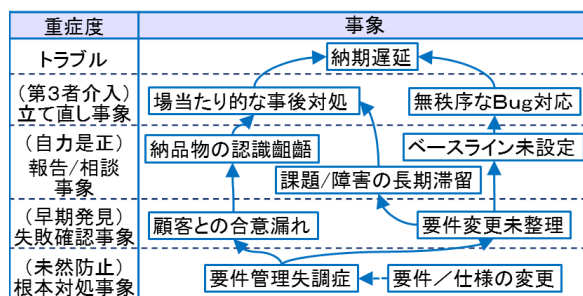


図5. プロジェクトの問題の具体例

(3) 成果物問題の可視化

成果物問題は、検証に関する事象、保守に関する事象、実装に関する事象の3種類に切り分けて因果関係のある事象を全て捉えることとした（図6と同時に活用例である付録5を参照）。検証に関する事象は、例えばレビュー観点漏れがあると、テスト項目漏れに繋がるという因果関係がある。保守に関する事象の例では、デッドコードの発生は保守性を下げるが必ずしも障害に繋がるとは言えないという因果関係がある。次に、実装に関する事象の因果関係であるが、成果物は段階的に詳細化する形で開発するため、上流工程で作り込まれた欠陥は、見逃してしまうと必然的に下流工程へ流れ、最終的にはコードに欠陥が作り込まれて障害として現れる。これらの因果関係を図の上下の位置関係で現すこととした。図7に具体例を示す。（誘発要因がない関係は図6に示した）

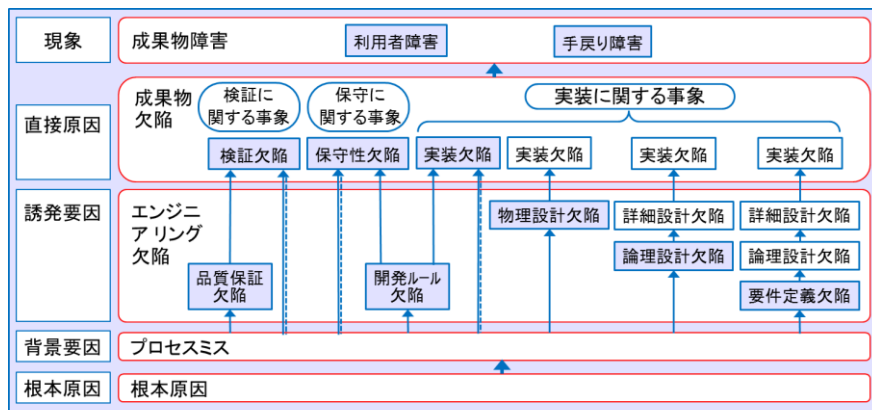


図6. 成果物問題の構造

【線表記】↑: 誘発要因がある関係、↑: 誘発要因がない関係

図7に具体例を示す。（誘発要因がない関係は図6に示した）

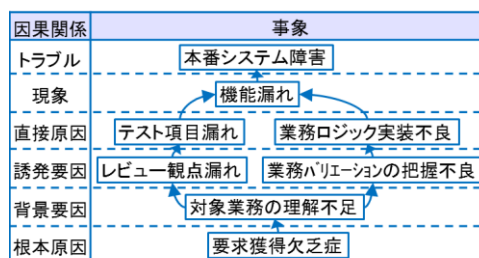


図7. 成果物問題の具体例

尚、実装に関する事象についてののみ、以前から社内展開している（以降、この展開内容を旧モデルと表記する）。また、旧モデルの展開の結果、事象（欠陥分類）の網羅性とその構造について、組織内での妥当性確認（プロジェクトで使えるかの確認）は完了し、この内容を用いて根本原因分析の手法を展開している。（付録1参照）。

次に、構造を明示するための見出しについて解説する。見出しは因果関係を連想できる表現を行い（図6を参照。直接原因以外は、主要な原因という意味合いで要因と表現）、直接原因と誘発要因の対処は、応急処置であることを明示し、エンジニアリング欠陥は、欠陥見逃し原因と欠陥作り込み原因の両方を意識させるように表現した（図2参照）。

第7分科会（RCAグループ）

3.3 プロセスミスの可視化（図2の「G. プロセスミス」欄を参照）

プロジェクト問題、または成果物問題を引き起こす原因をプロセスミスと位置付けた。プロセスミスと名付けたのは、その事象を失敗と捉えることでQAは指摘をし易くなり、PLは指摘を受け入れやすくなると考えたためである。また、事象の分類名は、その事象の捉え方を示す名称とすることで、見逃してはいけない失敗の種類の具体化を行った。

3.4 根本原因の可視化

プロセスミスを引き起こす原因を根本原因と位置付け、根本原因の事象名に対して「考え方を正す」表現をとることとした。その理由は、稲盛和夫氏が著書「生き方」^[2]で述べられている「考え方次第で仕事の結果は決まってしまう」と同様の考え方（付録2参照）であり、「PLの考え方次第でプロジェクトの結果は変わる」という考え方である。よって、トラブルを誘発する考え方に対して、改善事項（治療対象）であることを明確に示すために「病名」を付けることとした。また、自分が病気を患っていると認識すると自ら直そうとする気持ちも働くと考えた。

(1) 根本原因の分類

成果物問題の欠陥作り込み原因にのみ限定した分析（旧モデル）を社内展開した時点では、佐藤允一帝京大学名誉教授が著書「問題構造学入門」^[2]で述べられている考え方をそのまま踏襲し、[①制約条件、②手段、③活動、④外乱]の4分類を使って展開した（付録3参照）が、納得感を持たないメンバーも居たため、以下の分類に改訂した。

[①最初からある事象、②プロジェクトを失敗に陥れる事象、③開始後に起こる事象]

(2) 病名を付ける

「①最初からある事象」は制約条件、「③開始後に起こる事象」は不可抗力と捉えたため、考え方を正す対象は、「②プロジェクトを失敗に陥れる事象」と考え、この分類にのみ、病名を付けた。具体的には、事象の分類名に「疾患」という接尾語を付け、根本原因の種類を具体化した（図2の「H. 根本原因」欄を参照）。また、事象に付ける病名としては、ある事柄が十分でない場合は「～欠乏症」、全く欠如している場合は「～失調症」、過剰なことが問題となる場合は「～過剰症」、自信過剰が問題となる場合は「～過信症」とした。（活用例である付録4、5の根本原因部分を参照）

4. 問題構造MAPの活用例「プロジェクト健康診断問診票」

過去のトラブル情報を「問題構造MAP」に当てはめて文章表現することで、問題視すべき事象と根本原因を可視化した活用事例である「プロジェクト健康診断問診票」を以下で解説する。尚、付録4～6に拡大版（左半分と右半分）と全体イメージを掲載した。

(1) 収集範囲 : ソフトウェアの開発や保守を請負う場合に発生したトラブル事象

(367 事象を収集. 360 個を超えたので全方位を包含できたと考えている)

(2) 収集元情報 : 過去2年間に引き起こされたトラブルの報告書

(3) 記載手順

① 報告書からトラブル事象を抽出し、対応する重症度の適切な分類に記載する

② 記載したトラブル事象に紐着く他のトラブル事象を洗い出し、同様に記載する
もしくは、紐着くトラブル事象を新たに想定して同様に記載する

尚、「成果物問題」の事象については、検証済みのため、そのまま使えると想定する。
また、根本原因の事象については、雛形（作成元）として使えると想定する。

(4) 組織内展開時の工夫

① ネーミングは、QA監査の形式張ったイメージを和らげ、かつ必要性を感じさせる名前が良いと考え「プロジェクト健康診断問診票」と命名

② 展開フォームは、視覚的に全体を俯瞰出来る形が全体理解と個々の部分の位置付け理解に繋がると考え、1枚で表現することを重要視して作成

第7分科会（RCAグループ）

5. 欠陥を中心に置いた五つの品質向上施策

トラブル情報を「問題構造MAP」に当てはめて組織内展開することで、取組みが可能となる「品質向上施策」を以下に提案する。

5.1 品質意識の底上げ

可視化したトラブル情報を組織内へ展開するだけで以下の効果が出ると考える。

- ① 問題として捉える必要のある事象の認識が一致する
- ② 起こった事象の重症度の認識が一致する
- ③ 問題を放置した場合にどのように悪化するかの共通認識ができる
- ④ 共通の言葉で根本原因を特定できる

5.2 根本原因分析の定着化による成果物欠陥混入の未然防止

現象から根本原因を導き出すことは難しいことである。この難易度の高い分析を簡単に行えるようにしないと定着化は進まないと考え、一覧から特定するだけで根本原因分析が可能な手法（簡単RCA）を考案した。以下に分析手順を述べる。

- ① 検出した障害から原因（直接原因）を、[E. 成果物欠陥]の一覧から特定する
- ② 直接原因から、その原因（誘発原因）を[F. エンジニアリング欠陥]の一覧から特定する（直接原因が製造時のミスであれば、欠陥作り込みの誘発要因はない）
- ③ 直接原因、または誘発原因から背景要因を[G. プロセスミス]の一覧から特定する
- ④ 背景要因から根本原因を[H. 根本原因]の一覧から特定する

尚、分析は必ず「検証欠陥」と「作り込んだ欠陥」の両方を特定するルールとした。

次に分析自体の考え方であるが、分析行為はアクション（改善）にまで繋がられないと、「やらされ感」が発生し形骸化を引き起こすと考えている。しかし、「問題構造MAP」は、背景要因を網羅的に洗い出せるフレームであるため、根本原因まで「原因と結果の連鎖」を繋がられる。そして、根本原因には病名を付けていることで、「正すべき考え方」に辿り着く。つまり、アクションに繋がるため、形骸化は起こりにくいと考える。

5.3 自己改善の推進（プロジェクトの自然治癒力を高める）

プロジェクト内での自己診断ツールとしての使用や、チーム作りでの認識共有ツールとしての使用を組織内のプロセスに組み込むことで、以下のような自己改善が進むと考える。

- ① プロジェクト問題も、一覧からトラブル事象を選び出す形で分析が進められるため、独自での原因分析が可能になる
- ② 病気の存在を知ることによって自分が病気を患っていると認識し、自ら直そうとする
- ③ プロジェクト内でトラブル事象に対するフィードバックが可能になる

5.4 監査品質の向上

QAがプロジェクト監査に利用することにより、以下の効果が出ると考える

- ① プロセスミスの一覧をチェックリストに利用し、早期発見漏れを防止できる
- ② 一覧上の事象を問題として指摘するため、PLとの認識齟齬がなくなる

更に、指摘内容の重症度の認識や放置するとどのような事態が引き起こされるかの認識が同じとなるため、以下のことも可能となると考える。

- ③ 欠陥を特定した上で、欠陥を中心ににおいて改善方針やその優先度の議論

5.5 新たなトラブル情報を組織内展開する仕組みの構築

同じ失敗を繰り返さない仕組みの構築として、以下が可能になると考える。

- ① トラブルで発生した新たな事象は一覧に追加することで展開できる。

6. 展開結果の評価

プロジェクト健康診断問診票（以降問診票と略記する）を展開することがトラブルの早期発見や未然防止に有効であるかを確認することを目的に、アンケート形式で有効性の評価を行った。現時点では、トラブル低減効果までは評価できないため、説明会を開催した上で、表7に示すアンケートを実施し、問診票に対しての認識データを収集し、その結果で評価を行った。尚、対象者は、技術者総勢53名で、内訳はPM／PL32名、QA14名、メンバー7名である。

表1. アンケート内容		
確認概要	No	内容
品質意識の底上げ	RQ1	「問題として捉える必要のある事象」を以下の[重症度]欄の切り分け、[第3者介入]立て直し事象、[自力是正]報告/相談事象、[早期発見]失敗確認事象に従って分類したことにより、各々の事象に対する重症度の認識合せが出来たと思うか？
	RQ2	問題として捉える必要のある事象を放置すると「プロジェクトがどのように悪化するか」の理解が深まったか？
自己改善の推進	RQ3	成果物欠陥の予防を行う場合、[因果関係]欄の[直接原因]や[誘発原因]で原因追究を終わらずに、[根本原因]まで追究した上で対策を講じないと、再発防止や未然防止に繋がらないということについて納得感を持てたか？
	RQ4	根本原因分析が一覧から選び出すこととなるため、「分析難易度が下がる」と感じたか？
	RQ5	[根本原因]欄に記載された「病気」を患っていると認識することで、自ら直そうとする気持ちが働くと思うか？
監査品質の向上	RQ6	[根本原因]欄の分類名称(PL意識疾患、マネジメント疾患、etc)や、その内訳として付けた「病名」は、プロジェクト状態の認識合せをする上で使う「共通の言葉」にできると思うか？
	RQ7	「問診票」を使うことで属人的なSQA監査が減少し、SQA担当のトラブル兆候の検知が高まると思うか？

6.1 評価結果と考察

各RQ（Research Questions）の回答を集計した結果を図8に示す。尚、根本原因分析は自己改善の一つでもあるので「自己改善の推進」に含めて評価を行うこととした。また、仕組み構築については、認識を聞くまでもないためアンケート評価の対象外とした。

(1) 品質意識の底上げ（RQ1，2）

問診票に対する理解度を確認する質問を行い、RQ1，2ともに100%の「はい」との回答を得た。この結果は、トラブル事象に対する認識合せができたことを示しているため、品質意識の底上げが出来たと言って良いと考える。

(2) 自己改善の推進（RQ3～5）

RQ3と4では手法の納得感を確かめる質問を行い、RQ5では行動に結びつく動機を与えられるかを問う質問を行い、ほぼ全員から「はい」との回答を得た。この結果は、問診票の使用を組織プロセスに組み込みさえすれば、プロジェクトの自然治癒力が高まることの確認が取れたと考える。

(3) 監査品質の向上（RQ6，7）

プロジェクト監査を行う上での有効性を確認する質問を行った結果。RQ6では9割の回答が「病名を組織としての共通の言葉にできる」との回答で、RQ7では、7割が「トラブル兆候の検知が高まる」との回答であった。この結果は、問診票を利用した品質監査は、PLとQA間の認識齟齬を減らす土台となることを示しており、トラブルの兆候を早期発見に有効であると考えている。

上述したように、現時点では認識データしか取れない状況であるが、「独自で治療するためには、処方箋を提供して欲しい」旨のアンケートコメントも返ってきた。今までQAに対して現場の改善方法を聞いてきたことがないPLが治療方法の提示を要求してきたのである。このことは、問診票を展開したことで、品質向上に対する潜在要求を引き出したと考える。

以上の評価結果から、前章で述べた「五つの品質向上施策」は有効な取り組みになると結論付ける。

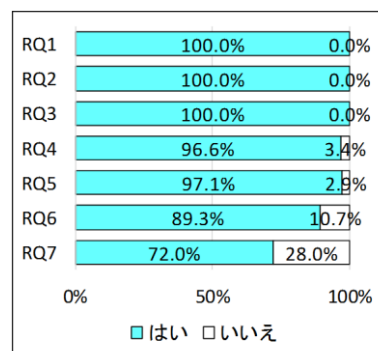


図8. アンケート集計結果

第7分科会（RCAグループ）

7. 今後のコト作り

今後の課題としては、ヒートマップ分析を行った上での新たな「コト作り」を想定している。具体的には、トラブル事象の発生頻度別に背景色の濃淡を付けることで、トラブル事象の分布を見える化する。この分析をトラブル別、根本原因別、プロジェクトタイプ別に行うことで、以下の様なコト作りが出来ると考えている。（図9参照）

(1) 失敗情報の活用

名医と呼ばれる執刀医は、手術の前に100件の手術記録を読破するという話しを聞いたことがある。100事例を頭に入れることで想定外の事象が無くなり、常に冷静に対処できるため、名医と呼ばれるのである。トラブル事象からトラブル事例（報告書等）を検索可能にすることで、使われずに眠っている失敗情報を活用し、更なる再発防止に繋げたい。

(2) 治療情報（処方箋）の提供

成果物問題は自然治癒しないが、それ以外のトラブル事象は自然治癒が可能

だと考える。特に根本治療は、本人の自助努力なくしては必ず再発する。よって、根本原因別に紐付くトラブル事象を明確にした上で、各種情報を辿る索引を整備したい。

(3) プロジェクトの見える化（テーラリング）

長期に継続する保守改善プロジェクトも珍しくない。問診票をテーラリングすれば、そのようなプロジェクトで起こった全てのトラブル事象をID化し、1枚の紙に表現することが可能となり、見える化が進む。このテーラリングは、プロジェクトタイプ別にトラブル事象を絞った問診票を提供することで、難易度が下がると考える。

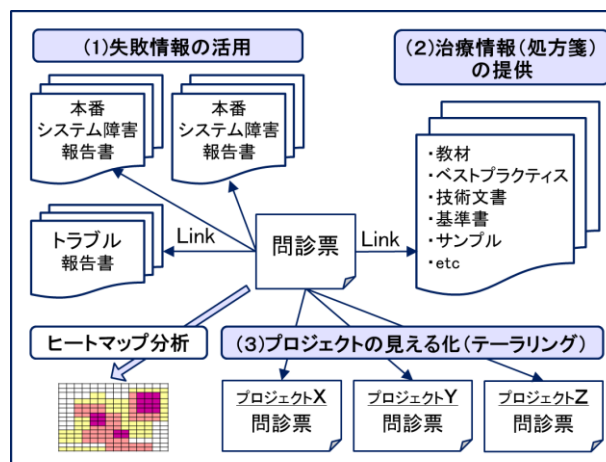


図9. 欠陥を中心に置いた更なるコト作り

8. まとめ

本研究を経て我々が出した答えは、トラブルがなくなる理由は「問題を問題と認識しないこと」と、組織が同じ失敗を繰り返すのは「トラブル情報を展開する仕組みを構築していないこと」の二つである。トラブル情報を組織展開する仕組みの構築をすることで、品質向上施策が可能となり、トラブルの早期発見と未然防止の基盤作りができるようになる。

PLは売上、利益、納期の狭間で短期的な問題解決に優先度を上げざるを得ないのだから、QAが長期的な視点で問題解決をする必要がある。よって、QAは「自分の組織で発生している欠陥を知らないことは問題である」と認識し、「欠陥を中心に置いた品質向上施策を定着化させる」ことをミッションとすべきであると考えている。尚、この定着化の取り組みは、CMMIレベル5・CAR（原因分析と解決）のプロセス作りにもなる。

また、本活用例ではソフトウェア開発や保守を請負うケースを紹介したが、このケースは多種多様な顧客とのプロジェクトを包含しなくてはならないため、事象表現が一般的な表現とならざるを得ない。しかし、製品開発や特定企業向けの開発や保守を行っているケースでは、事象をより具体的な表現で記述できる（例えば、業務名や機能名を使った表現が可能）ため、トラブル回避に対する有効度は更に高いものになると考える。つまり、今後のコト作り(3)で取り上げた見える化を、いきなり実施できると考える。

参考文献

- [1] 飯塚 悦功, 金子 龍三, 「原因分析」～構造モデルベース分析術～, P8, 2012 年
- [2] 稲盛 和夫, 「生き方」～人間として一番大切なこと～, P24～P27, 2004 年
- [3] 佐藤 允一, 「問題構造学入門」～知恵の方法を考える～, P107～P138, 2005 年

第7分科会（RCAグループ）

付録1. 旧「欠陥原因構造図」：「欠陥作り込み原因」に限定した欠陥原因の構造図

誘発要因				直接原因
要件定義欠陥	論理設計欠陥	物理設計欠陥	実装欠陥(製造)	
①ビジネスルールの把握不良	①要件定義書との不整合	①他の設計書との不整合	①代入/初期化不良	
②業務バリエーションの把握不良	②実装方式不良	②方式実装ロジックの不良	②チェック処理不良	
③利用条件の把握不良	③設計書標準不良	③環境バリエーションの考慮不良	③業務ロジック実装不良	
④システム構成の把握不良	④機能設計不良	④部品化/ID化の不良	④制御ロジック実装不良	
⑤外部I/Fの捕捉不良	⑤データ項目設計不良	⑤部品設計の不良	⑤セキュリティ実装コード不良	
⑥実装機能の洗い出し不良	⑥DB設計不良	⑥業務ロジック不良	⑥信頼性実装コード不良	
⑦使用性要件の設定不良	⑦インタフェース設計不良	⑦DB設計不良	⑦性能実装コード不良	
⑧セキュリティ要件の設定不良	⑧レイアウト設計不良	⑧領域間インタフェース不良	⑧データ/テーブル定義不良	
⑨信頼性要件の設定不良	⑨入出力設計不良	⑨セキュリティ実装ロジック不良	⑨データ/テーブル値設定不良	
⑩性能(効率性)要件の設定不良	⑩処理前提(制約)不良	⑩信頼性実装ロジック不良	⑩環境設定不良	
⑪データ量/利用者数の把握不良	⑪セキュリティ設計不良	⑪性能実装ロジック不良	⑪構成管理操作不良	
⑫ドキュメント記述不良	⑫信頼性設計不良	⑫標準化/規約不良	⑫パッケージ・パラメタ設定不良	
⑬パッケージ・バージョン選定不良	⑬性能設計不良	⑬標準化/規約違反		
⑭基本ソフト/ツールの選定誤り	⑭標準化/規約策定不良	⑭リラン/リスタート手順の不良		
	⑮標準化/規約違反	⑮ログ出力ロジックの不良		
	⑯設計変更不良	⑯リソース容量見積り不良		
	⑰ジョブネット定義の不良	⑰処理時間見積り不良		
	⑱起こりうる障害の洗い出し不良	⑱H/W選定スペック見積り不良		
	⑲バックアップ/リカバリ設計不良			
背景要因				
①要件/仕様の確認不足	④仕様の理解不足(認識不足)	⑦周知連絡の不徹底	⑩調査検討不足	
②対象業務の理解不足	⑤実装技術の習熟不足	⑧再利用の修正誤り	⑪指摘事項の反映漏れ/間違い	
③仕様の勘違い(思い込み)	⑥実装環境の理解不足	⑨仕様変更漏れ/間違い	⑫凡ミス、失念	
根本原因				
①制約条件	②手段	③活動	④外乱	
①無理な納期	①基本計画不良	①要件定義タスク不良	①仕様変更	
②予算不足(見積りミス)	②レビュー計画不良	②設計タスク不良	②顧客の仕様決定力不足	
③マネジメント体制不良	③構成管理計画不良	③製造タスク不良	③顧客の技術スキル不足	
④メンバー能力不足	④SQA計画不良	④テストタスク不良	④顧客のマネジメント不良	
⑤非機能要件の許容値不良	⑤外注化計画不良	⑤プロジェクト管理不良	⑤顧客の発注意識不良	
⑥既存設計書不良	⑥方法論不適合	⑥リーダーシップ不良	⑥開発場所分離	
⑦既存設計書未存在	⑦標準化策定不良	⑦人間関係不良	⑦メンバーの体調不良	
	⑧テスト計画不良	⑧コミュニケーション不良		
	⑨トレーニング計画不良	⑨モチベーション不良		

組織展開により、誘発要因と直接原因については、欠陥種類の組織内統一が取れた。

付録2. 「生き方」^[2]に記載の「人生と仕事の結果の方程式」より抜粋した文章

$$\text{人生・仕事の結果} = \text{考え方} \times \text{熱意} \times \text{能力}$$

人生や仕事の成果は、これら三つの要素の「掛け算」によって得られるものであり、決して「足し算」ではない。熱意と能力はどちらもプラス点しかなく、高ければいい結果が得られる。しかし、考え方にはマイナス点もあり、最も大事なものである。この考え方次第で人生や仕事の結果は決まってしまうと言っても過言ではない。考え方とは、心のあり方や生きる姿勢であり、哲学、理念や思想なども含む。

付録3. 「問題構造学入門」^[3]に記載の「根本原因の分類」

①制約条件	プロジェクト開始時点で存在する客観的事実で、手段と活動を制約する条件となる。取りうる手段や活動の範囲を限定するものであり、絶対制約と一時制約（時間と努力により変えられる制約）があり、後者は問題点となりうる。
②手段	方針に従い打つ手や計画のことで、プロジェクトの方針を具体化したもの。
③活動	関係者の全ての行動であり、現象的事実と推論的事実の両方を含んだもの。
④外乱	プロセスの外部に生じ、偶発的に発生し、活動をかく乱する不可抗力的要因となるもの。一過性のものと、後遺症として残るものがあり、後者は制約条件に転化する。

第7分科会（RCAグループ）

付録4. プロジェクト健康診断問診票（拡大図：左半分を編集して掲載）

応急処置／再発防止の対象（症状）	重症度 トラブル（危篤）	問題として捉える必要のある事象（367事象）				
	第三者介入 立て直し事象 （脱水状態）	A	①メンバー離脱	②取り引き停止	③プロジェクト中断	④品質起因での無償業務代行
	B	B. プロジェクト障害（28事象）				
	B1.モチベーション障害	B2. プロジェクト障害				
	①役割不履行 ②顧客からの叱咤によるPL萎縮 ③PL(PM)への不信感 ④PLとメンバー間に壁発生 ⑤人間関係不良 ⑥低いモチベーション ⑦低いセルフエスティーム ⑧鬱病状態	①顧客からのPL(PM)変更要請 ②顧客からの新たな報告資料の要請 ③場当たり的な事後対応 ④課題放置／未整理 ⑤作業漏れのリカバリー対応で忙殺 ⑥タスクの属人化 ⑦疲弊による単純ミスの増加 ⑧顧客からメンバーへの直接指示 ⑨必要な顧客対峙の未実施 ⑩（納品後）他Pjにアサイン不可な瑕疵対応	①品質に対する顧客クレーム ②本番リリース漏れ ③頻発する仕様変更対応 ④無秩序なBug対応 ⑤多発する前工程の障害 ⑥多発する類似障害 ⑦品質問題による検収NG ⑧リリース判定NG/不良 ⑨遅延に対する顧客クレーム ⑩マスタースケジュール変更			
	D	D. プロジェクト欠陥（74事象）				
	自力是正 報告/相談事象 （発熱）	D1. オーナーシップ欠陥(PM)	D2. オーナーシップ欠陥(PL)	D4. プロセス欠陥		
①プロジェクトの危険信号見逃し ②対会社関係の構築不足 ③プロジェクトを止める判断の欠落 ④ステアリングコミティ未開催 ⑤メンバー投入遅延 ⑥提案書レビューの未実施 ⑦プロジェクト計画レビューの未実施 ⑧キックオフMtgへの不参加 ⑨工程通過判定会議の未開催 ⑩プロジェクト振返りの未実施	①タスクの無着振り ②メンバースキルに合わないタスク割り当て ③WBSに無いタスクの指示 ④メンバーへの遅延対策のこり押し依頼 ⑤休出前提でのタスク依頼 ⑥メンバー任せの問題解決 ⑦メンバー放置 ⑧メンバーの高ストレス放置 ⑨BPMかせのタスク管理 ⑩営業との連携不足 ⑪エスカレーション不足 ⑫方針決定の遅延 ⑬課題／障害の長期滞留 ⑭不十分な体制強化 ⑮計画なしの過剰工数投入 ⑯スケジュール遅延の認識不足 ⑰遅延原因の聞き込み未実施 ⑱遅延原因の分析放棄 ⑲工程終了時の見積りトラッキング未実施 ⑳追加見積りの交渉遅延 ㉑スコープ外のタスク実施 ㉒契約前工程のタスクの実施 ㉓納品物の認識齟齬 ㉔仕様変更ルール の取決め不足 ㉕無償でやらざるを得ない仕様変更対応	①変更受入れ承認レビュー不良 ②納得感のない品質報告 ③レビューのスキル不足 ④未対応障害の積み残し増加 ⑤計画漏れタスクの発生 ⑥WBS不活用 ⑦期日が無理なWBS ⑧頻発するWBS組み直し ⑨タスクの進捗遅延 ⑩計画の倍の工数が掛かるタスク発生 ⑪進捗がないタスク発生 ⑫不透明な進捗報告 ⑬進捗の定量把握未実施 ⑭形骸化した進捗ミーティング ⑮ヘルプ要員での遅延対応 ⑯特機工数の発生 ⑰PLのパンク状態 ⑱顧客対応で忙殺 ⑲顧客指摘の横展開未実施 ⑳顧客への説明/報告不足 ㉑顧客との窓口設定不良 ㉒顧客担当の巻き込み不足 ㉓ユーザー部門の巻き込み不足 ㉔変更と障害の切り分け未実施 ㉕ベースライン未設定 ㉖進捗/課題の情報共有不足 ㉗内部リリース判定未実施 ㉘段取り不足のタスク実施 ㉙経験者のスキルトランスファー不足 ㉚移行データ調査不足 ㉛会議時間延長の常態化				
G	G. プロセスミス（80事象）					
早期発見 失敗確認事象 （咳・鼻水）	G1. 対外マネジメント・ミス	G2. 計画ミス	G3. プロジェクト運営ミス			
①顧客見積りミス ②顧客側工程定義の確認漏れ ③曖昧なスコープ設定 ④成果物スコープの見誤り ⑤役割リソースの合意漏れ ⑥重要ステークホルダー見落とし ⑦顧客との合意漏れ ⑧現行仕様と同じという合意 ⑨前提変化の見逃し ⑩顧客状況変化の見逃し ⑪オーバーコミットメント ⑫手順を踏まない仕様承認依頼 ⑬必要事項の交渉漏れ	①無謀なマスタースケジュール ②WBS作成遅延 ③必要なタスクの見落とし ④WBS網羅性不良 ⑤開発プラットフォーム習熟工数の見積りミス ⑥ソース解析工数の読み間違い ⑦移行元データ調査不足 ⑧経験者前提の見積り ⑨進捗管理指標の未設定 ⑩脆弱なマネジメント体制 ⑪脆弱なメンバー体制 ⑫不適格なメンバーアサイン ⑬形式的な役割定義 ⑭必要な会議体の設定漏れ ⑮報告ルート未定義 ⑯構成管理の計画未実施	①担当変更による知識欠落 ②計画文書の作成時間不足 ③計画書の未改訂/未使用 ④感覚レベルの進捗管理 ⑤課題/リスクの抽出漏れ ⑥要件変更未整理 ⑦文書フォルダ未整備 ⑧成果物へのバージョン未設定 ⑨採算のどんぶり勘定把握 ⑩必要書類の作成遅延 ⑪溢れタスクのPL引取り				
H	H. 根本原因（63事象）					
未然防止 根本対処事象 （病因）	H1. 制約条件	H2. PL意識疾患	H3. マネジメント疾患			
①既存設計書不在 ②既存設計書不良 ③要件/仕様の曖昧さ ④要件/仕様の変更 ⑤要件/仕様の取決め遅延 ⑥妥当性のない非機能要件の許容値 ⑦シビアな納期 ⑧予算不足（見積りミス） ⑨選定誤りのツール指定 ⑩体制作りが不十分な立上げ	①品質意識欠乏症 ②規模観欠乏症 ③重要度バランス失調症 ④顧客依存症 ⑤パートナー依存症 ⑥納期意識過剰症 ⑦成功体験過信症 ⑧頑固意識欠乏症 ⑨問題感受性欠乏症	①品質マネジメント欠乏症 ②定量管理意識欠乏症 ③レビュー意識欠乏症 ④テスト計画策定欠乏症 ⑤SQA監査ノウハウ欠乏症 ⑥プロジェクト計画策定欠乏症 ⑦進捗管理観点欠乏症 ⑧データ移行観点欠乏症				
	最初からある事象	プロジェクトを失敗に陥れる事象（4大疾患）				

付録5. プロジェクト健康診断問診票（拡大図：右半分を編集して掲載）

問題として捉える必要のある事象（367事象）				因果関係	
⑤コスト超過 ⑥納期遅延 ⑦本番誤操作 ⑧Goalが見えない障害対応 ⑨品質強化要請 ⑩本番システム障害					
C. 成果物障害（20事象）				現象	C
C1. 利用者障害		C2. 手戻り障害			
①機能漏れ ②業務効率低下 ③性能不良 ④キャパシティ不良 ⑤操作性不統一 ⑥無応答		⑦データ破壊 ⑧障害回復不能 ⑨運用要件見落とし ⑩保守要件見落とし ⑪災害／障害対策不備 ⑫セキュリティ対策不備			
E. 成果物欠陥（30事象）				直接原因	E
E1. 検証欠陥		E2. 保守性欠陥			
①テスト項目漏れ ②テスト項目解釈間違い ③テスト見逃し ④テストデータ不良 ⑤テスト環境不良 ⑥テスト結果のエビデンス不良 ⑦テスト結果エビデンスの見落とし ⑧コードレビュー見逃し		①標準化／規約違反 ②コードのスパゲティ化 ③デッドコードの発生			
		E3. 実装欠陥（製造）		E4. 実装欠陥（環境設定）	
		①代入／初期化不良 ②チェック処理不良 ③業務ロジック実装不良 ④制御ロジック実装不良 ⑤セキュリティ実装コード不良 ⑥信頼性実装コード不良 ⑦性能実装コード不良		①データ／テーブル定義不良 ②データ／テーブル値設定不良 ③環境設定不良 ④構成管理操作不良 ⑤パッケージ・パラメタ設定不良	
F. エンジニアリング欠陥（62事象）				発症要因	F
（欠陥見逃し）		（欠陥作り込み）			
F1. 品質保証欠陥		F2. 開発ルール欠陥		F4. 論理設計欠陥	
①レビュー未実施 ②レビュー観点漏れ ③レビュー見逃し ④レビュー時間不足 ⑤有識レビューアの欠席 ⑥テスト計画書の項目漏れ ⑦テスト観点漏れ ⑧テストシナリオ不良 ⑨欠陥分析未実施 ⑩品質の定量把握未実施 ⑪成果物の誤字脱字日本語間違い		①正しい仕様の未記載 ②設計書の未作成 ③設計標準の未整備 ④設計書記述レベルの不揃い ⑤仕様変更の設計書反映漏れ ⑥手順書の未作成 F3. 要件定義欠陥 ①ビジネスルールの把握不良 ②業務バリエーションの把握不良 ③利用条件の把握不良 ④システム構成の把握不良 ⑤外部 I/F の捕捉不良 ⑥実装機能の洗い出し不良 ⑦使用性要件の設定不良 ⑧セキュリティ要件の設定不良 ⑨信頼性要件の設定不良 ⑩性能（効率性）要件の設定不良 ⑪データ量／利用者数の把握不良 ⑫ドキュメント記述不良 ⑬パッケージ・バージョン選定不良 ⑭基本ソフト／ツールの選定誤り		①要件定義書との不整合 ②実装方式不良 ③設計書標準不良 ④機能設計不良 ⑤データ項目設計不良 ⑥DB設計不良 ⑦インタフェース設計不良 ⑧レイアウト設計不良 ⑨入出力設計不良 ⑩処理前提（制約）不良 ⑪セキュリティ設計不良 ⑫信頼性設計不良 ⑬性能設計不良 ⑭標準化／規約策定不良 ⑮標準化／規約違反 ⑯設計変更不良 ⑰ジョブネット定義の不良 ⑱起こりうる障害の洗い出し不良 ⑲バックアップ／リカバリ設計不良	
				F5. 物理設計欠陥	
				①他の設計書との不整合 ②方式実装ロジックの不良 ③環境バリエーションの考慮不良 ④部品化／ID化の不良 ⑤部品設計の不良 ⑥業務ロジック不良 ⑦DB設計不良 ⑧領域間インタフェース不良 ⑨セキュリティ実装ロジック不良 ⑩信頼性実装ロジック不良 ⑪性能実装ロジック不良 ⑫標準化／規約不良 ⑬標準化／規約違反 ⑭リラン／リスタート手順の不良 ⑮ログ出力ロジックの不良 ⑯リソース容量見積り不良 ⑰処理時間見積り不良 ⑱H/W選定スペック見積り不良	
G. プロセスミス（80事象）				背景要因	G
G4. 品質保証ミス		G5. コミュニケーションミス			
①レビュー計画不良 ②レビュー準備不足 ③チェックリストの形骸化 ④テスト計画不良 ⑤機能別難易度の設定不良 ⑥メトリクス収集不良 ⑦品質目標の未設定 ⑧SQA指摘不良		①解決した問題の共有漏れ ②仕様変更の伝達漏れ ③指摘事項の反映漏れ／間違い ④プロジェクト計画の共有漏れ ⑤ゴールの共有の未実施 ⑥プロジェクトルールの文書化漏れ ⑦キックオフMtgの議題不良 ⑧メンバーの士気を下げるPL言動 ⑨メンバーが理不尽と感じるPL指示 ⑩新規メンバーへのフォロー不良 ⑪高負荷メンバーへのフォロー不足 ⑫場を凍らすメンバー叱咤		G6. 開発プロセスミス ①要件／仕様のヒアリング不足 ②要件／仕様の理解不足 ③対象業務の理解不足 ④業務ロジックの文書化漏れ ⑤非機能要件の設計漏れ ⑥仕様の勘違い（思い込み） ⑦仕様の理解不足（認識不足） ⑧調査検討不足 ⑨利用ソフトの調査不足 ⑩影響範囲の調査漏れ ⑪実装方式設計の未実施 ⑫実装方式の設計漏れ ⑬本番データの理解不足	
				G7. トレーニングミス	
				①実装技術の習熟不足 ②標準化／規約の理解不足 ③実装環境の理解不足 ④メンバー習得スキルの把握不足 ⑤必要スキルの洗い出し不足 ⑥開発標準の不徹底 ⑦テスト方法の不徹底	
H. 根本原因（63事象）				再発防止・未然防止対象	H
H3. マネジメント疾患		H4. チーム作り疾患			
⑨リスク管理欠乏症 ⑩外注化計画策定欠乏症 ⑪技術習得欠乏症 ⑫変更管理失調症 ⑬要件管理失調症 ⑭構成管理失調症 ⑮スコープ管理観点欠乏症 ⑯顧客理解欠乏症		①ゴール共有欠乏症 ②ミーティング設計欠乏症 ③チーム統率欠乏症 ④チームビルド失調症 ⑤課題管理失調症 ⑥メンバー配属欠乏症 ⑦メンバー育成欠乏症		H5. 開発プロセス疾患 ①開発方法論欠乏症 ②標準化観点欠乏症 ③要求獲得欠乏症 ④業務ルール文書化欠乏症 ⑤業務ロジック文書化欠乏症 ⑥要件定義プロセス理解欠乏症 ⑦方式設計欠乏症 ⑧信頼性設計欠乏症 ⑨設計プロセス理解欠乏症 ⑩製造プロセス理解欠乏症 ⑪検証プロセス理解欠乏症 ⑫トレーニング視点欠乏症	
				H6. 不可抗力	
				①顧客のマネジメント不良 ②顧客の発注意識不良 ③顧客のスキル不足によるタスク発生 ④無理を通して顧客担当 ⑤関連するスコープ外タスクの遅延 ⑥顧客からのスコープ外タスク依頼 ⑦顧客都合のプロジェクト方針変更 ⑧開発場所分離 ⑨メンバーの体調不良	
プロジェクトを失敗に陥れる事象（4大疾患）				開始後に起こる事象	

付録6. プロジェクト健康診断問診票 (全体イメージ)

12