

なぜなぜ分析とシステム理論に基づく STAMP/CAST の事例による比較 ～ソフトウェアメンテナンスプロジェクトでの問題の分析事例に基づく比較～

日下部 茂
長崎県立大学
kusakabe@sun.ac.jp

三輪 東
SCSK 株式会社
azuma.miwa@scsk.jp

要旨

筆者らは、ソフトウェアメンテナンスプロジェクトで問題が発生した事例で、システム理論に基づく事故の説明モデル STAMP をベースとした分析手法 CAST を用いることで失いかけた心理的安全を回復できた。この組織では通常は問題分析においてなぜなぜ分析をベースとした分析を行っており、該当事例でも当初なぜなぜ分析をベースとした分析を行ったが、分析に限界があった。本稿では、システム理論に基づく STAMP/CAST となぜなぜ分析の比較について、分析結果に違いが出た上記事例をベースに比較と考察を行う。

1. はじめに

我々はソフトウェアメンテナンスプロジェクトで問題が発生した事例で、システム理論に基づく事故モデル STAMP (Systems-Theoretic Accident Model & Process) をベースとした分析手法 CAST(Causal Analysis using System Theory)[1]を用いることで、失いかけた心理的安全を回復できた例をいくつかの観点で紹介した[2][3][4]。その際、どのようにすれば同様の分析を皆が出来るようになるのかという質問や、必ずしも皆が使えるとは限らないことや適用可能性、使い分けに関する質問などを受けた。現時点では我々もそれらの質問に対する一般的な解は持っていないが、本稿では、部分的な解として、STAMP/CAST の分析と、比較的普及していると推測されるなぜなぜ分析との比較を事例ベースに行う。上記の事例を振り返りながら、分析の過程に焦点を当て比較する。

分析対象のプロジェクトはソフトウェアメンテナンスの元請けと委託先のメンバで実施されたものである。対象事例のドメインは金融関係で、多様で複雑な業務知識を要し、サブシステムごとにチームで担当している。それぞれのプロジェクトは、関係するサブシステムのチームによって実施する。プロジェクトはそれぞれ規模や期間が異なる上、複数が多重並行的に実行されることが多いため、

人材配置などに問題が生じやすい。そのため、組織的に人材の定着と多能工化を推進する育成活動にも力を入れている。そのような背景の下、対象事例では、図 1 の破線で囲んだ箇所を、委託先のメンバが未経験のサブシステムのサブチームリーダーを担当していた。

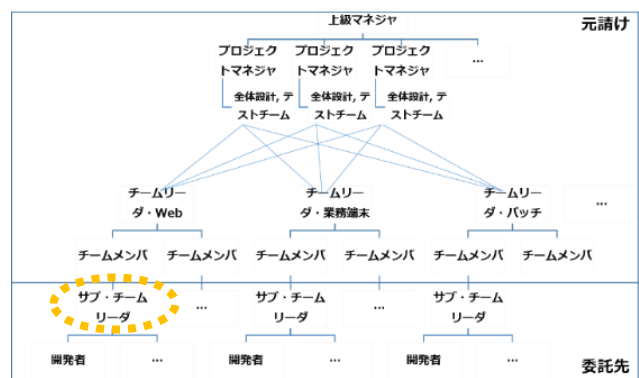


図 1 プロジェクト・チーム構成概要

当該プロジェクトでは、このような例外的な状況に元請け側が配慮し、入念に影響調査を行った上で追加・修正の対象を明確化し委託先に仕事を依頼した。しかしながら、指定範囲外での想定されていない修正が行われた。しかもその修正部分が検証されずに委託先から納品され、欠陥が運用直前の移行フェーズで見つかるインシデントが発生した。このような事態を受け、最初の調査はなぜなぜ分析ベースで行われ、委託先のルール違反、スキルや知識不足が根本原因という結論が報告された。個人非難につながるような結論により、元請けから委託先への信頼関係は崩れ、プロジェクト組織内での心理的安全は損なわれてしまった。しかしながら、システム理論に基づく STAMP/CAST でモデル化と分析を行った結果、元請け側も含めた関係者間での想定と実際のギャップに備えた望ましいチェックができてなかったという組織レベルの課題が明らかになった。これにより個人レベルの非難を回避でき、元請けと委託先との間の信頼関係を取り戻しチームの心理的安全を回復させることが出来た。

この事例において、上級マネージャであった著者の一人はなぜなぜ分析的なヒアリングや深堀から開始し、限界を感じたところで STAMP/CAST を導入するという、結果的にはハイブリッドなアプローチで、最終的な結果を導いている。本稿は、このような経緯も踏まえ、分析の結果よりも過程に焦点を当てて振り返りを行いながら、以下の構成で STAMP/CAST となぜなぜ分析の比較を論じる。2 章はなぜなぜ分析と STAMP/CAST を概説する。3 章は、事例でのポイントとなった心理的安全と暗黙知を説明する。4 章では、事例での分析過程を振り返りながら、なぜなぜ分析と CAST の比較を行う。5 章ではまとめを行う。

2. 分析法について

ここではなぜなぜ分析と STAMP/CAST の概説を行う。

2.1. なぜなぜ分析

なぜなぜ分析は、複数の要因がありえる現場で発生する問題の、真の原因を迅速に現場で発見して解決することを目的に、トヨタ生産技術の枠組みで提唱された[1]。Root Cause Analysis (RCA)手法のひとつとされ、問題の真の原因に到達するために、問題が発生した理由についての質問を 5 回以上続けて追究する。現場、現物、現実調査による三現主義のもと、推定される要因ではなく事実にもとづいて原因を見つけ、存在しない要因や不明の要因は検討の対象外とする。当初は人的・組織的レベルでの対処が必要な範囲でなく、現場レベルの問題解決手法として提唱されたが現在はより広く用いられている。

なぜなぜ分析は、現場での実際の問題解決が重視されており、著者の知る限り、おおよそ以下の手順を基本的にそれぞれにテーラリングされ適用されている。

- まず、分析の対象とする問題と分析の目的を明確化する。また、後述する深堀の過程で全体像を失わないよう初期の段階で全体像を把握しておく
- 時系列に経緯をさかのぼり、最初の「なぜ」として問題事象に至った経緯を分析し要因を列挙する。要因はひとつだけとは限らず、事象に対して論理的なつながりがなければならない。
- 次に、一回目の「なぜ」で得た要因ごとに、それが発生するに至った要因を列挙する。1 回目と同様、要因はひとつだけとは限らず、論理的なつながりがなければならない。
- 同様にして、3 回目の「なぜ」、4 回目の「なぜ」を繰り返していく。

このようななぜなぜ分析の実施には様々な注意事項が提唱され注意には原因を個人に帰さないことも含まれているが、対象事例でのなぜなぜ分析ベースの結論では個人非難のようなものになってしまった。

2.2. STAMP/CAST

STAMP は MIT の Nancy Leveson 教授によって提唱されたもので、従来の解析的還元論や信頼性理論ではなくシステム理論に基づき、システムを構成するコンポーネント間の相互作用に着目した事故モデルである。STAMP のモデルは、安全制約、階層的なコントロールストラクチャ、プロセスモデルという三つの基本要素で構成されている。

- コントロールストラクチャ: システムの構成要素間の構造と、相互作用を表したもの
- プロセスモデル: コントロールする側がその対象プロセスをコントロールするアルゴリズムと対象プロセスを(抽象化して)表現したもの
- 安全制約: 安全のために守るべき制約

STAMP では、コントロールストラクチャとプロセスモデルに対して、システムの安全制約が正しく適用されているかどうかに着目する。STAMP でのコントロールストラクチャとプロセスモデルの基本的な構造を図 2 に示す。

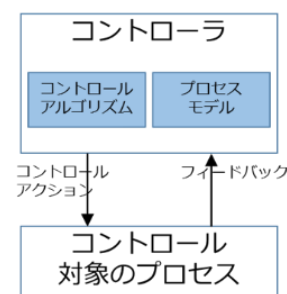


図 2 STAMP モデルの基本構成

STAMP 自身はアクシデントを説明するモデルであり分析技法ではないが、STAMP をベースとして、解析の道具立てやプロセスが複数提案されている。STAMP を用いた安全制約の分析は、作業成果物に対してだけでなく、そのような成果物の開発プロセスや運用プロセスに対しても行うことができ、今回の事例のようなソフトウェアのメンテナンスプロジェクトでも用いることができる。事故分析手法 CAST 以外にも代表的なものとしてハザード分析手法 STPA (System-Theoretic Process Analysis)、セキュリティ向け STPA-Sec (STPA for Security)といった分析手法が提案されている。STAMP は安全工学を対象として

提唱され、分析ではアクシデントやハザードを明確にする必要がある。しかしながら安全工学の技法を出来るだけ広く適用する意図で分析するアクシデントも慣習的な安全観点だけでなくミッションの未達なども含めた損失につながりうるイベントとなっている。関連して、ハザードは、環境のある最悪な条件の集合と重なるとアクシデントになり得るシステムの状態もしくは条件の集合、とされている。

今回なぜなぜ分析と比較する CAST 分析のステップは次のようなものである。

1. 損失に関与したシステムレベルのハザードを識別
2. そのハザードに関する安全制約やシステム要件を識別
3. そのハザードを制御するために存在した制御構造を識別
4. 損失につながった近接イベントを究明
5. 損失を物理システムレベルで分析(今回は組織の末端の個人行動等で適宜読み替えて分析)
 - (ア) 関与した物理的な制御や設備を識別
 - (イ) この事故の防止のための全ての物理的な安全要件と制約を識別
 - (ウ) 物理的な装置のあらゆる故障もしくは不適切な制御を識別
 - (エ) 物理的な故障もしくは不適切な制御を説明するコンテキスト要因を識別
6. 引き続き各高位のレベルが、どう、なぜ、現レベルの不適切な制御に寄与したか説明するため、高位のレベルを分析
 - (ア) 次の高位レベルの制御に対する、安全関連の責務を識別
 - (イ) 非安全な決定と制御アクションを識別
 - (ウ) 非安全な決定と制御アクションを説明するプロセスモデルの欠陥(誤信)を識別
 - (エ) その時点でなぜその振舞いが適切に思えたか説明するコンテキスト要因を識別
7. 損失に寄与した全体的な調整ややりとりを確認
8. 損失に関連したシステムと安全制御構造のダイナミクスや変化、継時的な弱さを割り出す
9. 改善勧告を出す

3. 心理的安全性と暗黙知

3.1. 心理的安全性

心理的安全性は、チームの各メンバーがそのチームに対し、気兼ねなく発言でき、本来の自分を安心してさらけ出せると感じられるような環境や場の状態をいう[6]。例え

ば Google の Project Aristotle の結果が示すように、心理的安全性はソフトウェアプロジェクトの成功にとって重要とされている[7]。チームに明確な目標を設定するなど、他にも重要な基準があるものの、Google の Project Aristotle の結果では、生産性の高いチームは心理的安全性が最も重要であるとされている。また、比較的先進的なソフトウェアプロセスである、モダンアジャイルプロセスでは、心理的安全性もアジャイルプロセスの最も重要な側面の 1 つと見なされており、推進者の中には、心理的安全性に関連する活動を特に「Anzeneering」と呼ぶものもいる[8]。

このようにソフトウェア開発のプロジェクトやプロセスでは心理的安全性が重要とされており、今回のプロジェクトの上級マネージャにとっても関心事の一つであった。Google のガイドでは以下の 7 つの質問の中で、ポジティブな回答が多ければ、メンバがチームの中で心理的安全性を高く感じているとされている[9]。

1. チームの中でミスをする、たいがい非難される。
2. チームのメンバは課題や難しい問題を指摘し合える。
3. チームのメンバは、自分と異なるということを理由に他者を拒絶することがある。
4. チームに対してリスクのある行動をしても安全である。
5. チームの他のメンバに助けを求めることは難しい。
6. チームメンバは誰も、自分の仕事を意図的におとしめるような行動をしない。
7. チームメンバと仕事をするとき、自分のスキルと才能が尊重され、活かされていると感じる

例えば、インシデントの調査で個人非難のような結論が出てしまうとネガティブな意味で上記第 1 項目から当てはまり心理的安全性が損なわれる。

3.2. 暗黙知

上級マネージャは経験から暗黙知に関してトラブルの要因となりえる問題点が二つあると考えていた。一つは暗黙知のそのものの問題、もう一つが暗黙知とアジリティ的な価値観が合わさった複雑なスキル不足の問題である。ここでの暗黙知は「ももとのチームではあたりまえであるが、新規参入者には理解しにくい知の集合体と現状。ルール・規律、言語パッケージやフレームワーク構造、作法・判断基準など、普段は言及されないけれど存在するもの全般を含む。」とする。例えば、仕様書に明示的に記載されている情報は暗黙知には含まないが、その設計思想や、様々な実装者が修正してきたソースコードの経緯など、知の集合体を背景に持つファイル自身と、そのファイルを取り扱う前提となる様々な作法や期待される行動などの全般を指す。

こういった暗黙知は長年チームで蓄積されたノウハウや認知的側面[10]が多いため問題は単純ではない。SECIモデル[11]のように歴史と共に変化・進化し、プロセスや成果物フォーマットなどにちりばめられつつ、その現場の慣習や文化になったものは、可視化が難しい。7Sの「ソフトの4S(Shared value, Style, Staff, Skill)」[12]に染み渡った状況であり、プロセス資産、個人のスキルや意志・行動、文化といった組織の環境要因に相互依存しながら混じり合い、境界線を引くことも難しい。当事者でも、その構造を端的に説明することは難しい場合も多い。新規参入者の場合、既存メンバすら言語化出来ないものを簡単に理解できるはずもなく、その対処は容易でない。

4. モデリングと分析の過程

当時の分析経緯を可能な限りふりかえりながら、事例ベースでSTAMP/CAST分析となぜなぜ分析の比較を行う。心理的安全が脅かされる結論が出た後の上級マネージャの再調査でも、当初はなぜなぜ分析ベースのアプローチから着手した。しかしながら、問題のきっかけとなった逸脱的振る舞いは単純ミスではなく、未経験メンバと暗黙知に起因する部分がありそうだと分かった段階でSTAMP/CASTの採用を検討した。

4.1. なぜなぜ分析の深堀とヒアリング

上級マネージャは、まず、次のような視点(以降、視点一と呼ぶ)で元請け側のチームリーダーにヒアリングを行った。(WHY/ANSの下線付き強調表示部は詳細を後述)

表 1 視点一によるなぜなぜ分析

WHY: なぜ、想定外のソースコードに不具合が混入したのか?
ANS: 委託先のミスによる
WHY: なぜ委託先はミスしたのか?
ANS: ルール通り行ってくれなかった
WHY: モラルハザードでも起きているのか?
ANS: そんなことは無く、今回の件は全員が残念に感じている。

この視点での掘り下げは難しかったため、次のように視点を変えて(以降、視点二と呼ぶ)再度元請け側のチームリーダーにヒアリングを行った。

表 2 視点二によるなぜなぜ分析

WHY: 何か他の事情はないのか?
ANS: 未経験のメンバがいて不慣れだったのは事実
WHY: スキルが大幅に足りなかったということか?
ANS: 他のチームではリーダー経験があり、スキル不足とは言い難い
WHY: このチームでは初仕事ということで、仕事の仕方を理解していないということか
ANS: その通り。
WHY: すると、それをフォローするのは元請け側の責任ではないか?
ANS: こちらも対象範囲を限定して依頼することで、最大限考慮した。
WHY: 対象範囲を間違ったか否かをチェックするのも元請け側の役目ではないのか?
ANS: 初作業だったので、普段は委託側にお願いしているソースへの影響調査からテスト範囲まですべてこちらで確認したうえで、事故が起きないように「これだけやってください」とお願いした。それを逸脱されたら、こちらもフォローのしようがない。
WHY: 事前に相談や確認は無かったのか?
ANS: なかった
WHY: いつも勝手にやるのか?
ANS ₁ : まさか、そんなはずはない。こちらが気づかないケースで指示以外のソースを修正すべきケースも勿論ある。その場合は当然、報告・確認があるし、その為に会議をやっている。その会議で相談・確認もなく、「これだけやって」と指示した範囲外を修正されても、こちらは対処のしようがない。

ここで、対象を委託側のリーダー(当事者の上司)に変え、次のような視点(以降、視点三)でヒアリングを行った。

表 3 視点三によるなぜなぜ分析

WHY: 限定した範囲を逸脱したのはなぜ?
ANS: バックログで該当箇所があり、いつも通り修正をかけた
WHY: なぜ、事前に相談・確認が無かったのか?
ANS: いつもなら確認することが漏れてしまった、

プロセスの問題かもしれない
WHY: 状況を確認する会議体は定期的に行われており、プロセスとして存在していると思われるが、違うか？
ANS: 確かにプロセスとして存在している。やはり、不慣れだったと思われる。
WHY₂: スキル不足ということか？
ANS: それは否定できない。

上記のようなやり取りが行われ、対話上では深堀ができていくようにも見える。実際の現場で通常行われる深堀はこういった流れであり、表 2 や表 3 のように深堀が STOP してしまうと表 2 の ANS₁ や表 3 の WHY₂ で対話を終えることになり、委託先の指示逸脱行為、スキルや知識不足が根本原因と片づけてしまいたくなる。

4.2. STAMP/CAST での再分析

前述の視点二と視点三について図 2 のような STAMP の基本構造を念頭にモデル化と CAST での再分析を試みた。CAST では節 2.2 で紹介した手順 3 にあるように、安全制約を満たす全体的な制御構造を考える必要がある。今回の場合、必ずしも手順通りでない試行錯誤の分析での構築であるが、制御構造は図 3 のようになる(ただし、プロセスモデル等の詳細は略してある)。

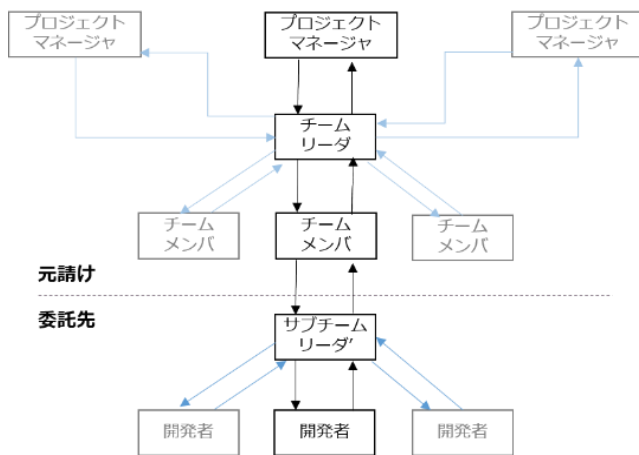


図 3 該当事例でのプロセスの制御構造の概要

STAMP でのモデル表現のために、視点二は元請けから委託先への制御、視点三は委託先内での制御に対応付けて考えると次のようになる。

表 4 視点二での STAMP モデリングの試行

コントローラ	仕事の進め方の理解が不十分と思われた → 範囲を限定して指示
被コントローラ	良くするつもりで指示範囲を逸脱
コントローラ	実際に範囲が限定されているかの確認は、会議体で追加作業の報告が無かったことで問題なしとしてしまった

表 5 視点三での STAMP モデリングの試行

コントローラ	限定された範囲で作業指示をした
被コントローラ	指示された範囲で修正を試みたが、バックログがあったため、指示範囲以外の作業も必要となった
コントローラ	限定された範囲外での作業に気付いたが、被コントローラからの説明でそれが適切だと、納得してしまった

この段階で、視点二と視点三のモデル化の結果をつなげることを試みると、視点三では範囲外の修正について適切と納得してしまっており、視点二のコントローラが想定する「会議体で追加作業の報告」につながらないことに気づく。より上位である視点二では指示した限定範囲が正しいという前提に基づいているが、下位側の視点三では、そもそも範囲の限定自体が不適切と判断したことを表しており、このつながりは構造上で不整合が発生している。何か、大事な要素が抜け落ちていることが分かる。

5. 考察

5.1. 比較概要

なぜなぜ分析は直線的なイベントチェーンやイベントツリー的な分析が当てはまる場合には有効だが、今回のような単純ではない暗黙知が関係する文脈依存の問題では必ずしも有効ではないと思われる。STAMP/STPA のようにモデル要素として制御構造やプロセスモデルが含まれている方がそのような問題でも分析しやすいと思われる。しかしながら、使い慣れていない場合、STAMP/STPA での分析は、着手時点から困難を感じる可能性がある。今回は、なぜなぜ分析をベースにインタビューと深堀を行いながら、STAMP でのモデリングを試行した。制御構造の階層をふまえてハザードを分析するとともに、問題のある構造を探し出すという段階を踏んでいった。その

経験をもとに比較の概要をまとめると以下となる。

表 6 なぜなぜ分析と STAMP/CAST の比較概要

	なぜなぜ	STAMP/CAST
着手しやすさ	易しい	難しい
分析スピード	早い	時間がかかる
制御構造	実施者が意識してつなげる	モデル要素に含まれる
制御決定の背景(プロセスモデル)	実施者が意識して抽出	モデル要素に含まれる

なぜなぜ分析は特にモデルなどを前提とせずに実施できるため着手しやすしい。しかしながら実際の分析は、実施者の知識やスキルに依存した属人性が高くなると考える。STAMP はコントロールの問題に焦点を当てた事故モデルであり、イベントチェーン的なものでない複雑な問題でも広義のコントロールの問題に帰着できる場合は有効なモデリングと感じた。分析に効果的なモデル要素が含まれているため、STAMP/CAST 分析は強力であり、いったんマスターできればある程度低い属人性で実施できると感じた。しかしながら、STAMP 固有のモデリングに加え、適切な抽象化と具象化といった一般的な意味でのモデリングに関する能力やドメイン知識が必要であると感じた。

5.2. 事例のなぜなぜ分析での欠落に関する考察

なぜなぜ分析の深堀は、掘り下げていく性質上、最後の対話に着目されやすい。今回の件で言えば、表 2 の ANS₁ や表 3 の WHY₂ のようなものが原因として取り上げられやすい。しかしながら、表 2 や表 3 中に下線付き太字で強調したような、やり取りの途中で出て来たハザード分析候補(表 7 参照)への考慮が欠落しがちである。

表 7 ハザード分析の候補

WHY (表 2)	このチームでは初仕事ということで、仕事の仕方を理解していないということか
ANS (表 3)	いつもなら確認することが漏れてしまった、プロセスの問題かもしれない
WHY (表 3)	状況を確認する会議体は定期的に行われておりプロセスに存在していると思われるが、違うか?
ANS (表 3)	確かにプロセスとして存在している。やはり、不慣れだったと思われる。

上記のハザード分析候補を STAMP で書き表すと以下のようになり、コントローラと非コントローラの違いがあり、プロセスの実践上問題があることが分かる。

表 8 コントローラと非コントローラの不一致

コントローラ	確認するプロセスが存在し、その遵守は当然。不明点は質問するはず。
被コントローラ	確認するプロセスには不慣れで理解は不十分。不明点を適切に言語化して適切な人に聞けるとも限らない

残念ながら、なぜなぜ分析の対話の中では、プロセスのルールがあれば当然守るべきという前提で対話が進んでしまっている。今回は発注側と受注側の力関係や権威勾配なども背景にあるかもしれない。そもそもプロセスが存在してもその実践の仕方について理解していない・分かっていないという可能性についての考慮が消えてしまっており、より良い深堀になっているとは言い難い。

5.3. 実際の現場での活用について

現場ではコストや時間に制限があるので、着手の障壁や実施コストが相対的に低いなぜなぜ分析から始めるのが現実的と思われる。しかし、なぜなぜ分析は最後の対話に着目されやすく、それ以前の(5W まで深堀なら)1W ~4W の内容を適切に反映しない結論にする可能性がある。実施者が最後の対話以外の関連性を適切に取り扱うスキルが必要と思われる。このようなある種属人的なスキルを補う意味で、分析に必要な要素をもとからモデルに含む STAMP/CAST と組み合わせることは現実的な解と思われる。STAMP/CAST のように制約を保つ制御構造とプロセスモデルを対象に分析を行うことで、なぜなぜ分析で欠けている点を補完できると考える。

単純な線形思考の場合、先に答えを仮定し、その答えに合う情報で肉付けして終わらせてしまうといった、単純化や確証バイアスが生じやすい可能性もある。STAMP では、制約とそれを保つ制御ループ構造があるので、ある種のごまかしが難しいと感じた。仮定がある側面だけから導き出されたものだとすると、フィードバックサイクルの中で矛盾を分析しやすく、それがハザードと関連づくケースが多かった。それらを分析すると新たな矛盾が出るといった繰り返しで全体がつながるようになった。

STAMP/CAST のような分析手法は非常に強力と感じる一方、誰もが簡単に使えるものではないと感じた。分析

手法が明快に定義され、他で成功事例があったとしても、実際の対象事案に関するさまざまな構成要素や文脈を適切に取り扱える論理的な思考や抽象化能力などが無いと、簡単には活用できないと思われる。数名にワークショップに参加してもらったが、結局「よく分からなかった」で、実際にモデリングや分析ができる人材は増えなかった。

今回の事例でも最初のモデリングは、表記だけみればコントローラ、被コントロールプロセス、コントロールアクション、フィードバックデータといった STAMP でのモデル表記ができたものの、制約条件やハザードと種々のモデル要素の関係は当初明確ではなかった。STAMP の良さは、人が関係するプロセス上の事故であっても、誰が・何が悪いのかではなく、なぜ・どのようにコントロール出来なかったのかを明らかに出来ることと感じた。それを明らかに出来るまで試行錯誤を繰り返せたのが STAMP の良いところだと感じている。構造的には書いているように見えても、そこで終わらせられず、なぜ・どのようにコントロール出来ていないのかまで明らかにしようとすると、必然的に様々な依存関係まで明らかにする必要があり、そこまで書かないと納得できるものが作成できないのが利点だと感じた。今回のようなソフトウェア開発だけでなく、医療も含め人の振る舞いが関係する事故に対して、対策志向型で取り組み、エラーの背後要因[13]を明らかにするには高いポテンシャルを持つ良い手法と考える。

5.4. 発展的活用に向けて

STAMP は、様々な安全工学の関連領域について議論され提唱された、セーフウェア[14]のコンセプトを具現化するための手法とされている。そのため STAMP の活用においてもそのような関連領域の知識があるとさらに活用の可能性が広がると考える。

例えば、認知心理学の認知情報処理特性の観点から、人の行動を三つのレイヤに分けた Rasmussen の SRK モデルを STAMP/STPA に取り入れた事例もある[15]。今回の事例で重要な切り口であった暗黙知の問題も、STAMP のプロセスモデルの、特にメンタルモデルを認知心理学や社会心理学のモデルを関連付けることができれば、心理学の知見を活用して分析できると考える。

このような関連領域の知見は、なぜなぜ分析などの他の分析法でも活用は不可能ではないと考えるが、分析に効果的なモデル要素を持つ STAMP でのモデリングと分析では、関連領域の系統的な対応付けと活用がより容易になると考える。

6. おわりに

ソフトウェアメンテナンスプロジェクトで問題が発生した事例で、システム理論的な事故説明のモデル STAMP をベースとした分析手法 CAST を用いることで失いかけた心理的安全を回復できた報告に対し、手法の一般性、適用可能性、適用の再現性、使い分けなどの質問を受けていた。本稿では、それらの問いに対する部分的な解として、既報告の事例を分析結果よりも分析過程に焦点を当て振り返りながら、事例ベースで STAMP/CAST と比較的普及していると推測されるなぜなぜ分析を比較した。

主観が入っている可能性があるが、因果関係が直線的もしくはそれに近い場合は、なぜなぜ分析の方が低コストで実施可能であり、暗黙知を紐解くような複雑な要因間の関係がある場合は、相対的に着手の障壁やコストが高くなるが、システム理論に基づく STAMP/CAST を用いると効果的な分析ができるとの感触を持った。教条主義的な使い分けでなく、なぜなぜ分析をベースに分析を開始し限界を感じた場合に STAMP のモデリングを行うといった使い方も有効と考えられる。

STAMP のような特定のモデリング方式に基づくモデルの構築や分析は、分析に効果的なモデル要素を系統的に含むことができる一方で、学習のコストやバリアが存在する。しかしながら、システム的な観点で多層的に関連領域の知見を系統的に活用し強力な分析ができる可能性を感じており、引き続き関連する取り組みを継続する予定である。

参考文献

- [1] Nancy Leveson, Engineering a Safer World, MIT press, 2012.
- [2] 日下部 茂, 三輪 東, ソフトウェアメンテナンスプロジェクトチームの心理的安全確保に向けた CAST 適用の事例, 第 3 回 STAMP ワークショップ, 2018
- [3] 三輪 東, 日下部 茂, ソフトウェアメンテナンスプロジェクトチームの心理的安全確保に向けた CAST 適用の事例, ソフトウェア信頼性研究会 第 14 回ワークショップ (FORCE2018), 2018
- [4] 日下部 茂, 三輪 東, プロジェクトチームの心理的安全のための STAMP/CAST による事後分析, プロジェクトマネジメント学会 2019 年度春季研究発表大会予稿集, pp.335-340, 2019
- [5] 大野耐一, トヨタ生産方式:脱規模の経営をめざして, ダイアモンド社, 1978

- [6] Amy Edmondson. Psychological safety and learning behavior in work teams, *Administrative Science Quarterly*, 44:2, 350-383, 1999
- [7] Charles Duhigg, What Google Learned from Its Quest to Build the Perfect Team. *The New York Times Magazine*. <https://www.nytimes.com/2016/02/28/magazine/what-google-learned-from-its-quest-to-build-the-perfect-team.html> (アクセス 2019-02-25)
- [8] Modern Agile. <http://modernagile.org/> (アクセス 2019-02-25)
- [9] Google re:Work.「効果的なチームとは何か」を知る, <https://rework.withgoogle.com/guides/understanding-team-effectiveness/steps/introduction/> (アクセス 2019-02-25)
- [10] 知識創造企業, 野中郁次郎・竹内弘高, 梅本勝博訳, 東洋経済新報社, ISBN978-4-492-52081-9
- [11] SECI model of knowledge dimensions, https://en.wikipedia.org/wiki/SECI_model_of_knowledge_dimensions, (アクセス 2018-03-12)
- [12] 7S, https://mba.globis.ac.jp/about_mba/glossary/detail-12513.html, (アクセス 2018-03-12)
- [13] 石橋明, ヒューマンファクターとエラー対策, *J. Natl. Inst. Public Health*. 51(4):2002
- [14] ナンシー・レブソン著, 松原友夫監訳, セーフウェア, 翔泳社, 2009
- [15] Nobuyuki Hoshino, Applying Human Mental Model to STAMP/STPA, 3rd MIT STAMP Workshop, 2014