IPA Better Life with IT

情報処理システム 高信頼化教訓集

ITサービス編

独立行政法人 情報処理推進機構
社会基盤センター

別冊Ⅱ：障害分析手法

2020年3月16日 改訂

情報処理システム高信頼化教訓集（IT サービス編）
別冊Ⅱ：障害分析手法

独立行政法人情報処理推進機構

別冊Ⅱ

障害分析手法

別冊Ⅱ：障害分析手法　目次

１．はじめに

一般的な原因分析手法を整理し、理解することを目的として、各種手法の概要をまとめた。

原因分析手法の種類（主要原因分析手法一覧を参照）

・過程関連型

要因を抽出後、要因間の関連に着目して整理する。

・組織関連型

組織要因、機能要因に着目して整理する。

・リスク評価型

リスク分析の方法を原因分析に応用する。

・基本型

過程関連型の一種。各種手法のテクニックとしても用いられる。

・IT 特化型

ベンダ F 社の分析サービス。従来手法と比べて、リーダに負担の少ない傾向分析と根本原因分析が可能。

・発展型

コンポーネント（製品）主体である「HAZOP」と「FTA」と「FMEA」に対してソフトウェアを、インタフェースを含めたシステムとしてとらえ、製品主体だった手法を拡張した。

今回取り上げる原因分析手法は以下とする。

・広く分野を越えて利用されている基本型の「なぜなぜ分析」

・過程関連型の「ImSAFER」と「RCA」

・IT 特化型の「総合的インシデント分析」

・組み込み制御で使われる未然防止をそなえた「HAZOP」と「FTA」と「FMEA」

・従来の製品／ハード中心技術から発展し、システム安全を上流で設計する手法である「STAMP」と分析ツールの「STPA」、「CAST」

２．主要原因分析手法

文献調査に基づく主要な原因分析手法の一覧を表２－１に示す。

表２－１　主要原因分析手法一覧

番号	分類	名称	開発者開発機関	概要の説明
1	基本型	なぜなぜ分析	（品質管理手法）	発生した問題の根本原因を事後に分析する手法として広く使われている
2	過程関連型	ImSAFER（Improvement for medical System by Analyzing Fault root in human Error incident）	自治医科大学（河野龍太郎）	ヒューマンエラー関連の事後分析において原因追求と対策立案を支援
3	過程関連型	RCA（Root Cause Analysis）	米国退役軍人省　患者安全センタ	医療分野における問題の事後原因分析手法で、なぜなぜ分析を包含する
4	IT 特化型	総合的インシデント分析	富士通	IT 分野のインシデントに特化した事後分析手法でサービスとして提供
5	リスク評価型	HAZOP（Hazard and Operability Studies）	イギリスの ICI 社（Imperial Chemical Industries）	事前のリスク分析手法（ボトムアップ型、FMEA と類似）
6	過程関連型	FTA （Fault Tree Analysis）	Bell Labs. 他	事前の故障の木解析手法でトップダウン型
7	リスク評価型	FMEA（Failure Mode and Effects Analysis）	US.Army 他	事前のリスク分析手法（ボトムアップ型、HAZOP と類似）
8	発展型	STAMP（Systems-Theoretic Accident Model and Processes）	MIT	事故モデル（複雑なシステムの安全解析）
9	発展型	STPA（System-Theoretic Process Analysis）	MIT	STAMP に基づく事前の安全解析（トップダウン型）
10	発展型	CAST（Causal Analysis using STAMP）	MIT	STAMP に基づく事後の事故分析（ボトムアップ型）

以下では、各手法について詳しく説明する。

２．１　なぜなぜ分析

（a）手法の利用シーン

過程関連型の原因分析手法の一種であり、発生した問題の根本原因を分析する際に広く使用される。

実施の際に特段のツールの導入を必要としないこと、また、製造業で成功した方法であることから、エンタープライズ系システムの障害原因の分析においても、広く普及している。

（b）手法の概要

なぜなぜ分析には一般に、以下 4 点の目的がある。（文献 1）

・再発防止のため

同一障害の再発や類似障害の発生を防止するために発生の根本原因を解明してそれを解消する。

・開発力改善のため

問題の分析能力を向上させ技術力を高める。

（技法、ツール、標準等の改善を含む）

・QCD 向上のため

品質を向上させ、その結果としてコストと納期を改善させる。

・組織活動全体の改善のため

同一障害、類似障害の再発防止だけでなく、業務の仕組みに根ざす原因の洗い出しおよび解消により、組織活動を根本的に改善する。

（c）記法 ならびに分析記法

問題事象を起点として、その発生原因を「なぜ」と問いながら、根本原因まで遡っていく分析手法。

「なぜなぜ分析」の実施方法の詳細については解説書も多数出版されていることから、初めて実施する際には、それらを一読して内容の理解を深めた上で実施すると、分析の精度をより深めることが期待できる。

（d）分析の例

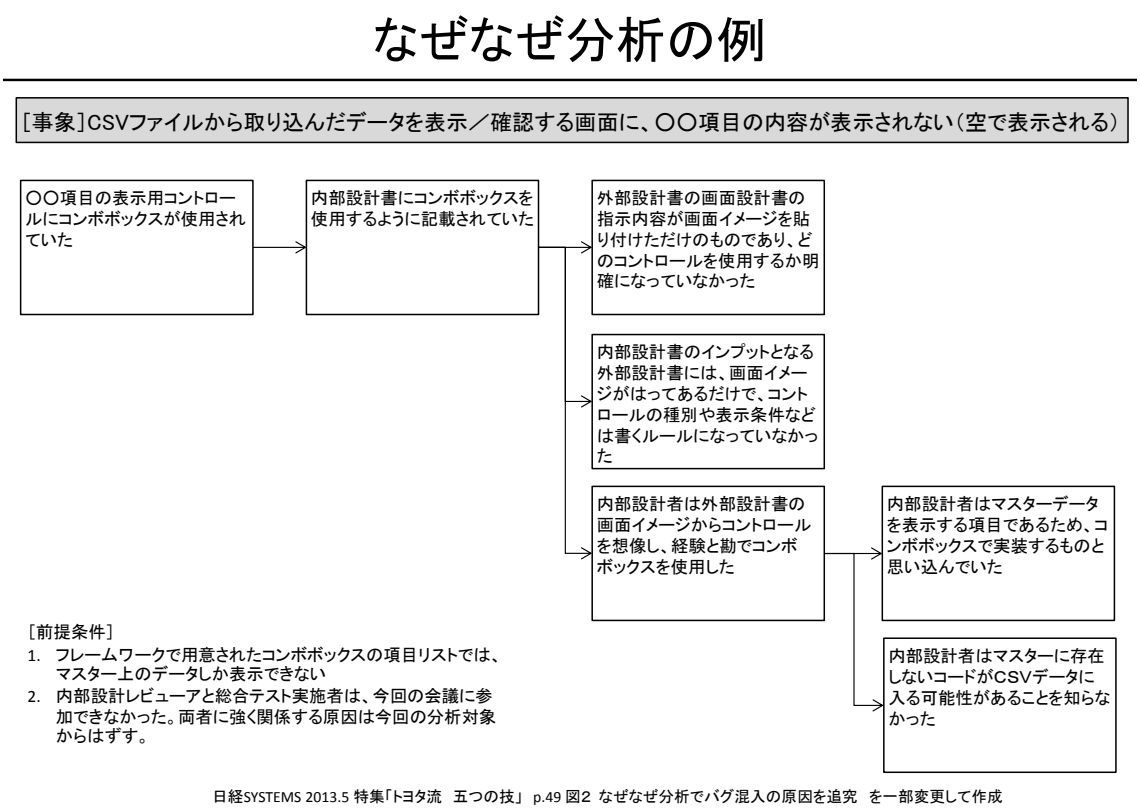


図２－１　なぜなぜ分析の例（文献 3）

２．１．１　なぜなぜ分析による原因分析

【インプット：分析対象の問題事象、アウトプット：根本原因】

なぜなぜ分析は、発生した事象（問題：対策を必要とする事実）を起点として、その事象が発生した根本原因（それに対策すれば問題が再発しなくなる原因）に至るまで、「なぜ（それが発生したのか）」と問いながら原因を遡っていく分析手法である。

分析を実施するにあたってのポイントは以下の 2 点である。

① 問題事象を発生させた直接の原因（直接原因）を見つけ、さらに直接原因を引き起こした原因を見つけるという手順をくり返し、本質的な原因（根本原因）を見つけ出す。

② 分析は事実をもとに実施する。そのため、実施の際には、可能な限り障害事象の当事者（障害を引き起こした人等）の参加を促すことが望ましい。その際に注意すべきことは、あくまで、実施の目的は障害を発生させた「個人」を責めることではなく、原因を究明して「組織」としての再発防止策を検討することである、ということを参加者に徹底することである。

以下の図２．１．１－１になぜなぜ分析の例を示す。

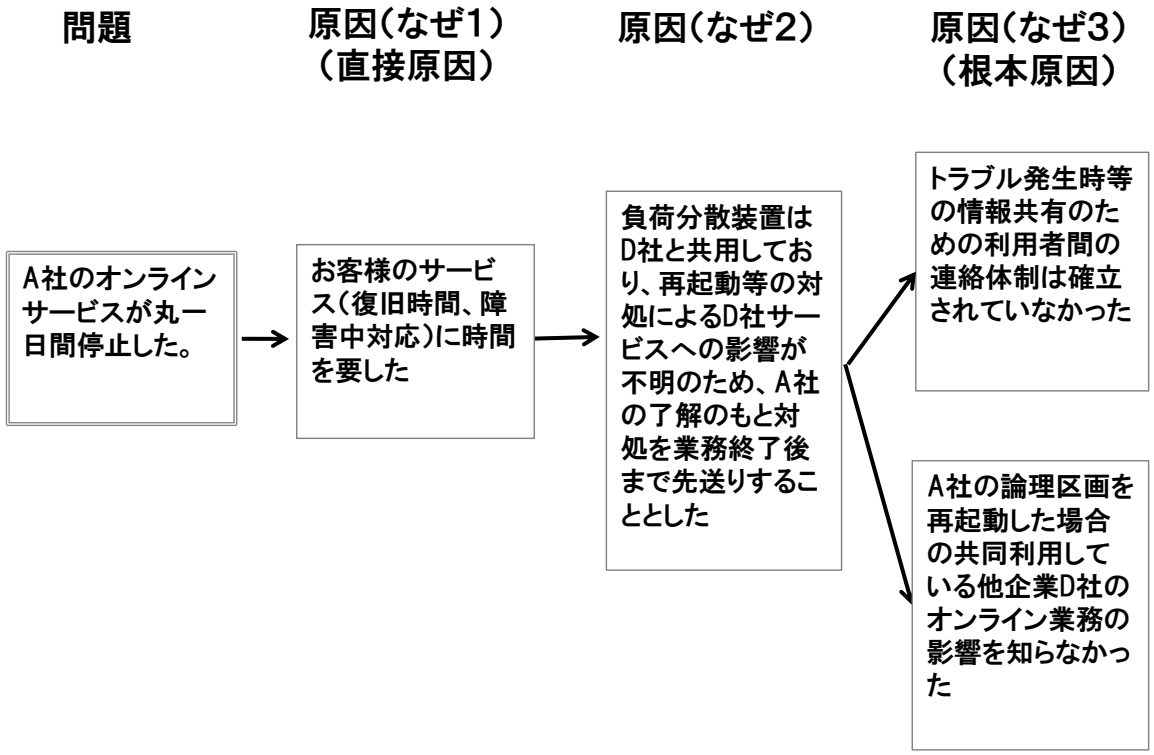


図２．１．１－１　障害事例（なぜなぜ分析の例（原因分析））

【参考１．３．障害事例をなぜなぜ分析で検討した例　の一部】

２．１．２　なぜなぜ分析による再発防止策／未然防止策の検討

【インプット：直接原因／根本原因、アウトプット：再発防止策／未然防止策】

なぜなぜ分析から再発防止策と未然防止策を導く手順を以下の図２．２．２－１に示す。対策を検討するにあたってのポイントは以下の２点である。

① 直接原因→反転→直接原因への対応策　を検討する。
発生した障害への直接的な対応としてすでに実施されている場合が多いが、それらの対応が同一原因による障害の再発防止策として妥当であったかも検証する

② 根本原因→反転→再発防止策、未然防止策　を洗い出す。
分析された根本原因が妥当であったかどうかも検証する。根本原因を解消できる妥当な（対策に要する費用や人的・物的リソースが投入でき、それに見合う効果が期待できる）再発防止策が見出せない場合には、分析のやり直しを行う。

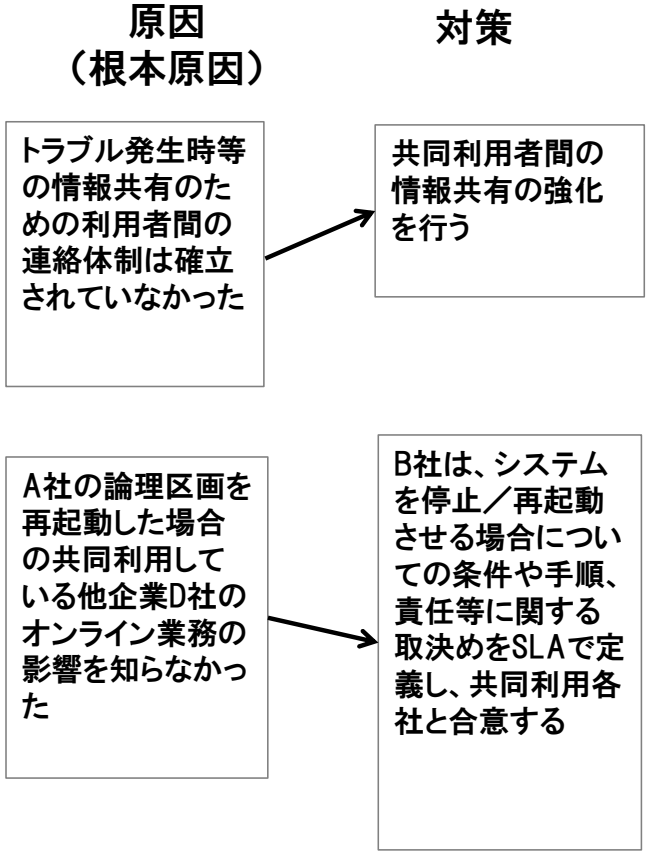


図２．２．２－１　障害事例（なぜなぜ分析の例（原因分析と対策））

【参考１．３．障害事例をなぜなぜ分析で検討した例　の一部】

２．２　ImSAFER

２．２．１　概要

（a）手法の利用シーン

ヒューマンエラーが関係した事象分析手法であり、原因追究と対策立案を支援する。
人間の行動モデルをベースにしているのが最大の特徴である。

（b）手法の概要

ImSAFER は「Improvement for medical System by Analyzing Fault root in human Error incident」の略で、SAFER（Medical SAFER：Systematic Approach For Error　Reduction）を現場が使いやすいように改良したものである。医療分野でのヒューマンエラーを分析する手法ではあるが、IT 分野で起きているヒューマンエラーの分析、対策にも活用できると考えている。（文献 4）

（c）実施手順

以下に、分析と対策を行う実施手順の概略を示す。

手順１：事象の整理

障害が起きた際の事象を整理して、何がどのように発生したのかという事実を把握する。

手順２：問題点の抽出

事象をよく理解して、含まれている問題点を抽出する。それをカードに書き込む。

手順３：背後要因の探索（レベル別）

なぜ、そのような問題点が発生したのかを推定、調査する。分析のレベルを以下の 3 つに分けて、目的やリソースに応じて選択して使う。基本的手法は、「なぜなぜ分析」となる。

・Level Ⅰ：ワンポイントなぜなぜ分析（利用者は個人）
手順 2 で抽出した問題点のついたカードから分析対象行動を選び出し、これだけについてなぜなぜ分析を行う。

・Level Ⅱ：出来事流れ図分析（利用者は部署のリスクマネジャ）
問題カードの中から、事象の流れが把握できるようなカードを選び出し、時間軸にそって縦に並べる。各問題カードについてなぜなぜ分析を行う。

・Level Ⅲ：エラー事象の構造分析（利用者は部門責任者）
最終的に発生した問題から、ロジカルに背後要因を探索する。初めに最も重要と思われる問題点を 1 つだけ選び出してスタートし、なぜなぜ分析で背後要因を探索する。続いて残された問題カードについてもこれを行う。

手順４：考えられる改善策の列挙

この段階では一旦実行可能性を無視し、問題や背後要因をなくす改善策を列挙する。手順 3 のそれぞれのレベルに対応して対策を考える。

手順 5：実行可能な改善策の決定

現実の制約を考え、手順 4 で列挙した中で実施する改善策を評価し、優先順位をつけて決定する。

手順 6：改善策の実施

誰が、いつまでに、どのように、といったことを具体的に決め、実施する。

手順 7：実施した改善策の評価

実施した対策の効果、あるいは、新たな問題点の発生等を評価する。

２．２．２　ImSAFER をシステム障害の分析、対策に活用

IPA では、ImSAFER を IT のシステム障害の分析、対策に使いたいと考えている。そこで、IT のシステム障害、特にヒューマンエラーに起因する障害において「ImSAFER の IT 活用版」として、その手順と考え方を説明する。

実際にシステム障害の分析、対策にどう活用すればよいかを示したプロセスを教訓集の「教訓Ｔ２１：作業ミスを減らすためには、作業指示者と作業者の連携で漏れのない対策を！」の障害事例を使って説明する。

手順 1：事象の整理

教訓Ｔ２１の障害事象は、以下の通りである。

A 社は、顧客の集荷依頼をその顧客の所属するエリアの該当集荷本部へ電話を転送するサービスを行っている。A 社は、B 社の集荷依頼を関係外の C 社集荷本部に転送していたことを、C 社からの連絡で知った。このとき、4 回に 1 回の割合で B 社の集荷依頼を C 社集荷本部に転送していた。

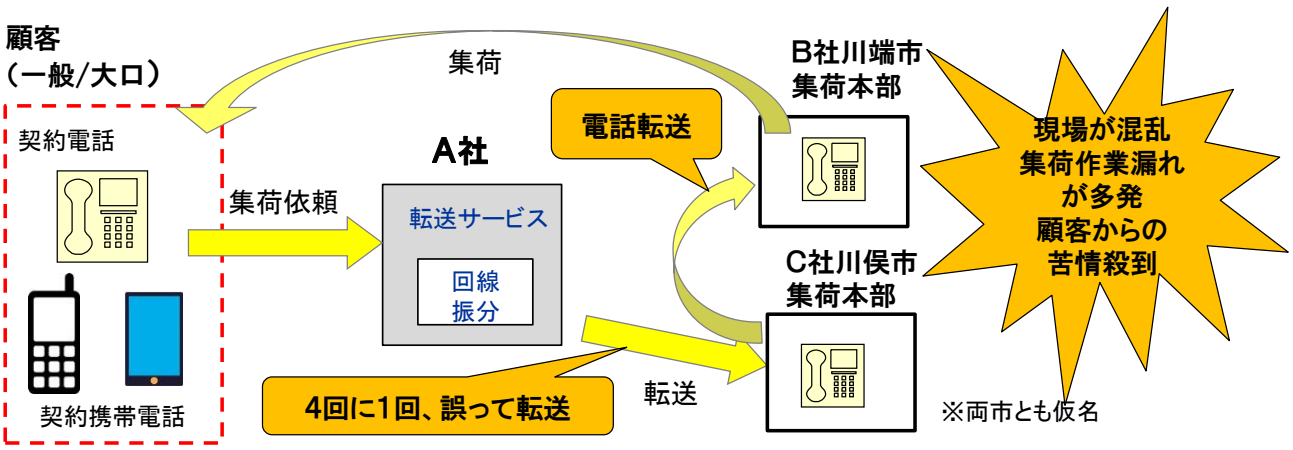


図２．２．２－１　障害状況

手順 2：問題点の抽出

事象をよく理解して、含まれている問題点を抽出する。それをカードに書き込む。

この事例の場合は、「A 社は、4 回に 1 回の割合で B 社の集荷依頼を C 社集荷本部に誤って転送していた。」になる。

手順 3：背後要因の探索（レベル別）

なぜ、そのような問題点が発生したのかを推定、調査する。システム障害が大きな問題になる場合、以下の 2 つの観点が考えられる。

①エラーはなぜ発生したか

システム障害の直接原因は、大きくハードウェア障害、ソフトウェア障害、作業ミスの 3 つに分類されることが多い。

②エラー拡大阻止はなぜ失敗したか

IT システムでは、上記のエラーが発生することを想定して、それを速やかに復旧させる手立てを講じている場合が多い。社会的に重要なインフラとなるシステムでは、特に万全な対策を講じている。したがって、システム障害が大きな問題になる場合は、この拡大阻止対策が失敗していることが問題になる。

また、要因の洗い出しを行う場合、m-SHEL モデルをベースに要因の洗い出しを行うと原因分析に漏れがなくなる。

m（マネジメント）の観点から分析

S（ソフトウェア）、運用の観点から分析

H（ハードウェア）の観点から分析

E（環境）の観点から分析

L-L（人）当事者、支援体制の観点から分析する

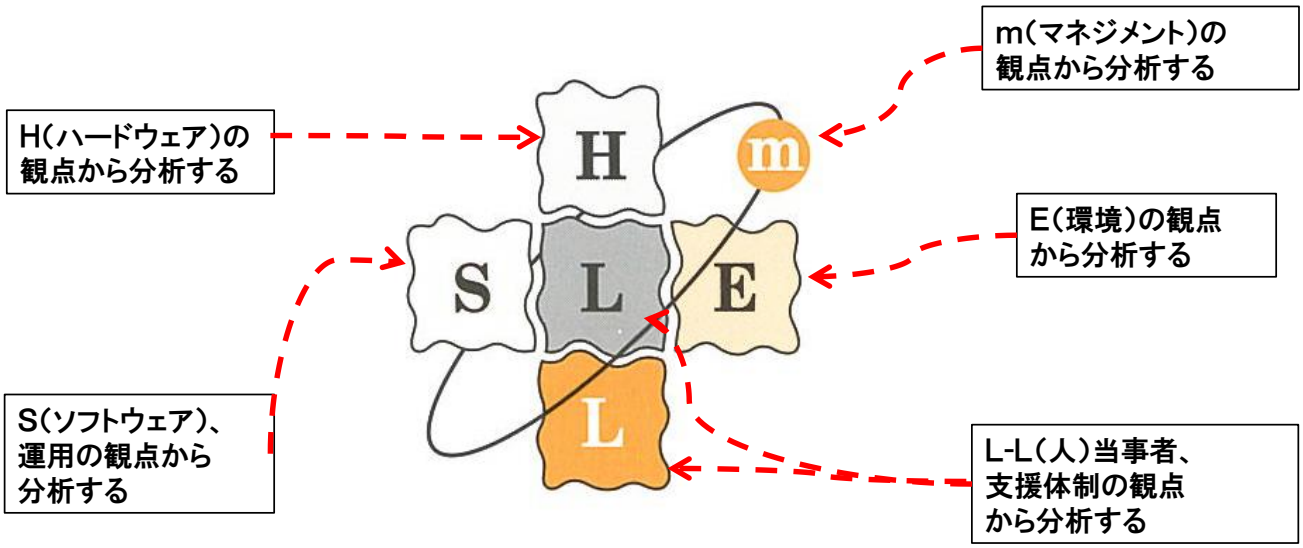


図２．２．２－２　m-SHEL モデル[1]

[1]河野龍太郎「医療におけるヒューマンエラー第 2 版」医学書院 2014　※説明を加筆しています

次に、背後要因を見つけるために、なぜなぜ分析を行う（図２．２．２－３）。直接原因から正しいと判断した事象を選び出し、「なぜエラーを起こしてしまったのか」の「なぜ？」を繰り返しながら、なぜその事象を正しいと判断したのかを考えてみたり、当事者から聞き取りを行ったりしていく。

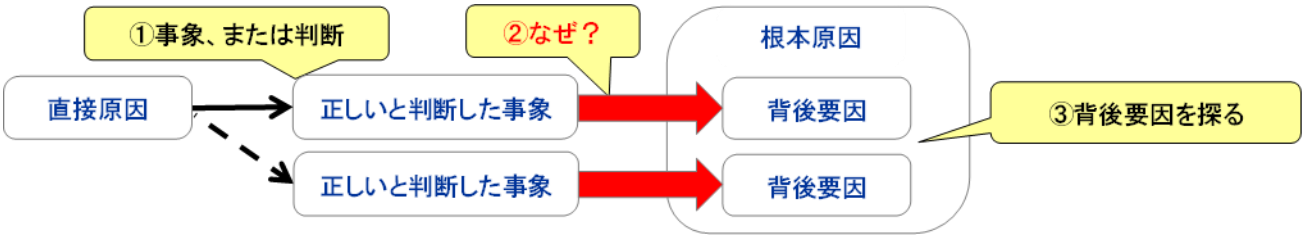


図２．２．２－３　なぜなぜ分析

この障害事例からなぜなぜ分析を行っていくと、以下のような背後要因を見つけることができる。事例では、「A 社は、4 回に 1 回の割合で B 社の集荷依頼を C 社集荷本部に誤って転送していた」直接原因は、以下の内容であった。

A 社は、コール数の増加に対応するため、ゲートウェイ(GW）の増設作業を行った。その時、作業指示者から依頼された作業者は、エリア情報管理サーバの転送先データの登録作業で、転送先名「KAWAHATA（カワハタ)」を「KAWAMATA（カワマタ)」と誤って設定してしまった。

なぜなぜ分析では、この直接原因から、正しいと判断した事象を列挙し、そこから背後要因を探っていくことになる（図２．２．２－４）。

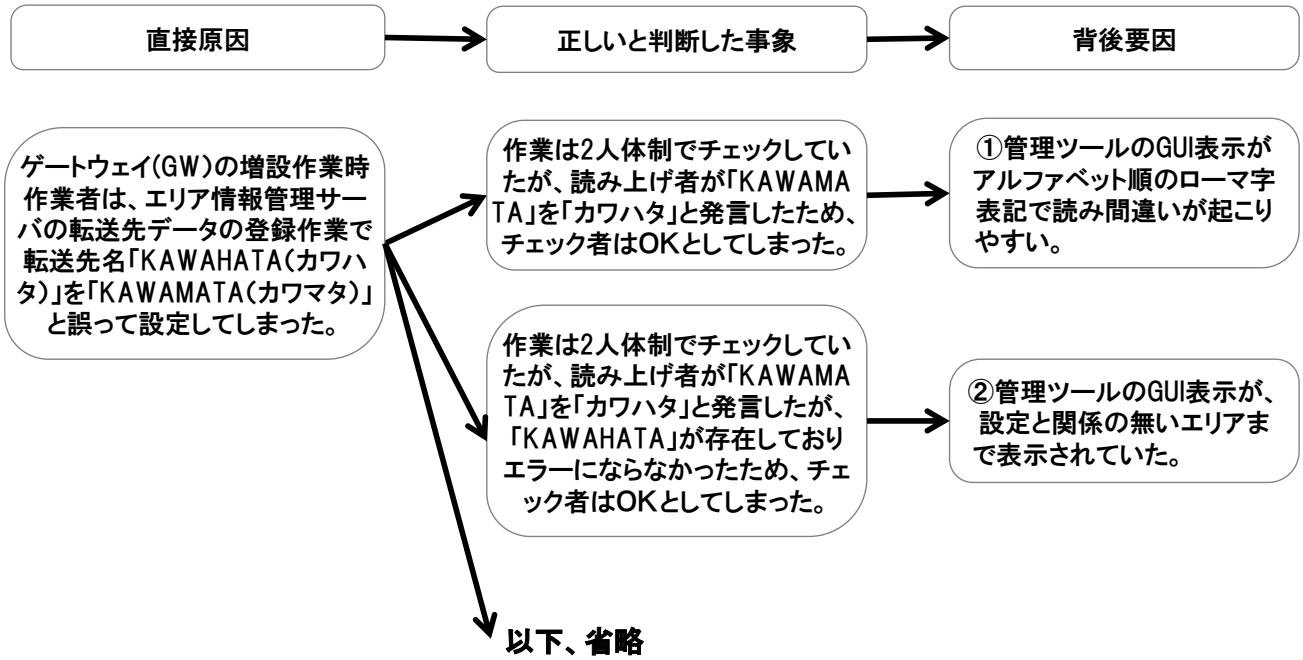


図２．２．２－４　背後要因

この背後要因（上記の①、②）をまとめると、根本原因として、「作業ミスを起こさせるような状況、環境がある」ことが問題であることが分かる。

・パラメータがローマ字で見誤り易い。（背後要因 1）

・設定に関係しないパラメータも表示されている。（背後要因 2）

手順 4：考えられる改善策の列挙

まずは実行可能性を考えず、その問題や背後要因をなくす改善策を列挙する。

対策は、背後要因の裏返しである場合が多いので、背後要因から対策を導き出すことになるが、m-SHEL モデルを活用して、対策の観点に漏れがない対応を考えることも必要である（図２．２．２－５）。

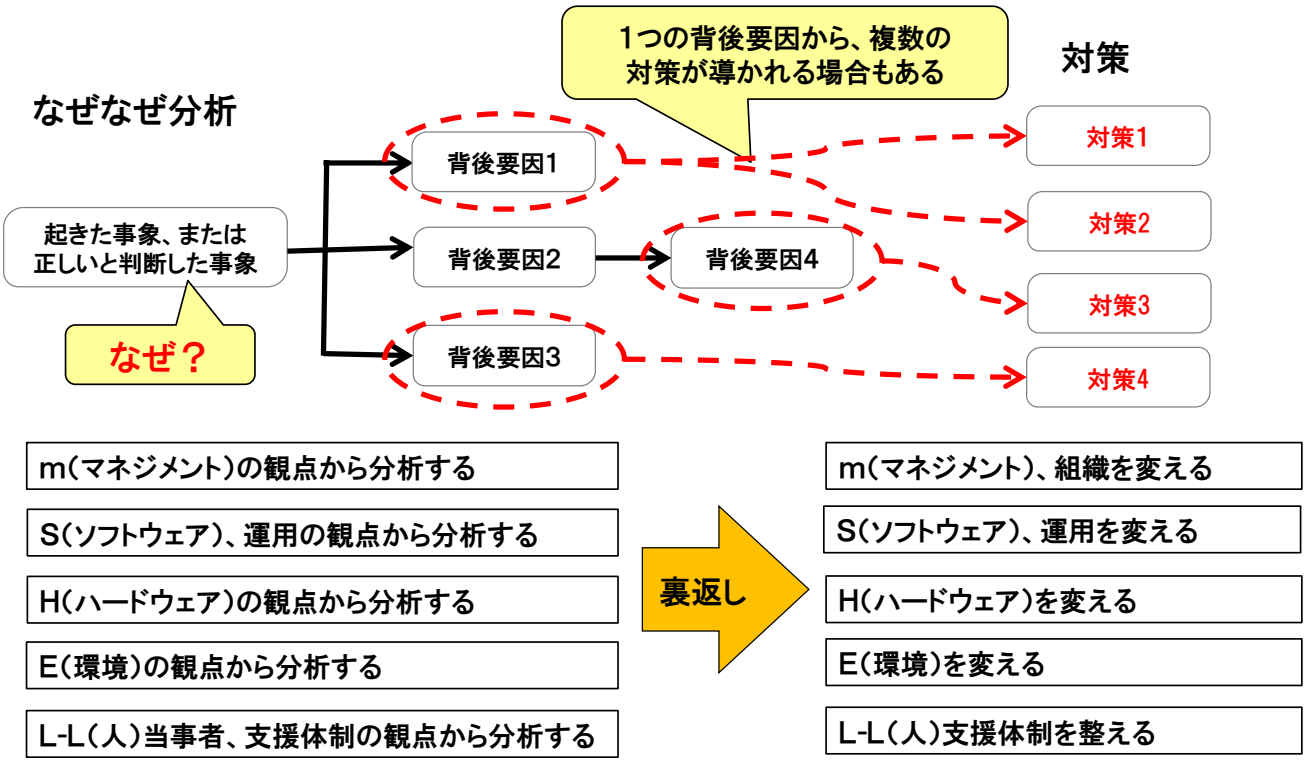


図２．２．２－５　背後要因から対策を考える

m-SHEL モデルでは、m（マネジメント）の観点では、m（マネジメント）、組織を変えることになり、S（ソフトウェア）、運用の観点では、S（ソフトウェア）、運用を変える、H（ハードウェア）の観点では、H（ハードウェア）を変える、E（環境）の観点では、E（環境）を変える、L-L（人）当事者、支援体制の観点では、L-L（人）支援体制を変える、となる。

この障害事例から見つかった背後要因 1、2 を元に対策を考えると、以下のような対策を見つけることができる（図２．２．２－６）。

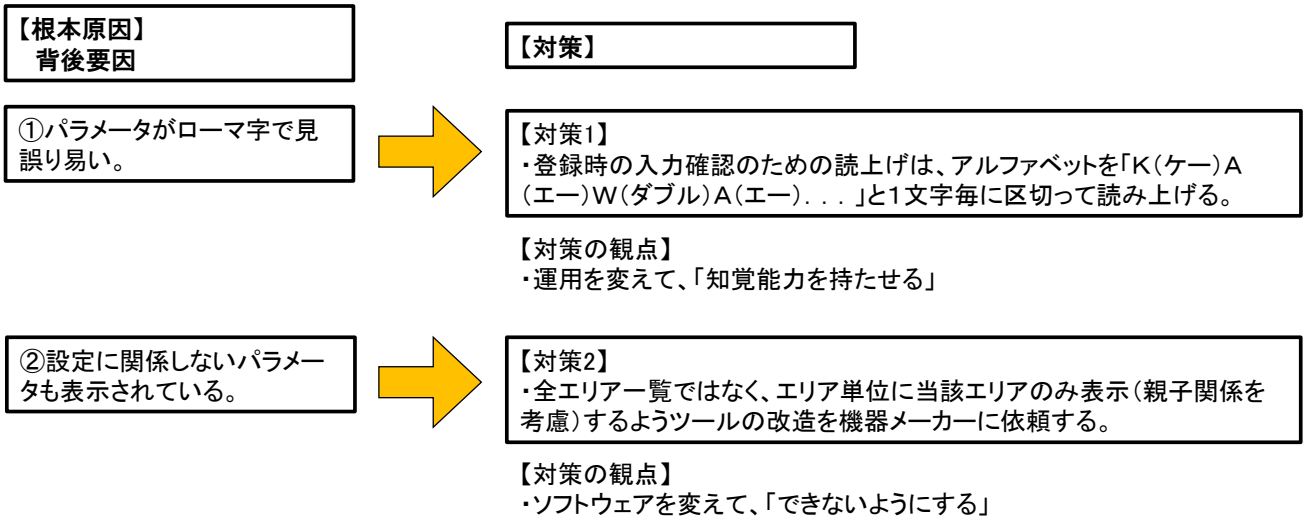


図２．２．２－６　背後要因から導かれた対策

手順５：実行可能な改善策の決定

現実の制約を考え、実施する対策を評価し、優先順位をつけて決定する。

まず、対策を以下のマトリックスで整理することにより、対策に漏れがないことを確認することができる（表２．２．２－１）。

表２．２．２－１　対策一覧

対策分類 ＼ エラー対策の発想手順	作業者環境の対策						作業者自身の対策				
	やめる（なくす、減らす）	できないようにする	分かりやすくする	やりやすくする	検出する	備える	知覚能力を持たせる	認知・予測させる	安全を優先させる	できる能力を持たせる	自分で気づかせる
m（マネージメント）、組織　の対策											
S（ソフトウェア）、運用　の対策		2					1				
H（ハードウェア）　の対策											
E（環境）　の対策											
L-L（人）支援、体制　の対策											

作業者を中心とした関係性の対策分類

作業者を中心に対策をマッピング

対策が無いところは、問題ないか再検討

検討順序　対策の観点

数字は、対策番号

（出典）河野龍太郎「医療におけるヒューマンエラー第2版」医学書院2014　当事例用に変更しています。

この表の対策番号が入っていないところには原因、対策の漏れが考えられるので、その部分を再度確認することで障害の未然予防につながる。またこの表は、対策を考える場合の「気づき」にも活用できる。

この表の横軸は、作業ミスを起こした「作業者」を中心にした m-SHEL モデルでの関係性を示す対策（改善）項目である。この項目が、対策において改善すべき具体的な作業の分類を表す。

また、縦軸は、左から優先的に立てるべき行動を一般的に記述して並べている。対策を検討する上で、効果のある左側の事項から優先して検討することになる。表にも示した通り、IT システム障害では、「作業者環境の対策」を「作業者自身の対策」よりも優先すべきと考えている。

手順 6：改善策の実施

誰が、いつまでに、どのように、といったことを具体的に決め、実施する。

手順 7：実施した改善策の評価

実施した対策の効果、あるいは、新たな問題点の発生等を評価する。

この事例では、ImSAFER の手法を適用してヒューマンファクターズを活用した。このような手法を使った分析、対策は、関係者の合意形成を築き易い。更に、このようなツール活用の継続が、ノウハウの蓄積に繋がる。

一般に作業ミスは、ともすれば作業者の自覚の問題とされる場合が多い。しかし、ヒューマンファクターの観点から、個人、環境、ハードウェア、ソフトウェアの関係性の中で、作業ミスの原因、対策を考えることが重要である。作業を間違えた運用作業者、ソフトウェアのバグ、ハードウェアの故障などに責任を持っていかず、システムの問題を仕組みや組織として改善することに主眼を置くことが重要である。

２．３　RCA

（a）手法の利用シーン

問題や事象の根本原因を明らかにすることを目的として使用される。

（b）手法の概要

・米国退役軍人省（Veterans Affairs：VA）の患者安全センタ（National Center of Patient Safety：NCPS）で開発されたツール。

･VA－NCPS では多くの原因分析手法を調査し、その中に存在するパターンを抜き出した。

・事故の原因を個人の問題だけに終始せず、システムやプロセスに焦点をあてるという特徴がある。

（c）記法 ならびに分析記法（文献 5）

・出来事流れ図の作成

資料をもとに事例では何が起こったのか経過を時系列に並べる。

・なぜなぜ分析の実施

出来事流れ図のひとつひとつの出来事に対して、「なぜ」そのようなことが起こったのか、を回答する。

疑問が無くなるまで実施する。

・因果関係図の作成、根本原因の確定

なぜなぜ分析の結果、明らかになった根本原因候補と問題事象の因果関係を明らかにする。

原因と結果の関連性の見直しによって、分析過程を整理する。

・対策の立案

7 つの留意点をもとに具体的に検討する。

実効性、実施容易性、効果、コスト、効果の持続性、関連部署の業務量、なぜなぜ不足（対策が期待どおりでないと思われる場合はなぜなぜ不足）

２．４　総合的インシデント分析

（a）手法の利用シーン

ベンダ F 社の分析サービス。IT の分野に特化して、従来手法と比べて、リーダに負担の少ない傾向分析と根本原因分析が可能。（文献 6）

（b）手法の概要

・定義

日々発生するインシデントに着目した総合的なインシデント分析手法。

（c）記法 ならびに分析記法

以下の 5 プロセスから構成。

・インシデントを記録する。

（分析に必要な情報を明確にし、プロジェクトの基本行動に）

・インシデントを把握・管理する。

・インシデントから問題をとらえる。

・そもそも分析（表２．４－１参照）を活用して根本原因を見つけ出す。

・インシデントの発生を抑止する改善を提案する。

（インシデントの記述内容にビジネス、利用者の視点を加える）

（d）分析の例

表２．４－１　そもそも分析のテンプレート

そもそも分析のテンプレート

原因分析	問いかけによる原因の発見	対策提案	見つけ出した根本原因をできるに置き換える
そもそも1	決めていない	できる1	決めよう
そもそも2	知らない	できる2	知らせよう
そもそも3	知っていたが実施しない	できる3	知っていたら実施できるようにしよう
そもそも4	決まり事がおかしい	できる4	決まり事を正しくしよう
そもそも5	決まり事が古くなった	できる5	決まり事を刷新し陳腐化させないようにしよう
そもそも6	訓練が足りない	できる6	訓練できるようにしよう
そもそも7	運用しにくい	できる7	運用しやすいように考えて直そう

総合的インシデント分析によるサービスマネジメント領域の強化　2009/11　富士通ジャーナル　を参考に作成

２．５　HAZOP

（a）手法の利用シーン

設計意図からの逸脱によるハザードを明示する手法。

効率的な運転や操作に妨げとなる設計・運転上の意図からの「ズレ」を設定し、そこから想定される潜在的な危険性を定義し評価するための体系的な手法である。

（b）手法の概要

HAZOP（Hazard and Operability Studies）（文献 7）とは、1974 年にイギリスの ICI 社（Imperial Chemical Industries）によって開発された手法である。新しい設計のハザードや以前には考慮されなかったハザードを顕在化することができる。他の大部分の手法が、分析前にハザードを識別する必要があるという点で、HAZOP は他の技法とは異なる。

（c）記法 ならびに分析記法

HAZOP では、以下を考察する。（化学プラントでの導入を想定した考え方）

1. プラントの設計意図
2. 設計意図からの潜在的な逸脱
3. 設計意図からのこれらの逸脱の原因
4. こうした逸脱の結果

このプロセスで使われるガイドワードを、表２．５－１に示す。

表２．５－１　HAZOP のガイドワード

ガイドワード	意味
なし	意図された結果は達成されないが、他には何も起こらない。 （順方向に流れるべき時に何の流れもない）
増加	関連する物理的特性が、あるべき値よりも多い。 （例えば、圧力が高い、温度が高い、流量が多い、粘性が高いなど）
減少	関連する物理的特性が、あるべき値よりも少ない。
他に	意図されたことに加えて、ある活動が発生する、あるいは、あるべきものよりも多くのコンポーネントがシステム内に存在する。 （例えば空気、水、酸、腐食性生成物を含んだ余分な蒸気、固体または不純物など）
一部	設計意図の一部のみ達成される。 （例えば混合物中の 2 つのコンポーネントのうち 1 つだけ）
逆	意図されたこととは論理的に正反対なことが起こる。 （例えば順方向に流れずに逆流）
異なる	意図された結果のどのような部分も達成されず、全く違う何かが起こる。 （例えば誤った物質が流れる）

（d）分析の例

製品制御系／組込系への適用例を示す。（文献 8）

・HAZOP で設計意図からの逸脱（ハザード）を洗い出した例

表２．５－２　HAZOP で設計意図からの逸脱（ハザード）を洗い出した例

出力	期待値からの逸脱状態（ガイドワード）	設計意図と異なる挙動シチュエーション	ハザード
駆動制御の指示情報の出力	不作動	通常運転時	意図しない停止
	勝手に動作	通常運転時	勝手に動作制御
	過大	通常運転時	意図より過大な動作
	過小	通常運転時	意図より過小な動作

Embedded Technology 2013／組込み総合技術展基調講演

「ソフトウェア高信頼性への道程」講演資料（新誠一）を参考に作成

２．６　FTA（フォールトツリー解析）

（a）手法の利用シーン

発生頻度の分析のために、原因の潜在的な危険（フォールト）を論理的にたどり（ここで言う「フォールト」とは、機器の故障やヒューマンエラー等のイベントを指す)、それぞれの発生確率を加算し、基本的な事象が起こりうる確率を算出する。

（b）手法の概要

下位アイテム又は外部事象、若しくはこれらの組合せのフォールトモードのいずれが、定められたフォールトモードを発生させ得るか決めるための、フォールトの木形式で表された解析。FTA ではその発生が好ましくない事象について、発生経路、発生原因及び発生確率をフォールトの木を用いて解析する。

（c）記法 ならびに分析記法

発生頻度の分析のために、原因の潜在的な危険（フォールト）を論理的にたどり（ここで言う「フォールト」とは、機器の故障やヒューマンエラー等のイベントを指す)、それぞれの発生確率を加算し、基本的な事象が起こりうる確率を算出する。

なお、FTA は、望ましくない事象に対しその要因を探る、トップダウンの解析手法を特徴とする。これは、類似の FMEA（故障モード影響解析）とは逆のアプローチになる。

（d）分析の例（文献 8）

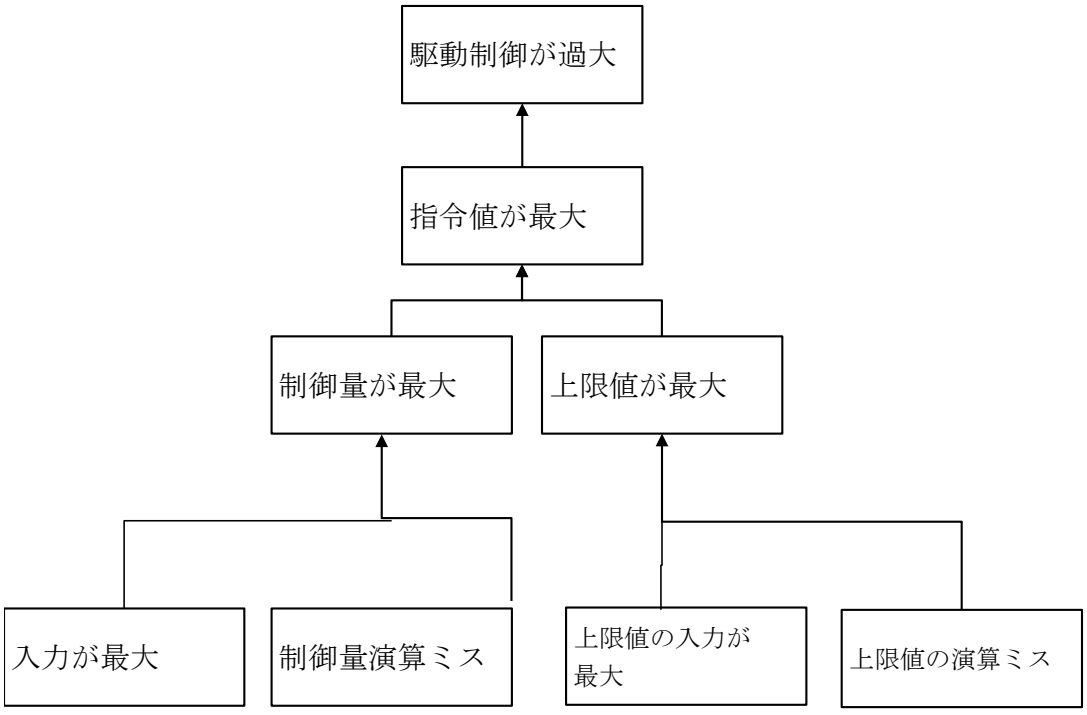


図２．６－１　FTA での分析例

Embedded Technology 2013／組込み総合技術展基調講演

「ソフトウェア高信頼性への道程」講演資料（新誠一）を参考に作成

２．７　FMEA（故障モード影響解析）

（a）手法の利用シーン

設計の不完全や潜在的な欠点を見出すために構成要素の故障モードとその上位アイテムへの影響を解析する技法。

（b）手法の概要

・定義

設計の不完全や潜在的な欠点を見出すために構成要素の故障モードとその上位アイテムへの影響を解析する技法。FTA がトップダウン手法であるのに対し、FMEA はボトムアップ手法という違いがある。FMEA は, FTA,HAZOP, デザインレビューとともに IEC の国際規格になっている。

・特徴

故障モード（英: failure mode）は「故障状態の形式による分類。例えば、断線、短絡、折損、摩耗、特性の劣化等」であり、「故障そのものではなく、故障をもたらす不具合事象の様式分類である。」別の言葉で言えば、製品システムを構成する項目(item)の構造的な(根源的な)破壊をいう。一方、故障とは機能障害である。何もなくただ機能しないということはありえなく、その製品が機能しない原因となる不具合が必ずある。この故障(機能障害)を引き起こした不具合、これが故障モードである。

全く用途も構造も異なる機器でも、電気回路を内蔵している限り、例えば「断線」ということが起こるかもしれない。機器やそのモデルごとに起こりうる故障を全て考えるのは一般的に不可能であるが、故障を引き起こす不具合、つまり故障モードは類型的に分類できる。また、ある製品の新しいモデルがどう故障しやすいか直接予想することは難しい。しかし、断線等の故障モードはどうして起こるか、どれくらい起こりやすいかある程度予想が可能である。それで、故障モードから故障にいたるメカニズムを手繰っていくことで「故障の質的予想を系統的に統一的に行うことが可能になる」。これが故障モードを考える意味である。

製造工程についても故障モードを考えるが(工程 FMEA)、この場合、部品をつけなかった、正しい手順でつけなかったというような、その工程で行うべきと決められていることに違反することが故障モードになる。結果として生ずる不良やトラブルは「影響」であって故障モードとは言わない。

(c) 分析の例（文献 8）

表２．７－１　FMEA での分析例

・構成部品の FMEA を実施し FTA の正しさを検証した例

ソフトウェアコンポーネント	故障モード	要因	処置
制御量算出	演算間違い	演算処理のロジック(プログラム)ミス	テストケース追加
	データ破壊	データ破壊の要件もれ	異常系仕様見直し
	遅延	性能要件もれ	性能仕様見直し
	不正アクセス	アクセス不正の要件もれ	異常系仕様見直し

Embedded Technology 2013／組込み総合技術展基調講演
「ソフトウェア高信頼性への道程」講演資料(新誠一)を参考に作成

２．８　STAMP（Systems-Theoretic Accident Model and Processes）

（a）手法の利用シーン

構成要素が多く、要素間の相互作用が複雑な昨今のシステムに対する安全解析

（b）手法の概要

STAMP（文献 9）（文献 10）（文献 11）とは、米国マサチューセッツ工科大学（MIT）の Nancy.G.Leveson 教授が著書“Engineering a Safer World”の中で提唱したシステム理論に基づく事故モデルである。

Nancy 教授は本書の中で、システムの安全性は構成要素の相互作用から創発されるものであり、個々の要素を分割して分析するべきではないと述べている。そして、現代のシステムのアクシデントの多くは、システム構成要素の故障によって起きるのではなく、システムの中で安全のための制御を行う要素（コントローラー：Controller）と制御される要素（被コントロールプロセス：Controlled Process）の相互作用が働かないことによって起きるというアクシデントモデルを提唱した。このモデルを「STAMP（Systems-Theoretic Accident Model and Processes）：システム理論に基づくアクシデントモデル」と呼ぶ。

（c）特徴

STAMP は従来の解析的な還元や信頼性理論ではなく、システム理論に基づく新たな事故モデルである。従来の事故モデルでは、事故は故障イベントの連鎖によって起こると考えられてきた。しかしながら、近年システムの複雑さが増し、ハードウェアのように「故障」することのないソフトウェアが多く用いられ、事故発生の仕組みが変わったのである。このため STAMP では、事故は単純なコンポーネント故障のみを原因とせず、コンポーネントの振る舞いやコンポーネント間の相互干渉が、システムの安全性制約（コンポーネントの振る舞いに関わる物理的、人、又は社会に関わる制約）を違反した場合に起こると考える。

STAMP は、安全制約、階層的安全コントロールストラクチャー、プロセスモデルという三つの基本要素で構成されており、コントロールストラクチャー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。

- 安全制約とは、安全が守られるために必要なルールを指しており、例えば、ヘルメット、機械の安全装置、災害時の避難ルールなどといったものが挙げられる。STAMP では、安全制約が不適切である、または守られていない場合に事故が起きると定義している。
- プロセスモデルとは、コントローラーが持つアルゴリズムであり、状況に応じて、コントロールアクションを他のコンポーネントに出す仕組みを指している。エアコンの自動温度調節機能を例に挙げると、コントローラーは、一定の温度に達すると冷暖房を動かすプロセスモデルに従って、制御アクションの指示を出す仕組みとなっている。
- コントロールストラクチャーは、コンポーネント間の相互作用を示すコントロールループを積み重ねることで構築することができ、コンポーネント間の機能動作を示したシステムの設計図としての

役割を果たす。コントロールストラクチャーを細かく確認することで、不適切な制御アクションや安全制約、システムの設計ミスといった事故につながる誘発要因を見つけ出すことが可能となる。

STAMP は事故モデルであるため、実際の使用にあたっては STAMP の考え方を使ったハザード分析（安全解析）手法や事故分析のツールを使用する必要がある。現在、STAMP をベースとしたツールとして以下のようなものがある。STPA と CAST が最も一般的に使われており、STPA-sec と STECA は近年考案されたツールとなっている。

・STPA：ハザード分析ツール
・STPA-sec：サイバーセキュリティ等に特化した事故解析ツール
・CAST：事故解析ツール
・STECA：仕様コンセプトのレベルでシステムの安全要件や安全に運用するための情報などのための分析手法

２．８．１　STPA（System-Theoretic Process Analysis）

（a）手法の利用シーン

現代の相互作用が複雑なシステムにおける安全解析

（b）手法の概要

システムアクシデントの根本原因を機器の故障や人間のオペレーションミスに置く、従来のアクシデントモデルでは、システムのアクシデントの可能性が潜在している状態（すなわちハザード）とその要因を事前に分析するための安全解析手法として、FTA（Fault Tree Analysis）や FMEA（Failure Mode and Effects Analysis）の手法が用いられてきた。しかしながら、これらの手法は、現代の相互作用が複雑なシステムにおける安全解析手法としては十分ではなく、新しいアクシデントモデルによる分析手法が必要とされた。Leveson 教授が提唱した STPA（System-Theoretic Process Analysis）は、STAMP アクシデントモデルを前提として、システムのハザード要因を分析する新しい安全解析手法である。

（c）特徴

安全解析手法である STPA は、トップダウン式でシステムの分析を行う方法となっており、システム全体から目標とする箇所まで少しずつ焦点を当てていく形となっている。STPA の導入には大きく分けて、4 つのステップ（準備 1、準備 2、STPA Step1、STPA Step 2）が必要となっている。準備 1 と 2 は、CAST 等でも同様に使われる事前準備の段階となっており、STPA Step1 と STPA Step 2 が STPA のハザード分析に重要なステップとなっている。

準備 1：アクシデント、ハザード、安全制約の識別

準備段階の最初のステップとして、分析で扱うアクシデント、ハザード、安全制約の 3 つを設定する必要がある。これらは、システムが回避すべき事象を事前に設定することで、目的に沿ったハザード分析を行うためのものであり、STPA Step1 で使用することとなる。アクシデントとハザード、安全制約の意味は下記の通りであり、図表２．８．１－１は、それらの例となっている。

・アクシデント：喪失（Loss）を伴う、システムの事故。
・ハザード：アクシデントにつながるシステムの状態。
・安全制約（Safety Constraint）：システムが安全に保たれるために必要なルール。

図表２．８．１－１　アクシデント、ハザード、安全制約の例

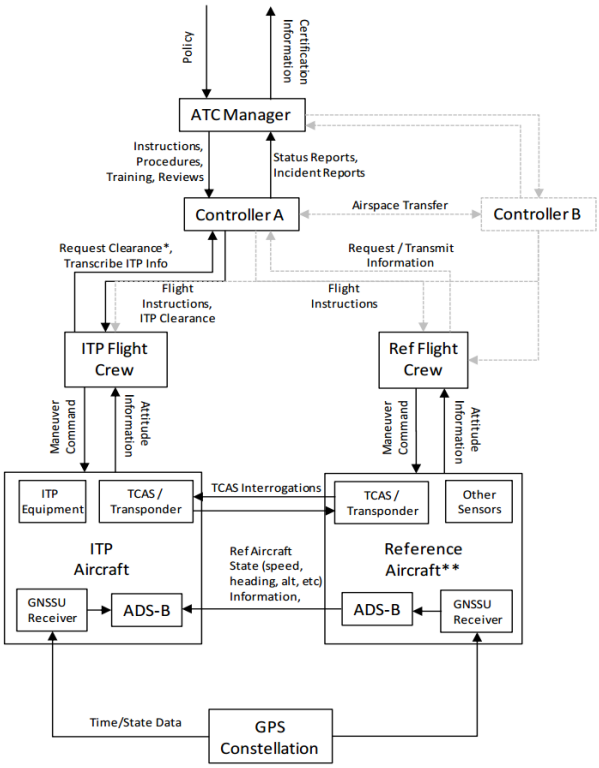
システム	アクシデント	ハザード	安全制約
クルーズコントロール	2 台の車が衝突する	自動車の前後に適切な車間距離をとっていない	自動車は定められた車間距離を破ってはならない
化学プラント	化学物質の流出によって人的被害が出る	化学物質が空気中や土壌へ流出する	化学物質が意図せず放出されてはならない
列車のドア	乗客が列車から落ちる	1. 列車の発車時にドアが開く 2. 列車の走行時にドアが開く 3. 緊急時に列車のドアが開かない 4. ドア付近に人がいるのに、ドアが閉まる	1. ドアが開いている時に列車は動いてはならない 2. 列車が動いている時はドアが開いてはならない
無人宇宙船	ミッションの失敗 他の惑星への汚染	1. ミッション完了時に科学的なデータが失われている 2. 地球由来の物質によって､他の惑星が汚染される 3. 他のミッションで必要な設備が使用不能になっている 4. 打ち上げ後に毒性のある物質や放射能物質などの危険な物質が､地球上か宇宙ステーション付近に残される。	

出典：An STPA Primer(MIT)を基に作成

準備 2：コントロールストラクチャーの構築

コントロールストラクチャーは、システムを制御する各機能の相関関係を示した図であり、コンポーネント間でやり取りされる制御の指示やフィードバック等を矢印で結んで表す。図表２．８．１－２は、コントロールストラクチャーの例となっている。この例は ITP（In-Trail Procedure）と呼ばれる航空機の追い越し手順に対するものである。

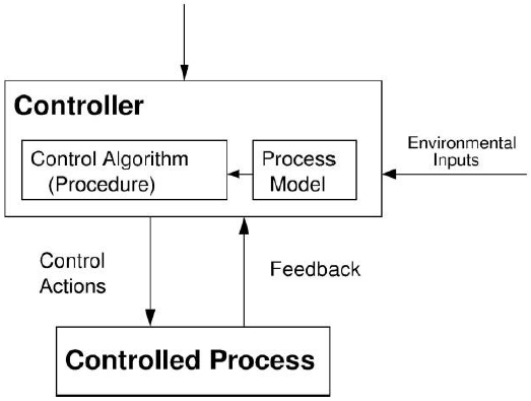
図表２．８．１－２　コントロールストラクチャーの例



出典：An STPA Primer(MIT)

図中、ボックスで示された各コンポーネントはコントローラー（Controller）と呼ばれ、その機能や役割に応じて階層構造になっている。上位のコントローラーからはコントロールアクション（Control action）と呼ばれる指示が出され、センサーなど下位のコントローラーからはフィードバック（Feedback）として情報が送られる。さらに各コントローラーには、そのコントローラーがどのような処理や指示を行うかというプロセスモデル（Process model）が含まれており、特に人間が行うプロセスモデルはメンタルモデル（Mental model）と呼ばれている。これらのコントローラー間のやり取りを表したものをコントロールループ（Control loop）と呼ぶ。図表２．８．１－３に、コントロールループのモデルを示す。

図表２．８．１－３　コントロールループのモデル



STPA Step1：安全でないコントロールアクション（Unsafe Control Action：UCA）の識別

STPA の最初の段階として、安全でないコントロールアクション（Unsafe Control Action：UCA）の識別を行う必要がある。UCA の識別は、ハザードにつながる恐れのあるコントロールアクションの不具合を明確にすることを目的としており、大きく分けて以下の 4 つの種類に分類される。

1. 安全のためのコントロールアクションが設置されていない。
2. ハザードにつながる恐れのある、安全ではないコントロールアクションが設置されている。
3. コントロールアクションのタイミングが遅すぎる、早すぎる、または定められた順序に設置されていない。
4. コントロールアクションがすぐに止まる、もしくは適用が長すぎる。

この 4 つ以外に 5 番目のシナリオとして、「要求されたコントロールアクションが設置されているが、それに従っていない」という UCA が考えられる。この原因には、コントロールループ内での不具合や遅れなどの不適切な動作が含まれる。5 番目に関しては、STPA Step2 の中で分析していくことになる。

上記 5 つのシナリオが、人間やコンピュータによる行動に関連した事故原因をより的確に表したモデルとなる。

図表２．８．１－４に、UCA の例を示す。このような表を使って整理していくと、システムに潜む UCA の識別と関連するハザードの種類などを整理しやすくなる。また、必要に応じて図表２．８．１－５のようなアクシデント、ハザード、UCA を整理した表を作成することもできる。この例は宇宙ステーション補給機（H-II Transfer Vehicle: HTV）「こうのとり」のシステムにおける STPA 適用で、国際宇宙ステーション（ISS）のロボットアームによる把持（キャプチャ）フェーズに適用したものである。図表中の FRGF（Flight Releasable Grapple Fixture）は取り外し可能型グラプルフィクスチャのことで、ISS のロボットアーム（SSRMS）の把持部である。

図表２．８．１－４　安全でないコントロールアクション（Unsafe Control Action：UCA）の例

#	コントロールアクション	「Not Providing」 がハザードを引起す	「Providing」 がハザードを引起す	「Wrong Timing / Order」 がハザードを引起す	「Stopping Too Soon/Applying Too Long」 がハザードを引起す
1	*FRGF 分離イネーブル*	[UCA1]キャプチャ準備完なのに FRGF 分離がイネーブルされていない	[UCA2]不必要なときに FRGF 分離がイネーブルされる	早：[UCA3]直ぐにキャプチャ可能でないのに FRGF 分離がイネーブルされる	
2	*制御停止*（非活動化）	[UCA4]キャプチャ準備完なのに HTV が停止していない	[UCA5]適切でないのに HTV が停止している（例えば、ISS に接近中に）	早：[UCA6]直ちにキャプチャ可能でないのに HTV が停止している 遅：[UCA7]直ぐにキャプチャ可能でないのに FRGF 分離がイネーブルされているのに長時間 HTV が停止していない	
C	キャプチャ実行	[UCA8]HTV が停止している間にキャプチャが実行されない	[UCA9]HTV が停止していないのにキャプチャが試みられる [UCA10]SSRMS が不注意に HTV に衝突する	早：[UCA11]HTV が停止する前にキャプチャが実行される 遅：[UCA12]ある時間内にキャプチャが実行されない	[UCA13]キャプチャ操作が途中で停止し、完了されない
3	*FRGF 分離防止*	[UCA14]キャプチャ成功後に FRGF 分離が防止されない	[UCA15]イネーブルされねばならない時に FRGF 分離が防止されている（例えば、キャプチャが試みられている時に）	遅：[UCA16] キャプチャ成功後に FRGF 分離防止が遅すぎる	
オフノミナル	*強制退避 一時後退 相対位置保持*	[UCA17]強制退避/一時後退/相対位置保持が、必要なときに実行されない（例えば、HTV が制御できず ISS に向かってドリフトしている時）	[UCA18]強制退避/一時後退/相対位置保持が、適切でないのに実行される（例えば、キャプチャ成功後）	遅：[UCA19]強制退避/一時後退/相対位置保持が直ちに必要なときに、遅すぎて実行される（例えば、HTV が制御できず ISS に向かってドリフトしている時）	
	FRGF 分離	[UCA20]必要なときに FRGF 分離が実行されない（例えば、HTV が安全でなく掴まれた時）	[UCA21]必要でないときに FRGF 分離が実行される（例えば、キャプチャ成功後）	遅：[UCA22]FRGF 分離が直ちに必要なときに、遅すぎて実行される（例えば、HTV が安全でなく掴まれた時）	

出典：An STPA Primer(MIT)

図表２．８．１－５　アクシデント、ハザード、UCA の例

アクシデント		ハザード		UCA
A1	ISS との衝突	H1	HTV が制御できず ISS に向かってドリフトしている（非活動化）	5, 6, 8, 12, 17, 19
		H2	キャプチャ成功後に、意図せず HTV が SSRMS から分離される	2, 14, 16, 21
A2	SSRMS へのダメージ	H3	SSRMS に近接して HTV が意図しない姿勢制御を行う	4, 9, 11
		H4	SSRMS に近接して HTV が大きな角度で傾く	10
		H5	HTV が安全でなく掴まれた時に、直ちに分離できない（例えば windmill）	1, 13, 15, 20, 22
		H6	HTV が、SSRMS にキャプチャされている時にスラストする	18, 20, 22
A3	HTV ミッションの喪失	H7	キャプチャの前や最中に FRGF が意図せず HTV から分離される	2, 3, 7, 21

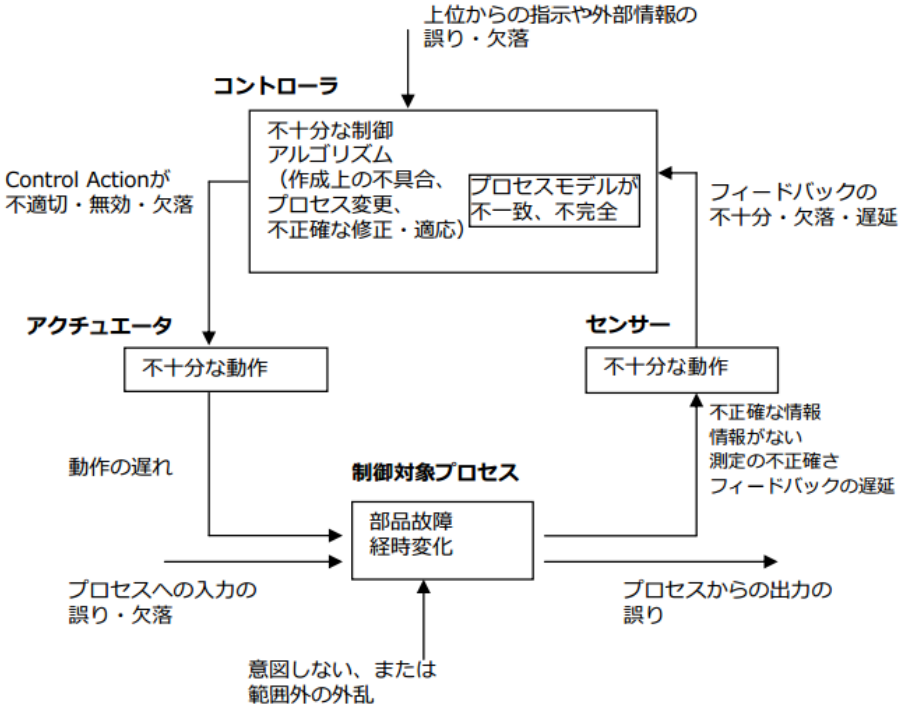
STPA Step2：Causal factor（潜在要因）の特定

STPA の最後の段階として、STPA Step1 で識別した UCA の原因となる Causal factor と、予想される事故シナリオの特定を行う。ここで、前述した 5 番目のシナリオを検討することになる。Causal factor の特定には、UCA の引き金となる 11 個のガイドワードを使って、コントロールストラクチャー内の各コントロールループを分析していく。このプロセスでは、UCA に対してガイドワードの適用を行い、どのようにハザードが発生するかという部分にまで分析を進める必要がある。

Step1 での 4 つの標準的な分類(ガイドワード)をコントロールストラクチャーに適用し、図表２．８．１－３のようなコントロールループの基本コンポーネントを調べて、誤った操作を分類し、その標準的な不適切な制御の原因となりうるかどうかを決定する。

図表２．８．１－６は、11 個のハザード要因(ガイドワード)の分類とコントロールループモデルとの関係を表している。

図表２．８．１－６　11 個のガイドワード



STPA Step1 の 4 つのガイドワードがハザードにつながる恐れのあるコントロールアクションの分類であるのに対して、11 個のガイドワードはコントロールループの流れにおいて予想される不備を示したものとなっている。

２．８．２　CAST（Causal Analysis using STAMP）

（a）手法の利用シーン

事故分析

（b）手法の概要

STAMP に基づく事故分析手法。

（c）特徴

システム理論に基づく原因分析手法である CAST（Causal Analysis using System Theory）は事前分析手法である STPA とは異なり、STAMP 事故モデルの考えに基づいた事後分析手法である（文献 10）（文献 12）（文献 13）（文献 14）。CAST は、事故全体のプロセスとシステマティックファクターの理解を可能とするフレームワークとプロセスを提供する。事故の原因と将来の予防にフォーカスし、先入観や偏見が影響して偏った分析をできる限り小さくする事故分析技術である。

CAST の決定すべきゴールは、人々が事故を起こした理由や事故発生を許した安全コントロールストラクチャーの弱点を明らかにすることである。具体的には各コンポーネントにおいて、安全制約、発生した非安全なコントロールアクション、その行動の前後関係に基づく理由、それを引き起こしたメンタル（プロセス）モデルを明確化していく。なお、コントローラーには制御するコンポーネントが認識するシステムや外部環境の状態を表すプロセスモデルが含まれており、特に人間が行うプロセスモデルはメンタルモデルと呼ばれている。

事故分析ツールである CAST は事故に関連した安全制約と各レベルのコントロールストラクチャーを分析することで、様々な観点から分析を行い、事故の要因がどの時点から発生しているかわかるようになっている。

事故分析に一般的にも用いられているなぜなぜ分析と CAST 分析の違いを考えると、なぜなぜ分析は局所的に原因自体の深ぼりとなることが多い。一方、CAST 分析は事故のあった物理的なコンポーネントに対する他のコンポーネントの要素を洗い出せる空間的な広がりのある原因分析である。運用作業者のミスなど、直接の原因に焦点が当たりがちな傾向性を回避し、相互作用の検討の中で見逃されがちな原因を見出せる特徴をもつ。また、コントロールストラクチャー図により、どのコンポーネントがどのような相互作用を与えているから事故が起きた、という流れを可視化し、さらに責任と権限を可視化する手法である。

（d）実施手順

「Engineering a safer world」（文献 10）によると CAST の分析は、CAST.1 から CAST.9 までの以下の手順になる。この分析手順は必ずしも一つが完了してから次のステップへというように逐次に実施されることを意味するものではなく、適宜統合的に実施可能である。CAST.4 から CAST.9 は CAST 独自の手順であるが、CAST.1 から CAST.3 までの三つのステップは STPA と同じものである。

以下は CAST の分析を進める手順となっている。基本的には STPA の手順と同じだが、STPA との大きな違いはコントロールストラクチャーの構築が下位レベル（物理レベル）から上位レベル（システムレベル）へと構築していく点である。事故の直接的な原因から分析を始め、関連した上位レベルのシステムへ

と範囲を広げていくため、最初から全体のコントロールストラクチャーを設計する必要がない。

【分析手順】

CAST 1. 損失に関連するシステムとハザードを識別する

CAST 2. ハザードに関連したシステムの安全制約やシステム要求を識別する

CAST 3. ハザードを制御し安全制約を課すよう整備されている安全コントロールストラクチャーを記述する。*1

CAST 4. 損失につながる近接したイベントを決定する

CAST 5. 損失を物理レベルで分析する *2

CAST 6. 安全コントロールストラクチャーの上位レベルに移り、如何にして、そして何故、より上位のレベルが現在のレベルにおける不適切な制御を許したかもしくは寄与したかを決定する

CAST 7. 損失に関与した共同作業、コミュニケーションの寄与者すべてを調査する

CAST 8. 損失に関連するシステムと安全コントロールストラクチャーの時間経過による動的な特性や変化、および安全コントロールストラクチャーの長期間での弱化を正確に定める

CAST 9.改善勧告を出す

*1)これはコントロールとフィードバックの実行と同様に各コンポーネントの構造上の責任と権限を含む。このステップは以降のステップと並行して実施できる。

*2）このステップは以下を含む。

・発生した事象に対する次のものの寄与を識別：物理的、運用的な操作、物理的な障害、機能が損なわれた相互作用、コミュニケーション、共同作業の欠陥、処理されなかった外乱

・損失を防止する際に何故、物理的なコントロールが効果的でなかったかを定義すること

【コンポーネントごとの記述事項】

またCASTのコントロールストラクチャーにおける各コンポーネントの記述は一般的に以下を含む。

安全要求と制約

コントロール[発生した非安全なコントロールアクション]

前提：[意思決定がされた状況（コンテキスト)特定]

-責任と権限

-環境や行為形成の要素

機能が損なわれた相互作用、故障、ミスコントロールアクションをひきおこす欠陥のある決定

欠陥のあるコントロールアクションと機能が損なわれた相互作用の理由：[プロセスモデル（メンタルモデル）の不備]

-制御アルゴリズムの欠陥

３．IT サービスへの原因分析手法の適用

・IT サービスにおける原因分析手法の現状

原因分析には様々な方法があり、各社で工夫した適用が見られるが、全般に、IT サービスでは、「なぜなぜ分析」が広く活用されている。これ以外の手法として、製品制御系で使われている設計と危険の解析や発生頻度を調べる手法（HAZOP, FTA, FMEA）に相当するような手法の活用はあまり見られない。

・原因分析手法を改善する取組み

製品制御系で多く使用されていて、非常に信頼性が高いシステムに利用されている手法で、原因分析の前に特性要因図を作り全体像を把握し、FTA で事前に予測し、FMEA でそれを詳細化するとシステムの信頼性が向上する。しかし、製造業等で、IT サービスにこのような手法を適用した事例はあるが成功した事例は少ない。また、適用する場合にはコストや期間も増加するので、システムのレベル分けを行って、高信頼を要求されるものに限定して適用する必要がある。

これに対して、STAMP（STPA、CAST）は上記の分析手法の長所を取り入れつつ適用範囲の問題点を解決する手法である。従来は、製品制御系主体だった手法を、対象をソフトウェアやインタフェースを含めたシステムに拡張し、IT サービスにも使えるようにしたものである。

・ IT サービスへの原因分析手法の適用

「なぜなぜ分析」は簡単に利用でき、問題発生時の原因分析に役に立つ手法であるが、ある程度対象を本格的に分析したい場合には、STAMP 等の新たな原因分析手法も目的にあわせて考え、現場で積極的に活用し、まずは小規模でよいので適用／評価していくべきである。

参考：障害分析事例

以下に障害発生を元に原因を分析した事例を示す。

ここに示す事例は、「情報処理システム高信頼化教訓集 IT サービス編」(文献 15)に掲載された教訓 G7「クラウド事業者と利用者が連携した統制がとれたトラブル対応体制を整備すべし」の作成に至る分析の過程を示す。

なお、本項で示す「教訓」とは、障害事例の分析・再発防止策の検討を通して得られた再発防止に向けた「学び」を、他者に伝わりやすいようにコンパクトにまとめたものである。

以下、参考１．１に実際の障害事例の概要（システム概要、障害の概要、発生した事象）を示す。

これをもとに時系列に登場人物をのせて障害の状況を整理したのが参考１．２である。

この中から問題として A 社のオンラインサービスが丸一日間停止したことをピックアップして、参考１．３でなぜなぜ分析を実施し、根本原因７つに対して対策方法を検討した。

これら対策方法のうち１つを基にして作成したのが参考１．４に記載した教訓の例である。IPA では参考１．３のなぜなぜ分析から参考１．４に記載した件を含めて７つの教訓を作成・公開しており、参考１．３ではそれぞれ公開済みの教訓 ID[2]との対応を記載した。

参考１．１　障害事例

以下に、障害発生の概要を示す。

参考１．１．１　システム概要

A 社はオンラインによる情報登録および情報照会の基幹業務システムを当初はオンプレミスで運用していたが、運用コストの削減を目的に複数企業間の共同利用を進める方針となり、B 社が提供するクラウドサービスに移行した。同時期に共同利用に移行するのは他に D 社があり、類似のビジネスを行っていた。B 社が提供するシステムは、業務システム用のサーバと負荷分散装置に分かれている。業務システムのサーバだけでなく、負荷分散装置も仮想化されており、それらの論理区画のうち一つをＡ社は利用していた。(図　参考１．１－１．システムと障害の概要)

A 社：今回の障害が発生したサービスのユーザ

B 社：クラウドサービスを提供するベンダ

C 社：B 社が採用した負荷分散装置のベンダ

D 社：サービスの共同利用ユーザで、今回の障害の影響はなし

参考１．１．２　障害の概要

ある日、オンライン開始時からこのシステムに障害が発生して丸１日業務が停止した。基幹オンラインシステムが端末から起動できず、すべての窓口でデータベースの更新を伴う処理の受付けができなかった（①)。

なお、A 社があらかじめ用意していたクラウド外の「障害時バックアップシステム」に切り替わり、データ照会処理はできたので、その日はデータの更新を伴わないサービスのみを実施した（②)。

[2] 情報処理システム高信頼化教訓のリンク集（IT サービス編）　http://www.ipa.go.jp/sec/system/lesson.html

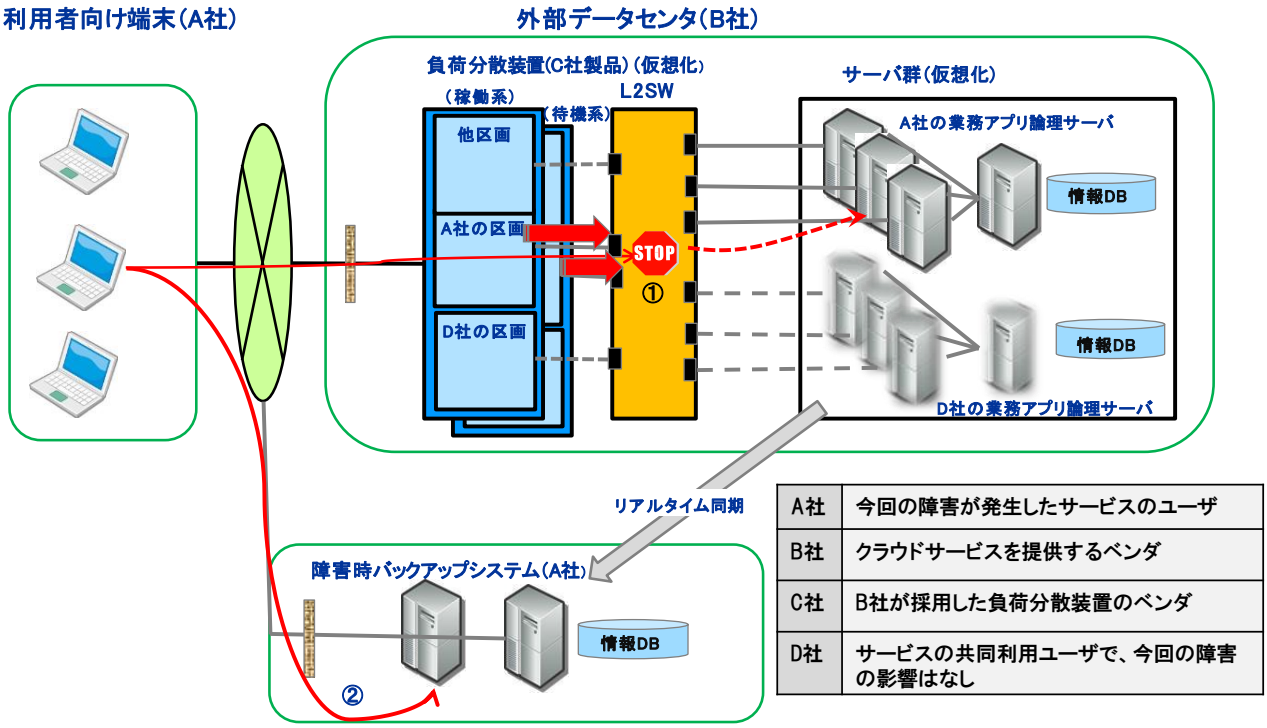


図　参考１．１－１．システムと障害の概要

参考１．１．３　障害の詳細説明

午前４：００

- 負荷分散装置のファームウェアで行っているある処理（sod プロセス）にてメモリ不足エラー（out of memory）が発生した。待機系がスタンバイからアクティブへ切り替わり、この段階で未だ稼動系がアクティブであったため、両系間で多数の電文が繰返し転送される現象（系間ループ形成によるマルチキャストストーム）が発生した（図　参考１．１－２）
- Ｌ２スイッチのポートが閉塞した
- sod プロセスが再起動された
- 稼動系がスタンバイへ切り替わった
- 系切替え（フェールオーバ）動作が完了し、待機系に切り替わったが、待機系自体が out of memory に近い状態であったため、極端なレスポンス悪化が発生した。
- 通信路の疎通状況を確認する ping が通ったことから、B 社はシステムの運用に問題ないと誤認した。

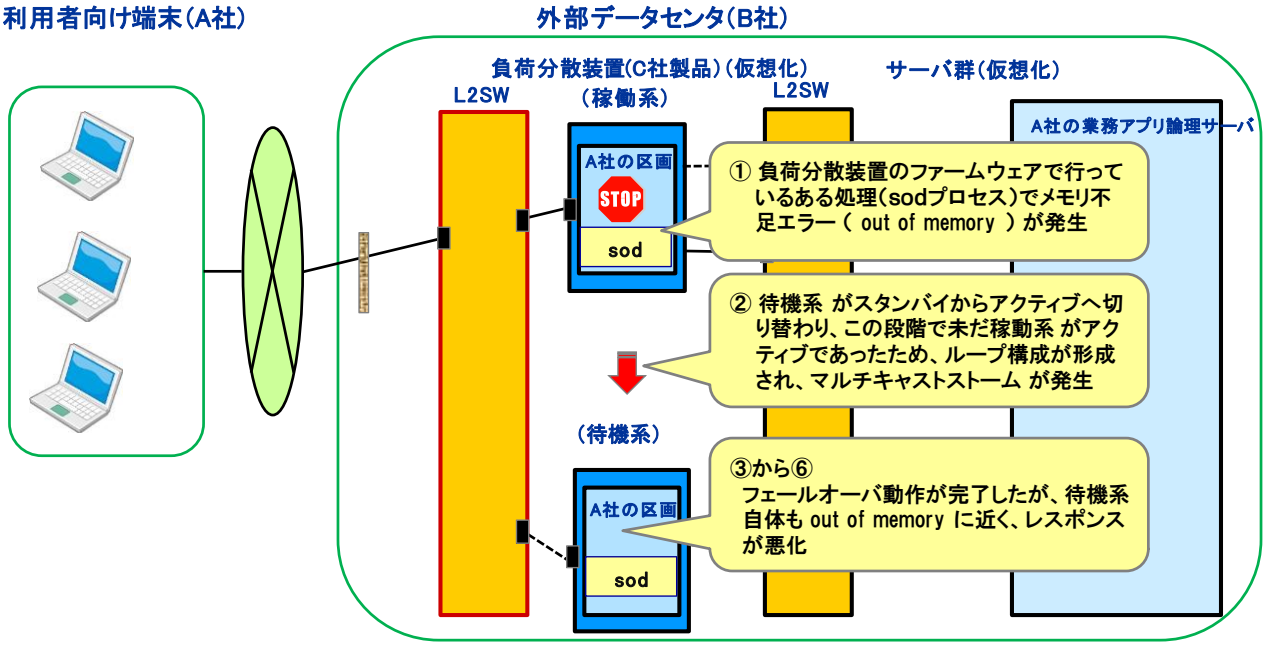


図　参考１．１－２　障害の詳細説明

８：００

- A 社の運用オペレータから情シス部門と B 社の SE に基幹システムのオンラインが起動できない旨の第一報が入った。B 社の SE は基幹システム端末機からオンラインが起動できないことを確認し、障害原因の特定・復旧作業に入った。

８：３０

- A 社の情シス部門は、業務ポータルに障害情報を掲載し、ダウン時対応システムの起動を周知した。

１０：３０

- A 社の情シス部門は、障害原因の特定が出来ず、復旧に時間を要すると判断し、ホームページ

と SNS に障害情報を掲載した。

１１：００

- B 社は初めのうちは AP サーバや専用線の問題と誤認し調査を行い、B 社の自社製品ではない C 社製負荷分散装置の障害調査は後回しにした。
- A 社は各方面への説明対応に追われた。
- この状況は D 社には全く伝えられていなかった。
- A 社の業務窓口は、登録系業務の処理の対応手順がわからなかったため顧客の登録・変更申請に対応できなかった。結果として、窓口に来た顧客に帰って頂く対応となった。

１６：００

- A 社の情シスと B 社の SE は、障害発生箇所を通信関連機器（負荷分散装置）にほぼ特定したが、同日中の復旧が困難と判断し、ホームページと SNS に情報を掲載した。
- 負荷分散装置は D 社と共用しており、再起動等の対処による D 社サービスへの影響が不明のため、A 社の了解のもと対処を業務終了後まで先送りすることとした。

２０：００

- 障害の原因は C 社製負荷分散装置の sod プロセスのメモリ資源が時間とともに増加するという既知の不具合によるものであったことがわかった。
- 障害の原因を負荷分散装置と特定し、試みに A 社の仮想負荷分散装置を再起動したところ、障害が復旧した。

翌 AM１：３０

- 負荷分散装置のハードウェア構成全体の再起動を行い、正常稼動の確認が完了した。

８：００

- ホームページと SNS に障害復旧の情報を掲載した。

参考１．１．４　特記事項

- B 社は、システム構成機器の修正情報の収集間隔を、３ヶ月に１回程度と非常に粗く設定していた。
- A 社のシステムでは、本稼動以来、負荷分散装置は 8 か月以上連続運転状態であり、一般的なネットワーク機器と同様に再起動をしたことがなかった。
- 今まで障害が発生したことが殆どなかったこともあり、システムが使えないと業務遂行はお手上げの状態であった。基幹業務システムが利用できない場合の事務マニュアルはなく、業務部門と情シス部門の対策検討もされていなかった。

参考１．２　障害事例の状況の整理例

参考１．１の障害事例を時系列・部門別に整理したものを図　参考１．２―１に示す。

日時	顧客	A社		B社	C社	D社
		今回の障害が発生したサービスのユーザ		クラウドサービスを提供するベンダ	B社が採用した負荷分散装置のベンダ	サービスの共同利用ユーザで、今回の障害の影響はなし
		業務窓口	情シス部門			
			負荷分散装置のファームウェアで行っているある処理（sodプロセス）にてメモリ不足エラー（out of memory）が発生した。			
朝4時			待機系がスタンバイからアクティブへ切り替わり、この段階で未だ稼動系がアクティブであったため、両系間で多数の電文が繰返し転送される現象（系間ループ形成によるマルチキャストストーム）が発生した			
			L2スイッチのポートが閉塞した			
			sodプロセスが再起動された			
			稼動系がスタンバイへ切り替わった			
			系切替え（フェールオーバ）動作が完了し、待機系に切り替わったが、待機系自体がout of memoryに近い状態であったため、極端なレスポンス悪化が発生した			
朝8時		オンライン開始時からこのシステムに障害が発生してまる1日業務が停止した。基幹オンラインシステムが端末から起動できず、すべての窓口でデータベースの更新を伴う処理の受付けができなかった				
				B社は障害箇所の特定に時間を要した		
				通信路の疎通状況を確認するpingが通ったことから、B社はシステムの運用に問題ないと誤認した		
				初めのうちはAPサーバや専用線の問題と誤認し調査を行っていた。B社の自社製品ではないC社製負荷分散装置の障害調査は後回しにした		
					原因はC社製負荷分散装置のsodプロセスのメモリ資源が時間とともに増加するという既知の不具合によるものであったことがわかった	
		A社があらかじめ用意していたクラウド外の「障害時バックアップシステム」に切り替わり、データ照会処理はできたので、データの更新を伴わないサービスのみを実施した				
						状況はD社には全く伝えられていなかった
			A社は各方面への説明対応に追われた			
		業務窓口は、登録系業務の処理の対応手順がわからなかったため　顧客の登録・変更申請に対応できなかった。				
	結果として、窓口に来た顧客に帰って頂く対応となった					
16時			障害箇所が判明したのは16：00であった			
				負荷分散装置はD社と共用しており、再起動等の対処によるD社サービスへの影響が不明のため、A社の了解のもと対処を業務終了後まで先送りすることとした		
20時			午後8時頃に、障害の原因を負荷分散装置と特定し、試みにA社の仮想LBを再起動したところ、障害が復旧した。			
午前1時半			負荷分散装置のハードウェア構成全体の再起動を行い、正常稼動の確認が完了した。			
その他				B社は、システム構成機器の修正情報の収集間隔を、3ヶ月に1回程度と非常に粗く設定していた		
			A社のシステムでは、本稼動以来、負荷分散装置は8か月以上連続運転状態であり、一般的なネットワーク機器と同様に再起動をしたことがなかった			
			基幹業務システムが利用できない場合の事務マニュアルはなく、システムが使えないと業務遂行はお手上げの状態であった			

図　参考１．２－１　障害状況表

参考１．３　障害事例をなぜなぜ分析で検討した例

参考１．１の障害事例をなぜなぜ分析で検討したものを図　参考１．３－１に示す。なお、図中の「対策」欄のうち、IPA が公開している教訓の作成に利用したものについては対応する教訓 ID を併記している他、参考１．４で紹介している教訓例に関係する項目を黄色の背景色で示している。

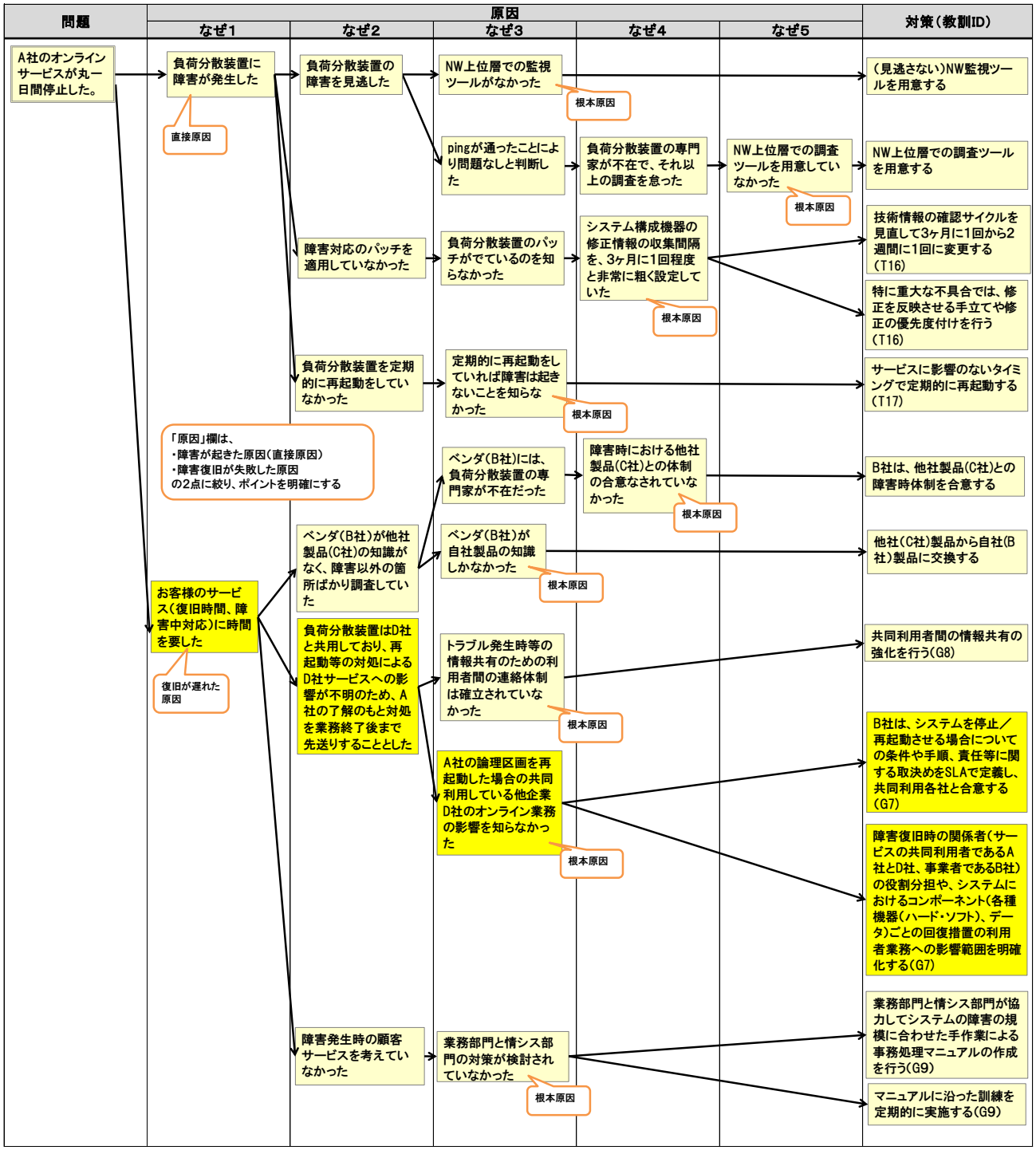


図　参考１．３－１　障害事例をなぜなぜ分析で検討した例

参考１．４　作成した教訓の例

[教訓 G7]
クラウド事業者と利用者が連携した統制がとれたトラブル対応体制を整備すべし

問題

A 社はオンラインによる情報登録及び情報照会の基幹業務システムを当初はオンプレミスで運用していたが、運用コストの削減を目的に複数企業間の共同利用を進める方針となり、B 社が提供するクラウドサービスに移行した。同時期に共同利用に移行するのは他に D 社があり、類似のビジネスを行っていた。B 社が提供するシステムは、業務システム用のサーバと負荷分散装置に分かれている。業務システムのサーバだけでなく、負荷分散装置も仮想化されており、その一つの論理区画を A 社は利用していた。(図　参考１．４－１システム概要)

ある日、オンライン開始時からこのシステムに障害が発生してまる１日業務が停止した。基幹オンラインシステムが端末から起動できず、すべての窓口でデータベースの更新を伴う処理の受付けができなかった（①)。

なお、A 社があらかじめ用意していたクラウド外の「障害時バックアップシステム」に切り替わり、データ照会処理はできたので、データの更新を伴わないサービスのみを実施した（②)。

B 社は障害箇所の特定に時間を要し、また A 社は各方面への説明対応に追われたこともあり、障害箇所が判明したのは１６：００であった。すでに業務終了時間が近づいていたためオンラインは終日停止、障害復旧作業はその後実施となった。

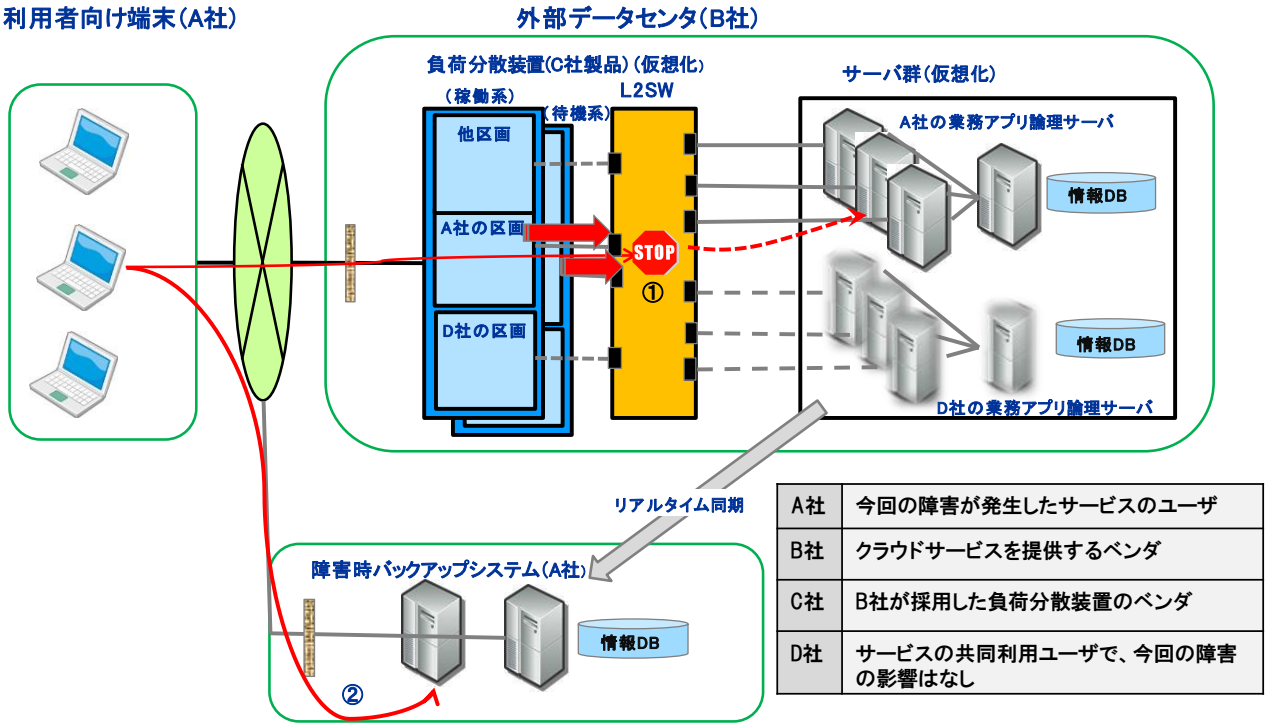


図　参考１．４－１システム概要

原因

直接の原因は、C 社製負荷分散装置の sod プロセスのメモリ資源が時間とともに増加するという既知の不具合であった。

単なる負荷分散装置の障害にも関わらず、その解決と業務の再開に多大の時間を要し丸一日間オンラインサービスが停止することとなった原因は、以下のとおりである。

- 通信路の疎通状況を確認する ping が通ったことから、B 社はシステムの運用に問題ないと誤認した。
- 初めのうちは AP サーバや専用線の問題と誤認し調査を行っていた。B 社の自社製品ではない C 社製負荷分散装置の障害調査は後回しにした。
- 負荷分散装置は D 社と共用しており、再起動等の対処による D 社サービスへの影響が不明のため、A 社の了解のもと対処を業務終了後まで先送りすることとした。

根本原因は以下のとおりである。

1．運用時のトラブル管理体制が決まっていなかった。
　障害調査を進め、一体となって協力して進めていく体制ができていなかった。B 社の SE は A 社に常駐していたが、トラブル管理体制が明確化されていないので、報告、連絡、相談がうまく回らなかった。
2．A 社は B 社と役割分担やサービスレベルが不明確な運用委託契約のままサービスを開始していた。
3．B 社に C 社製負荷分散装置の専門家が不在で、社外の製品のため障害情報の入手もしづらく、障害やパッチの情報をタイムリーに入手していなかった。

対策

再発防止策は以下のとおりである。

1．トラブル対応の体制の強化
　トラブル管理体制を明確化し障害発生時の報告、連絡、相談を行う。
　（考慮すべきこととして、ユーザは、対応をベンダ任せにせず積極的に働きかける。ベンダは、ユーザに対する状況報告を密に行う）
　トラブル発生時はユーザとベンダを TV 会議で結び、ユーザとベンダが密接に協力して対応するなどを検討する。
2．適切な契約でサービスのレベルの定義を行い、責任分界点を明確にする。
3．ベンダは関係する各サードパーティ業者とトラブル対応体制を確立し、障害時の連携を適確に行う

効果

クラウドサービスにおいても役割や責任が明確となり、障害発生時のエスカレーションや対応を迅速に行うことができる。

教訓

ユーザはクラウド型システムの障害発生に備えて、クラウド事業者と連携した統制がとれたトラブル対応体制の整備が必要である。特にユーザはベンダに対して、役割分担や契約などのやるべきことをはっきりと要求し、厳しく緊張感を持って対応すべきである。これによりシステムの信頼性が向上するだけでなく、両者がお互いに成長することができる。

参考資料）

クラウド適用のガイドラインについて

特定非営利活動法人 ASP・SaaS・IoT クラウドコンソーシアム（ASPIC）

ASPIC が取組んできたガイドライン・指針等

参考文献

（文献 1）小倉仁志「なぜなぜ分析 実践編」、日経 BP 社、2010/12/6

（文献 2）欠番

（文献 3）「日経 SYSTEMS」2013 年 5 月号『特集 トヨタ流 五つの技 なぜなぜ分析 応急処置で終わらせず原因を掘り下げ再発防ぐ』49 ページ 京都電子計算における「画面表示の不具合」に関するなぜなぜ分析の例

（文献 4）河野龍太郎「医療におけるヒューマンエラー［第 2 版］ なぜ間違える どう防ぐ」、医学書院、2014/3/1

（文献 5）石川雅彦「RCA 根本原因分析法 実践マニュアル 再発防止と医療安全教育への活用」、医学書院、2012/3/1、

（文献 6）「総合的インシデント分析によるサービスマネジメント領域の強化」、富士通ジャーナル 、2009/11

（文献 7）Nancy G. Leveson 著 松原友夫 監訳「セーフウェア ～安心・安全なシステムとソフトウェアを目指して～」、翔泳社、2009.

（文献 8）Embedded Technology 2013／組込み総合技術展基調講演
「ソフトウェア高信頼性への道程」講演資料（新誠一（電気通信大学情報理工学研究科）（2013））
https://www.ipa.go.jp/sec/old/users/seminar/seminar_et2013_20131121a-01.pdf

（文献 9）IPA「STAMP 手法に関する調査報告書」IPA
https://www.ipa.go.jp/sec/reports/20150918.html

（文献 10）Nancy G. Leveson, “Engineering a Safer World: Systems Thinking Applied to Safety (Engineering Systems),” 2012.
https://mitpress.mit.edu/books/engineering-safer-world

（文献 11）IPA「はじめての STAMP/STPA ～システム思考に基づく新しい安全性解析手法～」IPA
https://www.ipa.go.jp/sec/reports/20160428.html

（文献 12）Nancy G. Leveson, CAST-Tutorial, Nancy-Leveson,Nancy-Leveson_CAST-Tutorial-2017.pdf

（文献 13）Paul S. Nelson, A STAMP ANALYSIS OF THE LEX COMAIR 5191 ACCIDENT

（文献 14）Nancy G. Leveson, CAST Analysis of the Shell Moerdijk Accident

（文献 15）IPA「情報処理システム高信頼化教訓集 IT サービス編」
https://www.ipa.go.jp/ikc/publish/tn19-001.html

情報処理システム高信頼化 教訓集

IT サービス編　別冊Ⅱ：障害分析手法

2019 年 2 月 28 日 PDF 版発行
2020 年 3 月 16 日 改訂

監修者 独立行政法人情報処理推進機構（IPA）
社会基盤センター
発行人 片岡　晃
発行所 独立行政法人情報処理推進機構（IPA）
〒113-6591
東京都文京区本駒込二丁目 28 番 8 号
文京グリーンコート センターオフィス
https://www.ipa.go.jp/ikc/

独立行政法人 情報処理推進機構
社会基盤センター