

**第2部：
グループ演習
～問題の発見と教訓を作成するプロセスを体験～**

2019年3月5日

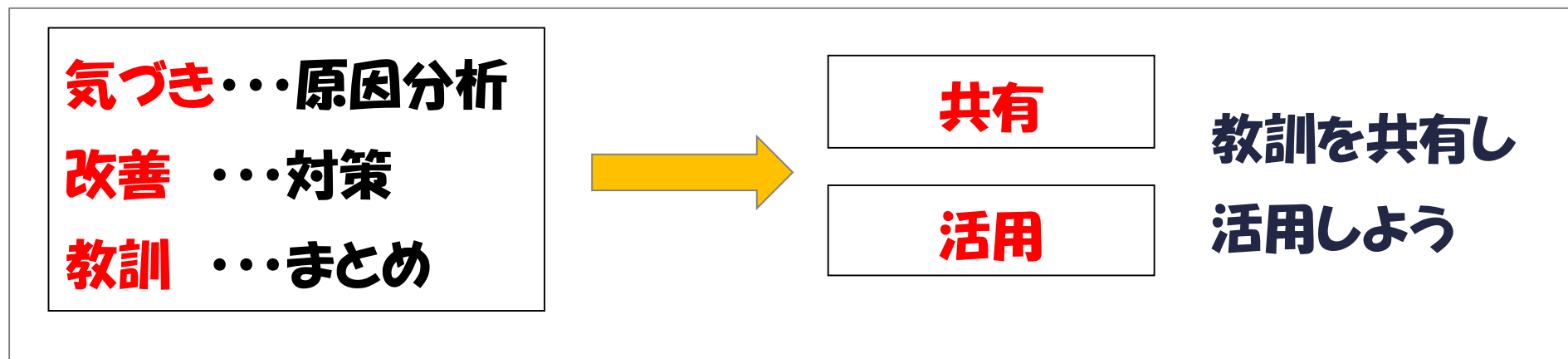
**独立行政法人情報処理推進機構(IPA)
社会基盤センター(IKC)
産業プラットフォーム部 目黒 達生**

～目次～

- **演習の目的・進め方**
チーム内自己紹介、チーム名を決める
13:30-13:40 (10分)
- **1. 教訓を活用する**
(教訓集を原因究明に活用する)
13:40-14:30 (50分)
14:30-17:10 (160分)
- **2. 教訓を作成するプロセスの事例演習**
(実際の事例から問題を分析して教訓化するプロセス
を体験する)
17:10-17:30 (20分)
- **3. 自分で教訓を作成する**
(自分の体験をもとに教訓を作成)

障害事例を教訓化し共有化することにより
システム障害を減らそう！

システム障害を減らす5つのキーワード(5K)

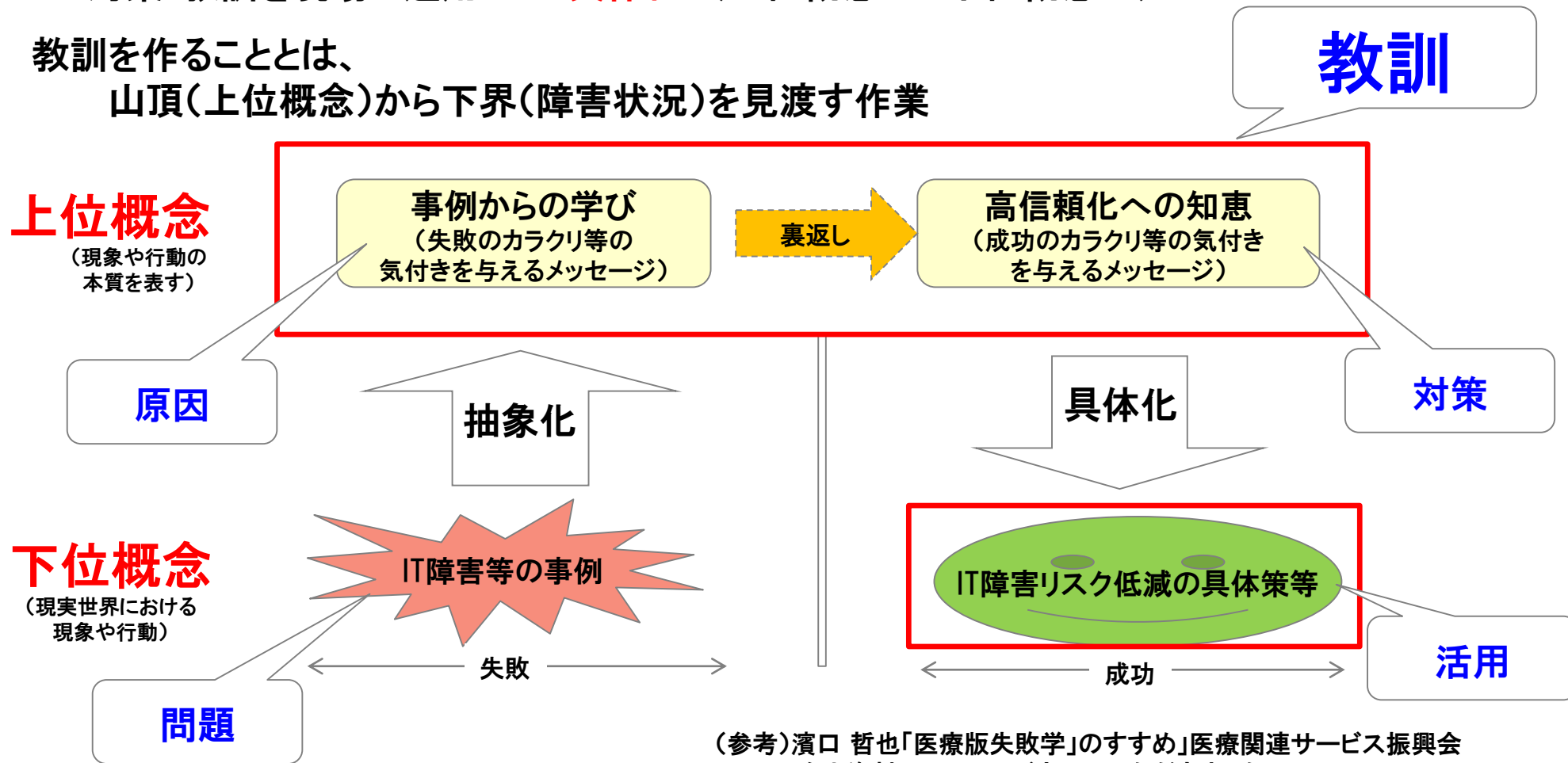


分析方法、対策方法、教訓作成を体験学習

演習の目的 教訓作成・共有活動の流れ

1. 問題(個々の事象)の原因を追求 → **抽象化**(下位概念から上位概念へ)
2. 原因から対策を導く → **裏返し** → **教訓化**(上位概念での普遍化)
3. 対策・教訓を現場で適用 → **具体化**(上位概念から下位概念へ)

教訓を作ることとは、
山頂(上位概念)から下界(障害状況)を見渡す作業



(参考)濱口 哲也「医療版失敗学」のすすめ」医療関連サービス振興会
を当資料用にアレンジさせていただきました。

1. 相手の意見は否定しない

→ 何故そう考えたかを理解しよう

2. どれも正解

→ 世の中、想定外が起きている

常識にとらわれない

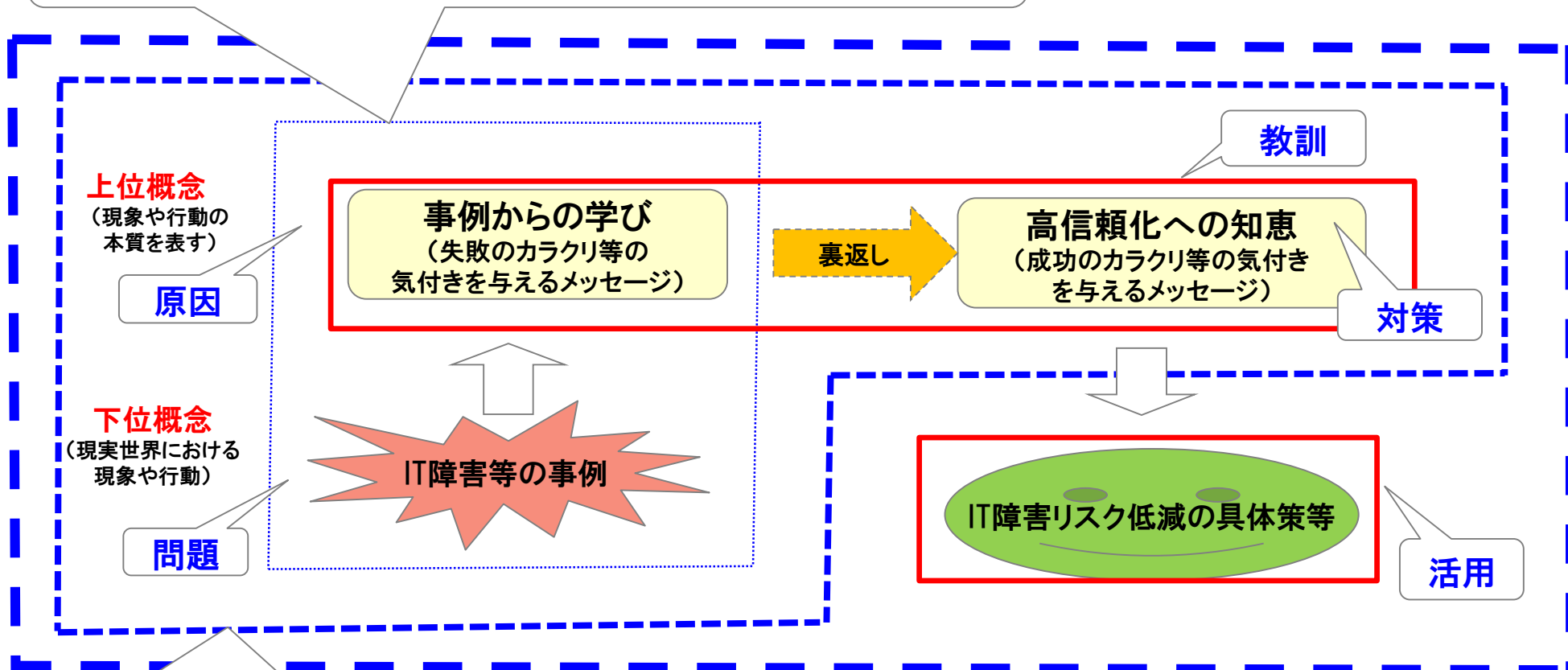
3. IT技術以外の知識、経験も重要

→ ハンテクニカルスキル

演習の進め方 3つの演習

1. 教訓を活用する

問題点、原因から、活用できる教訓を探し、対策に役立てよう



2. 教訓を作成するプロセスの事例演習

問題点 → 原因 → 対策 → 教訓
とひと通りのプロセスを体験しよう

3. 自分で教訓を作成する

自身の経験、本日の事例をもとに、
教訓を作成してみよう

1. 教訓の活用を考える

1-1. 事例説明

説明13:40-13:45（5分を想定）

1-2. 活用方法

グループ討議13:45-14:15（30分を想定）

演習事例

発表14:15-14:30（15分を想定）

1)このシステムの問題と思われる点をできるだけ取り上げ、その中で根本原因と思われるものは何かを考えてみよう。

→ この事例では、具体的なシステム障害の内容がほとんど記述されていません。運用上の問題をいろいろ考えてみましょう。想像も大歓迎です。

2)教訓集のどの教訓が活用できるか考えてみよう。

→ 教訓集の中には、教訓タイトルから「似ているかも・・・」、「この教訓からヒントが得られるかも・・・」とヒントになる事例がたくさんあります。どんどん、想像で見つけてみてください。その中に多くの「気づき」が潜んでいます。

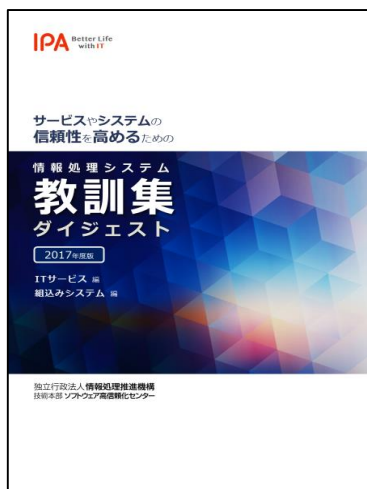
1-2. 活用方法

①考えられるだけ、問題点を探し出そう

・事例を読んで、「ここが問題点だ。」「ここがおかしいのではないか。」と抜き出してみましょう。

②「情報処理システム高信頼化教訓集ダイジェスト」2017年度版から、抜き出した問題点に役立ちそうな「教訓」を探してみてください。

想像を働かしてください
記述されていない問題点も
重要です



ITサービス編					ガバナンス・マネジメントに関する教訓一覧 (1/2)																
No.	教訓集の頁次(分類)	教訓概要	問題	発生原因	根本原因	No.	対策	キーワード	3.15	3.15	3.15	3.15	3.15	3.15	3.15	3.15	3.15	3.15	3.15	3.15	
G1	事業部門やシステム部門の担当者に対する教訓	システム開発や運用に関する業務は、システム担当者だけでなく、事業部門やシステム部門の担当者も理解し、責任を持って取り組む必要がある。	急務に迫ったシステム開発により、システムとデータの連携がうまくいかなかった。	・ビジネス側の要件の理解が不十分であった。 ・要件を厳格的に文書で確認していない。	上流の要件定義段階でのコミュニケーション・キックオフ	G1	システム開発におけるビジネスサイドの役割と責任の明確化	アプリケーション・オーナー制、要件定義、導入テスト	△												
G2	要件定義の要件定義書に關する教訓	要件定義はシステム構築に直結する重要な要素である。	要件定義に基本となる要件が漏れがあった。	要件定義漏れ	要件定義がベンダ任せだった。要件定義書の開発プロセスでなかった。	G2	要件定義書と導入テストは別注者の責任とする。上流工程、要件定義、開発責任	要件定義、要件定義書、要件定義書	○												
G3	上流工程での要件定義の精度に関する教訓	要件定義は上流工程(企画、要件定義)から開発部門と連携して進めるべき。	オペレータの操作ミス、オンライン発生業務のデータ入力ミスによる発生	要件定義段階でのオペレーション要件確認も、入力ミスの防止の設計がなかった。	運用者が企画、要件定義に参画していない。	G3	運用者の役割を把握し、プロセスマネジメントの観点からレビュー	運用設計、企画、要件定義、上流工程	○	△											
G4	障害発生時の連絡体制に関する教訓	障害発生時に迅速に連絡し、対応する必要がある。	システムサービスの発生	メモリコントローラの障害	運用担当者が発生した異常を、発生した時点で認識しなかった。	G4	オペレーションマニュアル改訂、上位管理者の検閲・承認、フェールシフトの仕組み導入、検閲、訓練実施	システム運用、異常発生時、検閲													
G5	共有利用システムの運用に関する教訓	サービスの拡大に伴って運用の効率化を図る必要がある。	共有システムによるシステムダウン	共有システムを起動とする際のソフトウェアの互換性	共有利用システムにおける各社のソフトウェアの互換性がなかった。	G5	利用会社による影響評価会の実施、共有システムを合わせた検証実施、ベンダ向けプランニング	共有利用、異常発生時、運用設計、ベンダ向けプランニング													
G6	作業ミス、ルール逸脱の発生に関する教訓	作業ミスやルール逸脱は、個人の経験ではなく、組織の課題である。	グループウェアの全ユーザへの更新	一部ユーザを削除する際の、運用者の作業ミス	慣習や経験で、不備のある運用が行われていた。	G6	作業ミス、ルール逸脱、グループウェア、システム管理、運用ルール	作業ミス、ルール逸脱、グループウェア、システム管理、運用ルール													
G7	クラウドサービス利用時の運用に関する教訓	クラウド事業者と利用者の連携が重要である。	クラウド事業者のシステム障害	運用担当者によるシステム障害	運用担当者が発生した異常を、発生した時点で認識しなかった。	G7	障害対応体制の強化、契約におけるサービスレベル定義、クラウド事業者とパートナー事業者間の連携強化	故障対応、共同利用、トラブル管理													
G8	共有利用システムの利用に関する教訓	共有利用システムでは、効果的な運用を確保する必要がある。	(同上)	(同上)	ベンダによる共有利用システムの運用に関する課題の発生	G8	障害発生時の停止/再開手順と共同運用の確保、システム停止/再稼働の条件や責任について話し合う	故障対応、共同利用、情報共有													
G9	企業向け標準マニュアルに関する教訓	システム利用に関する標準マニュアルを作成し、定期的な更新を行う必要がある。	(同上)	(同上)	過去にシステム障害が発生したことがない。システムが利用できない。従前の運用マニュアルがなかった。	G9	システム利用に関する標準マニュアルの作成	故障対応、共同利用、異常発生時													
G10	システム動作の異常発生に関する教訓	システム動作の異常発生は、システム運用者によるものである。	ユーザがシステムに接続できない状態が発生した。	ユーザがシステムに接続できない状態が発生した。	運用担当者によるシステム障害	G10	代替手段の確保と運用、運用設計の改善と検証、運用設計の改善と検証	コールセンター、デスクトップ、運用設計、運用設計													

2. 教訓を作成するプロセスの事例演習

教訓化アプローチの手順（演習の進め方）

2-1. 事例説明

2-2. 障害情報を整理する

2-3. 障害原因を分析する

2-4. なぜなぜ分析事例

2-5. なぜなぜ分析・留意点

2-6. 障害の対策手法を検討する

2-7. 教訓化する

説明 14:30-14:45
(15分)

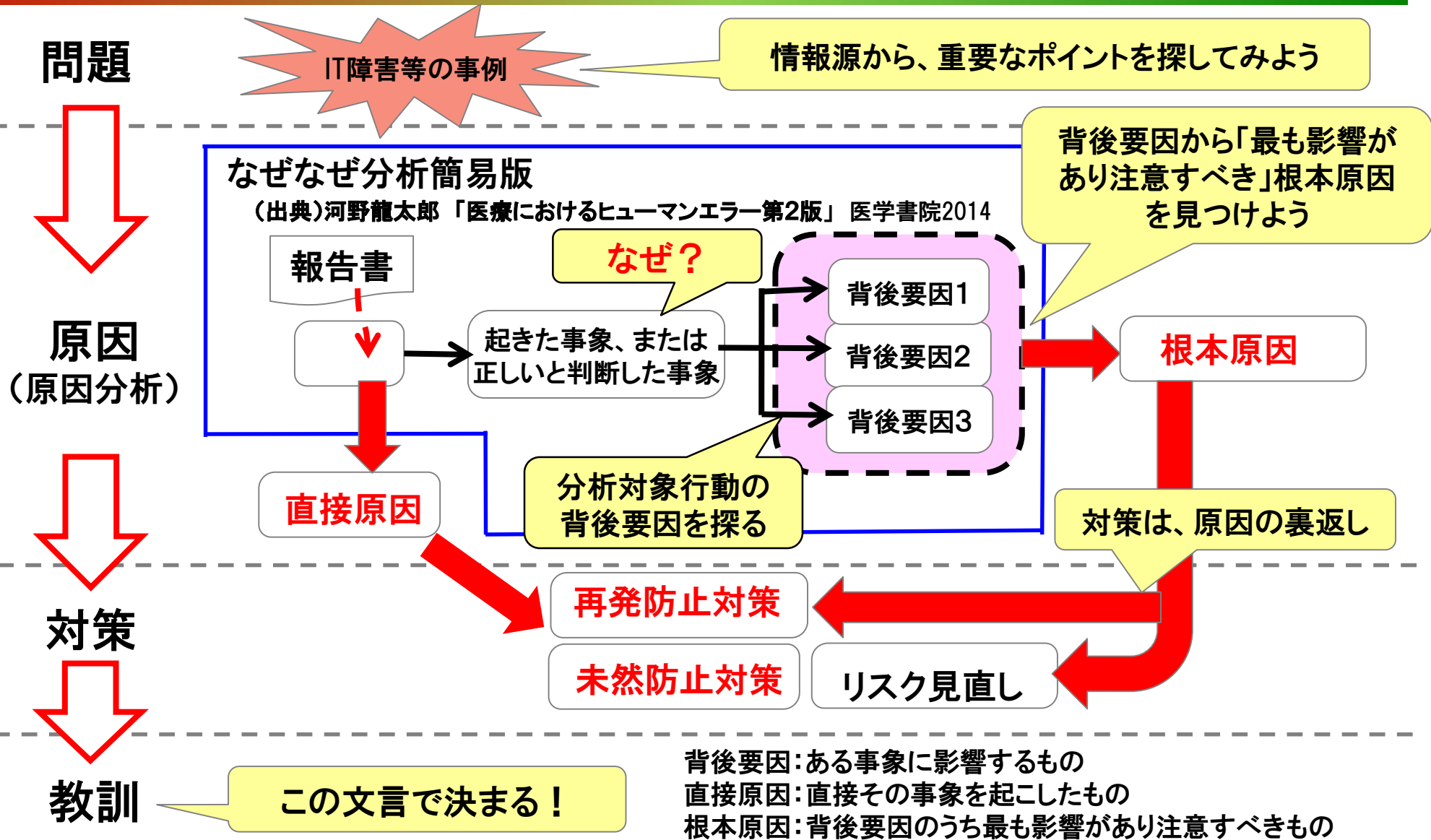
問題把握・原因分析 14:45-15:50
(説明15分、グループ討議50分)

対策検討・教訓化 15:50-17:10
(説明10分、グループ討議50分、
発表20分)

参考資料)

【資料2-2】 システム障害事例

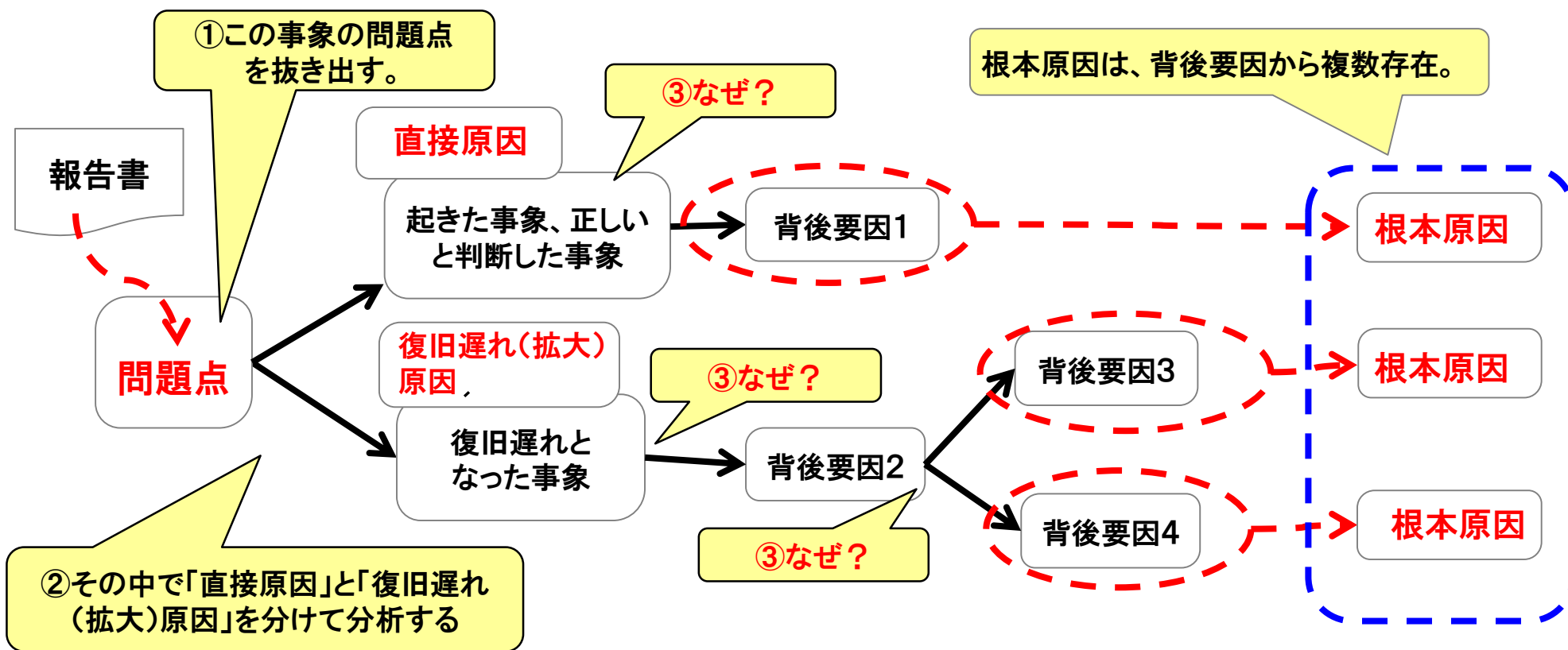
2. 教訓を作成するプロセスの事例演習 全体



2-2. 障害情報を整理する

・資料2-2の障害情報から、問題点を抜き出してください

1. 多くの事実を集め、整理
2. その中から、重要な問題点(直接原因、復旧遅れ(拡大)原因)を抽出→ **あらゆる想定(想像力)**



(参考)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

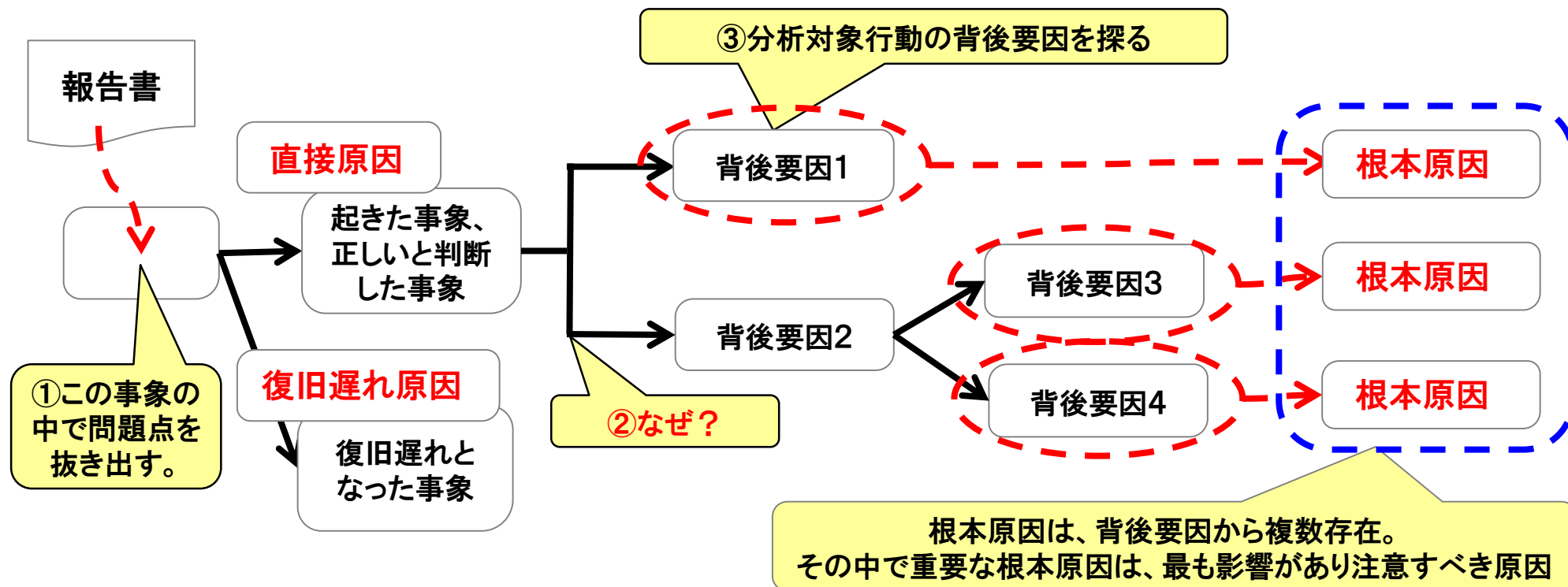
2-3. 障害原因を分析する

・このシステム障害の根本原因を探す

1. 直接原因 → 問題の発生から

2. 根本原因 → 本質は何か(上位概念)

・報道、報告書からの情報: あらゆる想定(想像力)



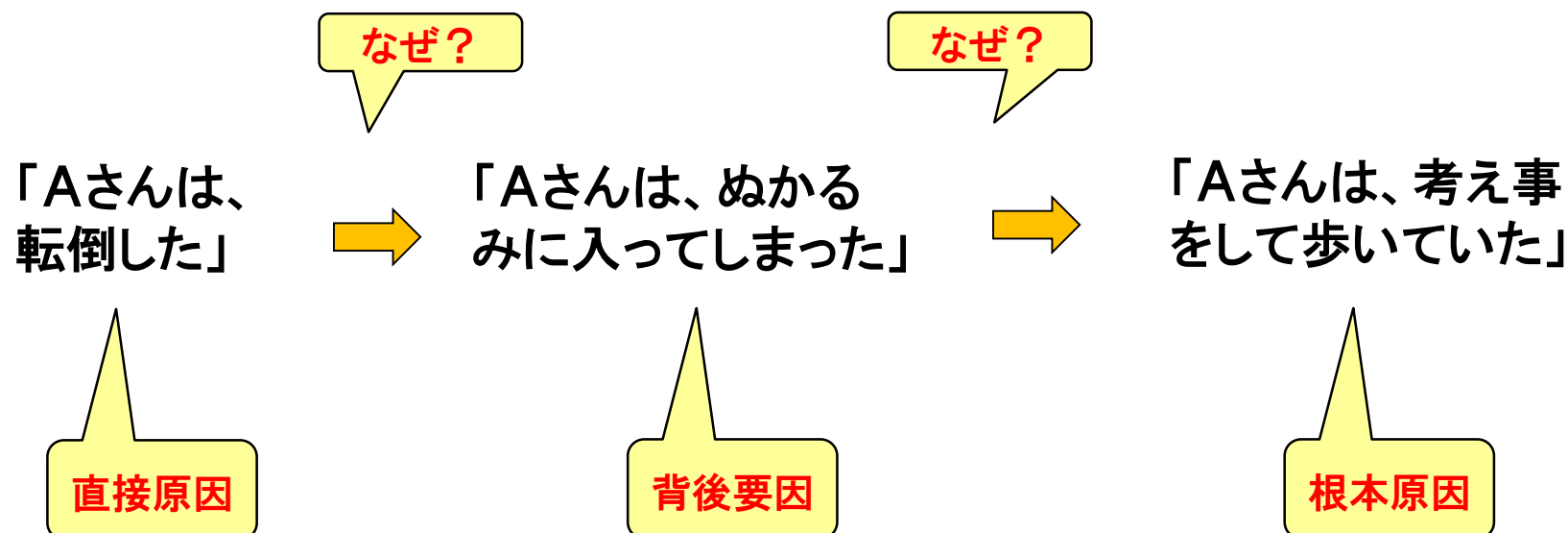
(参考)河野龍太郎 「医療におけるヒューマンエラー第2版」医学書院2014

2-3. 障害原因を分析する

◆なぜなぜ分析とは

ある問題とその問題に対する対策に関して、その問題を引き起こした要因(『なぜ』)を提示し、さらにその要因を引き起こした要因(『なぜ』)を提示することを繰り返すことにより、その問題への対策の効果を検証する手段である。トヨタ生産方式を構成する代表的な手段の一つ

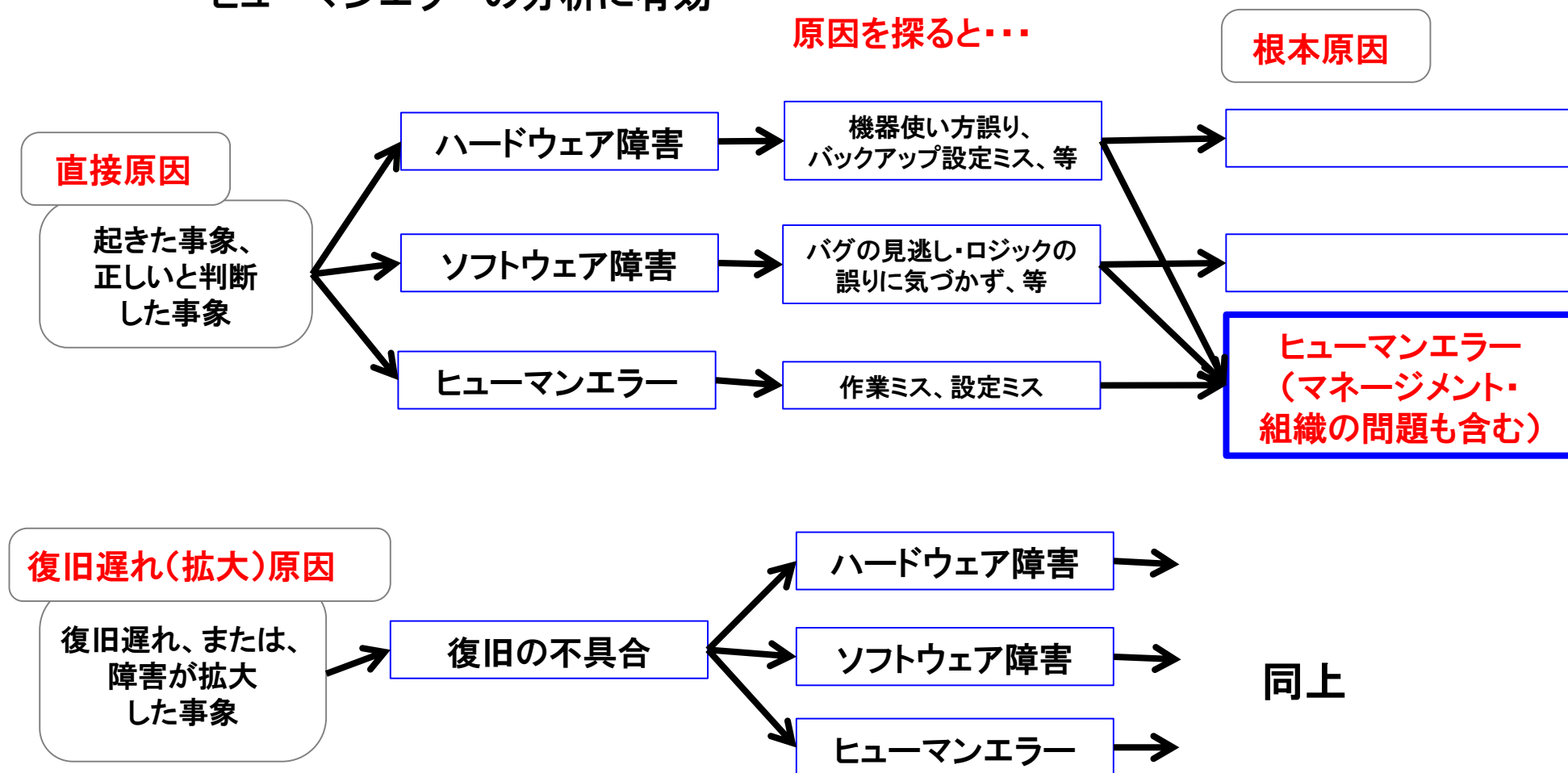
(出典)ウィキペディア



2-3. 障害原因を分析する

◆なぜなぜ分析 を使う理由

→ ヒューマンエラーの分析に有効



2-3. 障害原因を分析する

◆なぜなぜ分析 現場とセミナーの違い

現場では・・・

当事者を交えて

1. 多くの事実を集め、整理
2. その中から、重要な問題点を抽出

事実を探っていく作業

当事者が真実を語って
くれない場合も・・・
想像力も重要

セミナーでは・・・

報告書から(当事者はいない)

報告書の中から、重要な問題点を抽出

過去の経験から**想像力**を働かす作業

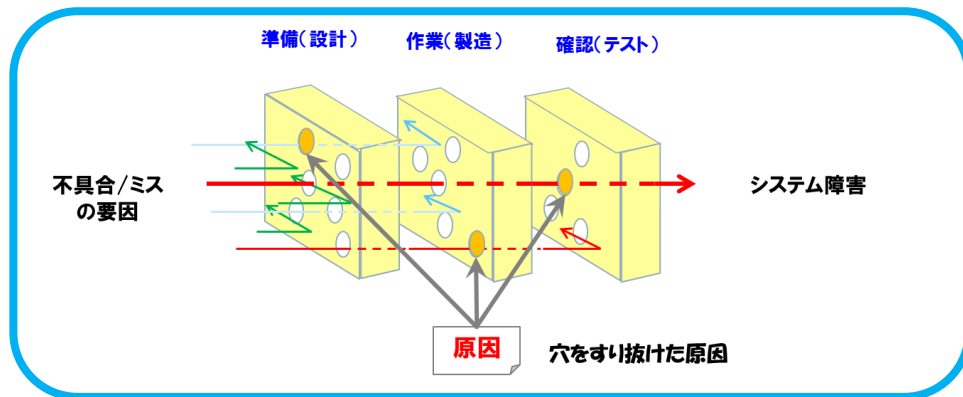
セミナーの演習は、現場でも役に立ちます！

2-3. 障害原因を分析する

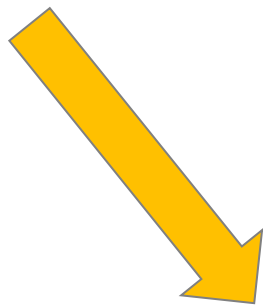
◆3つの観点による分析



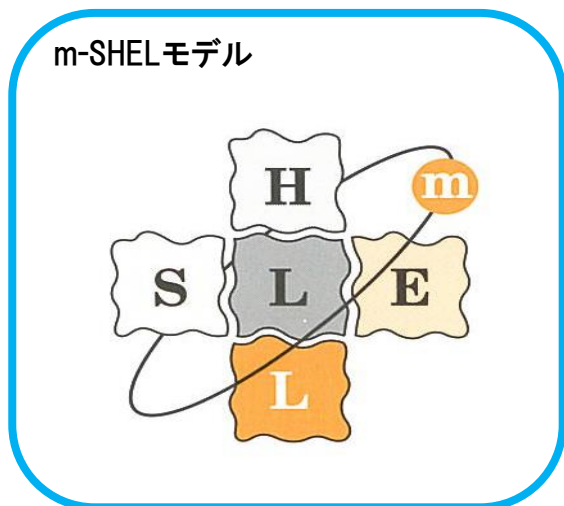
(1) IT運用/開発プロセスの観点



(2) 関係性の観点



(3) 人間特性の観点



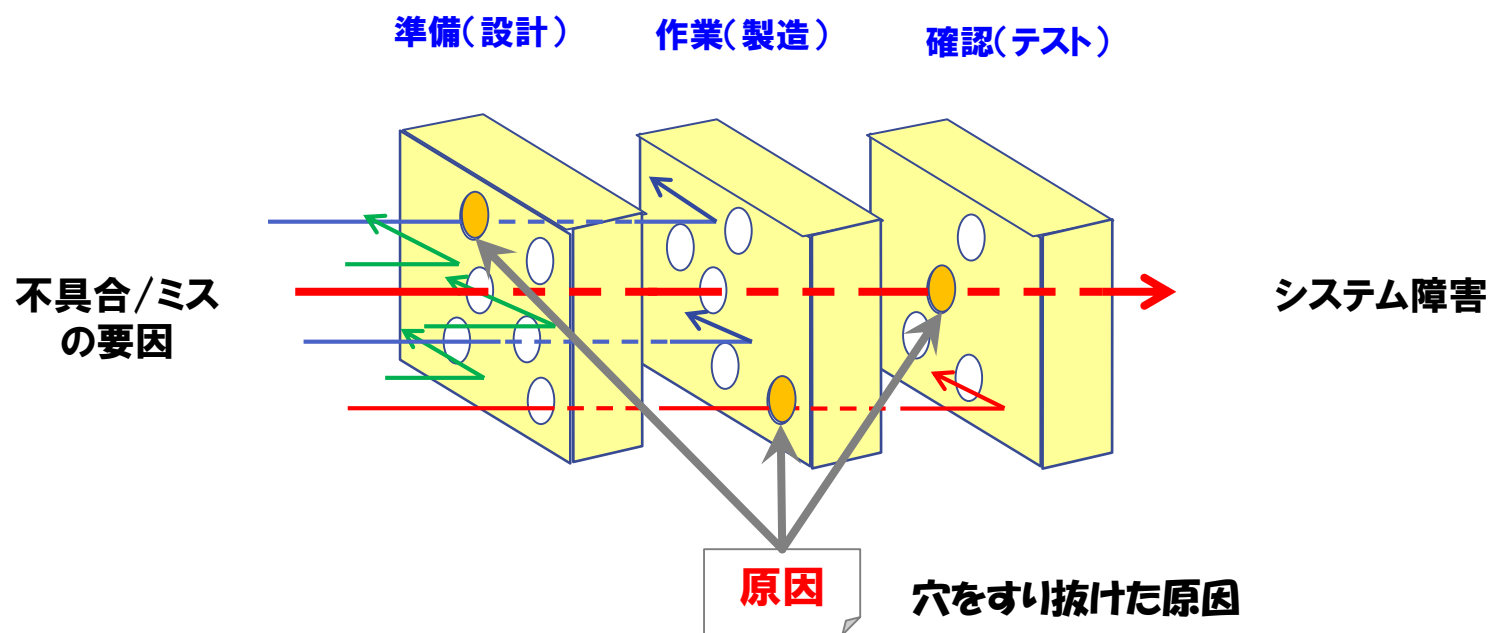
生理学的特性	認知的特性	社会心理学的特性
<ul style="list-style-type: none"> ・生理現象 ・睡眠不足 ・緊張 ・ストレス 	<ul style="list-style-type: none"> ・思い込み ・誤認 ・忘却 ・学習、経験 ・正常化の偏見 ・こじつけ解釈 ・緊急時 ・注意ちがい 	<ul style="list-style-type: none"> ・権威への服従 ・集団に合わせる心理 ・社会的手抜き ・集団浅慮 ・リスク・シフト現象

(参考)河野龍太郎 m-SHELモデル、ヒューマンエラー

2-3. 障害原因を分析する

(1) IT運用/開発プロセスの観点

誤りが起きた原因(プロセス)と見逃した原因(前後のプロセス)を分析



- ①どのプロセスで誤りが起きたのか、を考える。
- ②誤りを起こしたプロセス以外のプロセスで、なぜその誤りを見逃してしまったのかを考える。

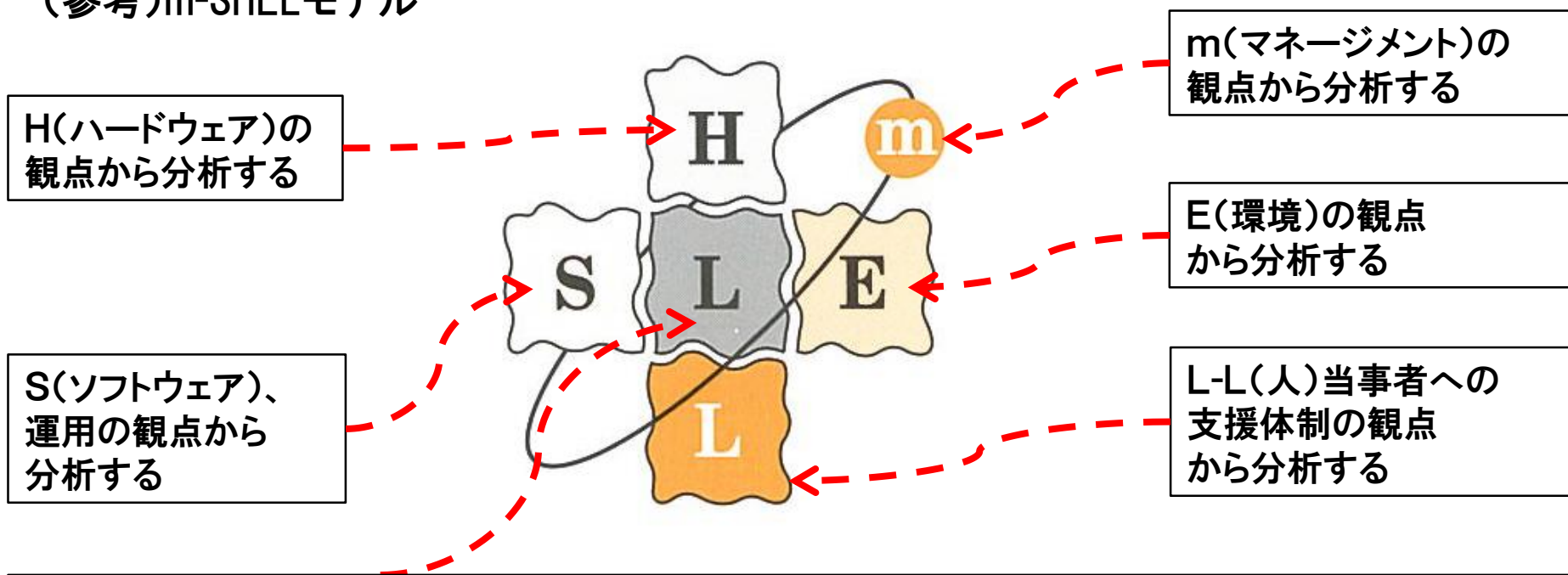
2-3. 障害原因を分析する

(2) 関係性の観点

m-SHELモデルをベースに要因の洗い出し

→ 原因調査の観点を理解すると分析がスムーズ！

(参考)m-SHELモデル



L(当事者): 通常はエラーを起こした人。

ITの場合は、情シス部門、プロジェクト、運用部門などを設定すると分析がやり易いことがある

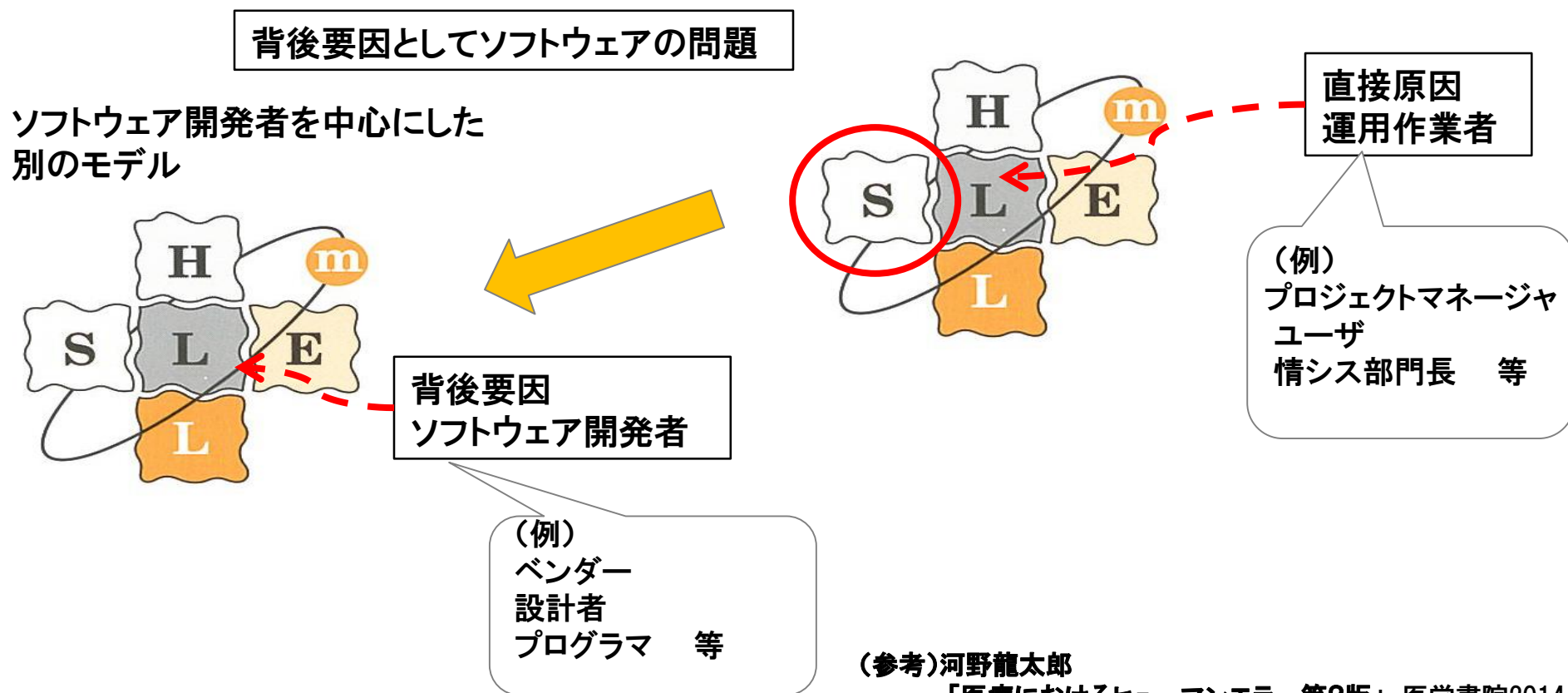
(参考)河野龍太郎

「医療におけるヒューマンエラー第2版」医学書院2014 (説明文を追加)

2-3. 障害原因を分析する

◆Lが、複数存在する時のモデル

- ・背後要因を分析するにつれて、直接原因のLとは、異なるLが登場する。
- ・L毎に、m-SHELモデルが作られる。



2-3. 障害原因を分析する

(3) 人間特性の観点

3種類の人間特性

生理学的特性

- ・生理現象
- ・睡眠不足
- ・緊張
- ・ストレス

認知的特性

- ・思い込み
- ・誤認
- ・忘却
- ・学習、経験
- ・正常化の偏見
- ・こじつけ解釈
- ・緊急時
- ・注意ちがい

社会心理学的特性

- ・権威への服従
- ・集団に合わせる心理
- ・社会的手抜き
- ・集団浅慮
- ・リスク・シフト現象

- ・正常化の偏見 ... 保守的で異常を認めない。明確な証拠がないと動かない、「兆候があったがたいしたことない」、非常ベルが鳴ってもすぐ逃げ出さない、周りの人が逃げるまで様子を見る。危険がある事を知りながら、地震が来てもほとんどの人が逃げない。
- ・こじつけ解釈 ... 情報の中から都合の悪い情報を都合の良いようにこじつける。患者取り違い。「もっと髪が長かった。でも散髪したのだろう」
- ・緊急時 ... 異常事態では、簡単なことでも思い出せない
- ・注意ちがい ... ①あるものに集中すると他の注意はおろそかになる
②関心があるものには注意がいくが、ないとおろそかになる
③注意は同じ水準で持続させることができない

- ・権威への服従 ... 思っても言えない。権威を持った人に言えない。
ベンダーは、ユーザに言えない。副操縦士は、機長に言えない
- ・集団に合わせる心理 ... みんなが言うからいいや。安易な追従。作業が増えるのが嫌な時、安易な追従
- ・社会的手抜き ... 誰かがやるだろう。1人単独作業量は、集団では低下する。1人月×2人 < 1.8人月
- ・集団浅慮 ... 連帯感が強い集団は、他の意見を聞かなくなる。結束を乱すものを排除。デモ隊？
- ・リスク・シフト現象 ... 赤信号みんなで渡れば怖くない。集団の決定は個人の決定よりも危険。
誰も責任を取らなくなるので。リスクな選択肢を選んでしまう。

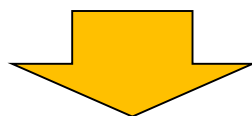
(参考)河野龍太郎「医療におけるヒューマンエラー第2版」医学書院2014

2-3. 障害原因を分析する

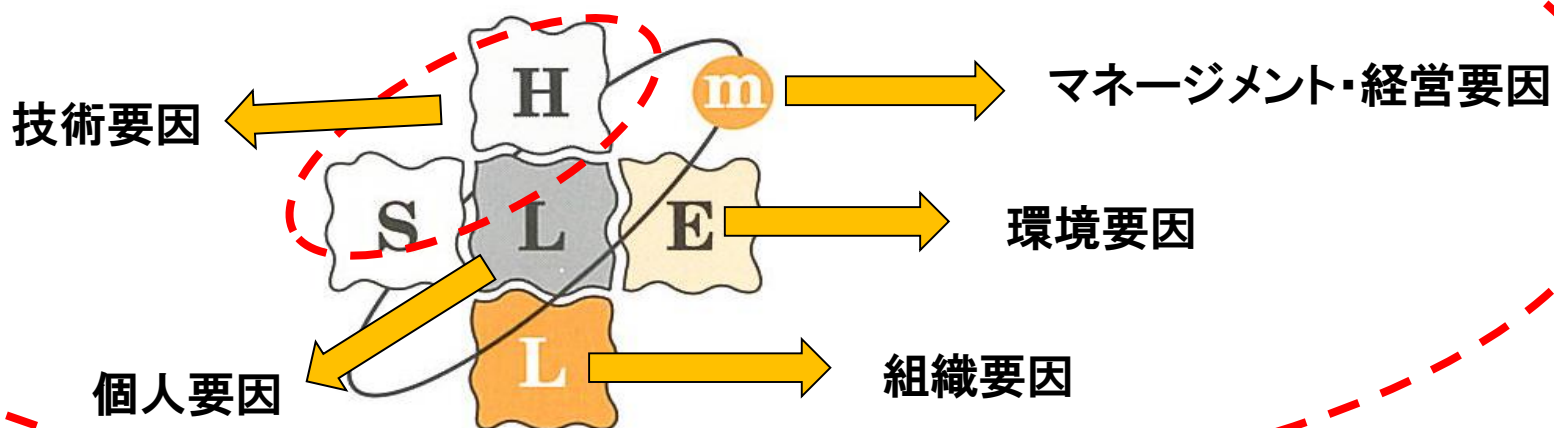
(4) 分析を行う上でのポイント

①なぜなぜ分析のゴールを定める

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に責任を持っていかず...



仕組みやシステム、組織の問題として考える



2-3. 障害原因を分析する

② 関心のある項目の洗い出し → 代表パターンを理解すると分析がスムーズ！

i .事象発生(障害発生)時の原因を、以下の2点から考える

- ・エラー発生はなぜ？
- ・エラー拡大阻止の失敗はなぜ？

ii .エラーを起こした当事者は、「自分は正しい」と判断。この「正しいと判断した」ことを「なぜ？」と考える(調べる)事で、その判断根拠、背後要因がでてきやすい。

(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

iii .IT技術者の観点でなく、エンドユーザの観点でも考えてみる

- ・システム障害の原因
- ・エンドユーザが困ったことは何？

2-3. 障害原因を分析する

③**背後要因の洗い出し方法** → パターンを理解すると分析がスムーズ！

i. 判断根拠(正しいと判断した理由)を考える情報の問題点

- ・誤った判断をうながす指示や情報はなかったか
→ ○○さんが「……」と指示を伝えた。…
- ・正しい判断に必要な手順や情報の欠如はなかったのか
→ 「……」との情報が伝わっていなかった

ii. やるべき行動をしなかった理由を考える。

- ・やらなくても良いと判断？ ・忘れた？ ・物理的にできなかった？

iii. 見逃した理由を考える。

- ・知覚しなかった？
→ 見えなかった、気づかなかった…
- ・認知しなかった？
→ 知覚したが、認知しなかった。見えたが、気に留めなかった。

(参考)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。

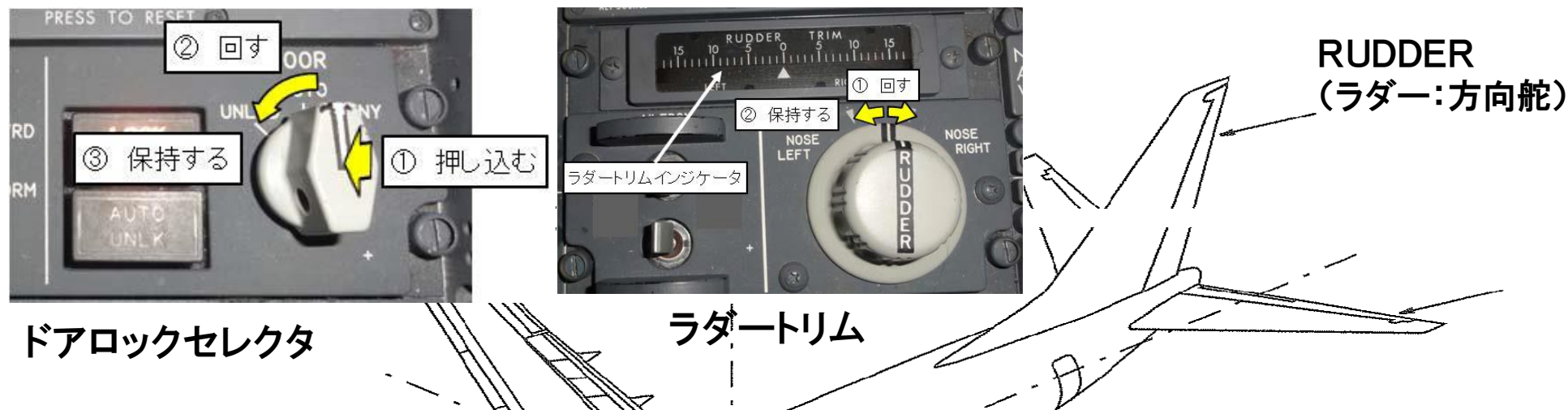
2-4. なぜなぜ分析事例

事例

ボーイング式737-700飛行機は、那覇空港から東京国際空港へ向けて飛行中、機体が異常姿勢の状態になり、急降下した。

原因

副操縦士が、機長を入室させることを目的に、操縦室の扉を解錠しようとして、**ラダートリムコントロールを誤操作**してしまった。



(出典) 国土交通省運輸安全委員会「航空重大インシデント調査の経過報告について」平成24年8月31日
<http://www.mlit.go.jp/jtsb/aircraft/rep-inci/keika120831-JA16AN.pdf>

2-4. なぜなぜ分析事例

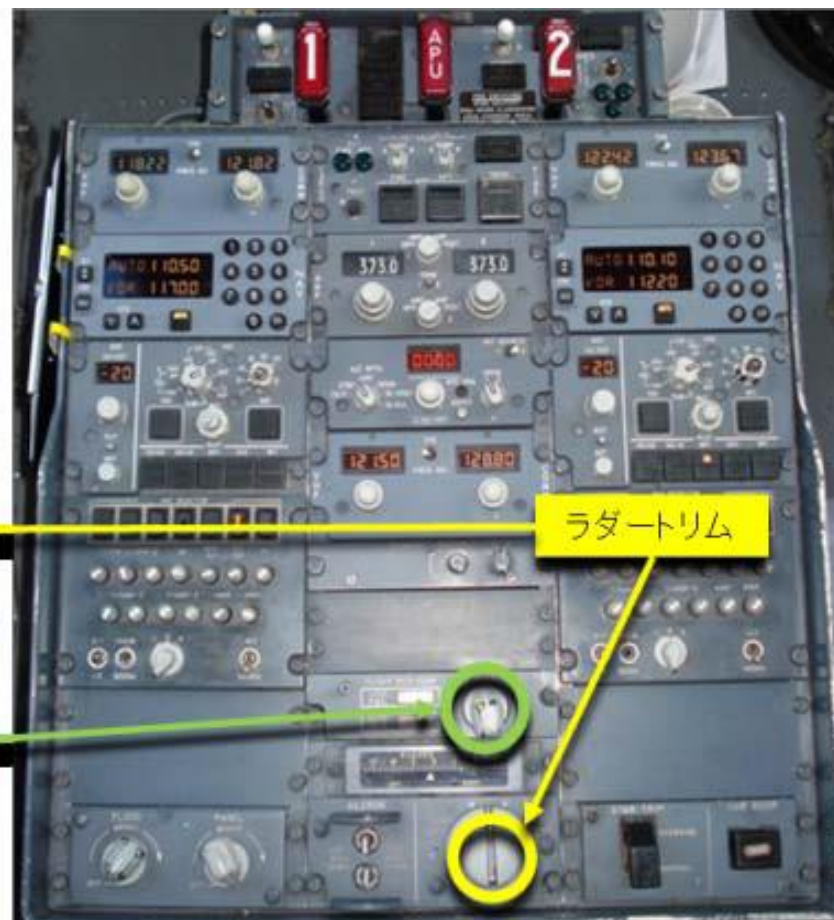
(1) インシデントの整理

- ・那覇空港から東京国際空港に向かって、串本の東約69nm、高度41,000ftを飛行
- ・**ひとり**で操縦していた副操縦士(機長は離室中)
- ・機長が入室を要求
- ・副操縦士が操縦室のドアを開けようとして、**誤って**航空機の姿勢をコントロールするラダートリム・スイッチを操作
- ・副操縦士の**回復操作が不適切**
- ・機体が異常な姿勢になり**急降下**

2-4. なぜなぜ分析事例



ボーイング式737-700型
副操縦士が**今回**操縦



ボーイング式737-500型
副操縦士が**今まで**操縦

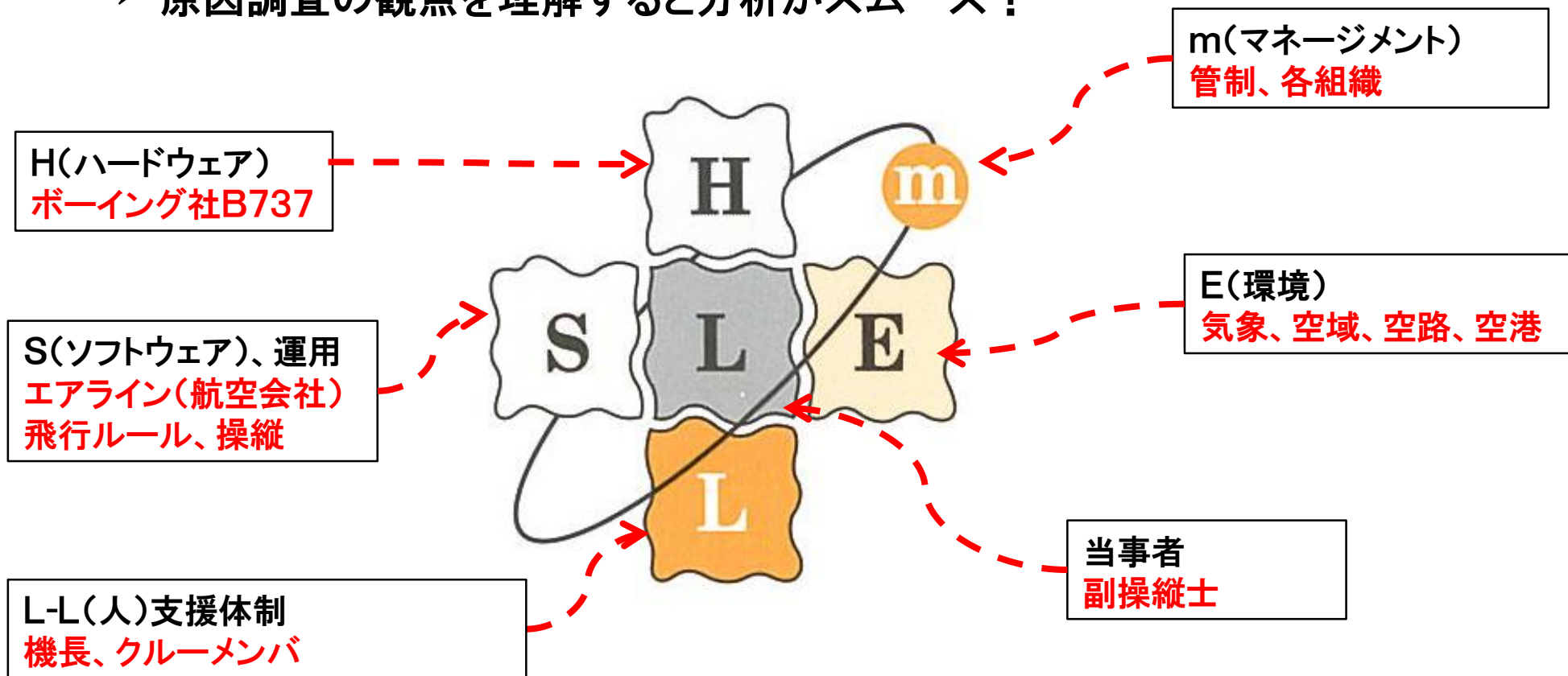
(出典) 国土交通省運輸安全委員会「航空重大インシデント調査の経過報告について」平成24年8月31日
<http://www.mlit.go.jp/jtsb/aircraft/rep-inci/keika120831-JA16AN.pdf>

2-4. なぜなぜ分析事例

(2)なぜなぜ分析

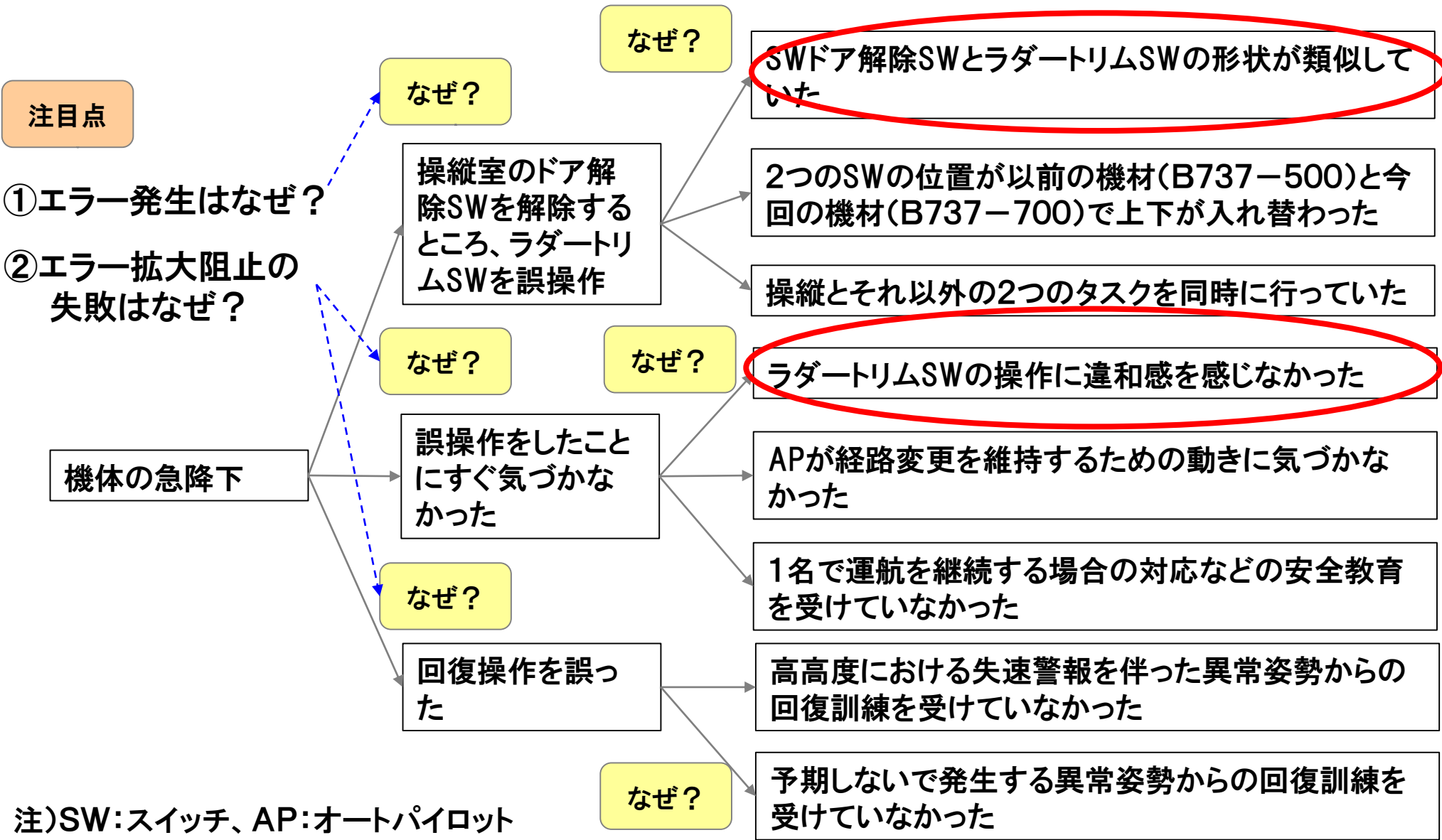
m-SHELモデルをベースに要因の洗い出し

→ 原因調査の観点を理解すると分析がスムーズ！



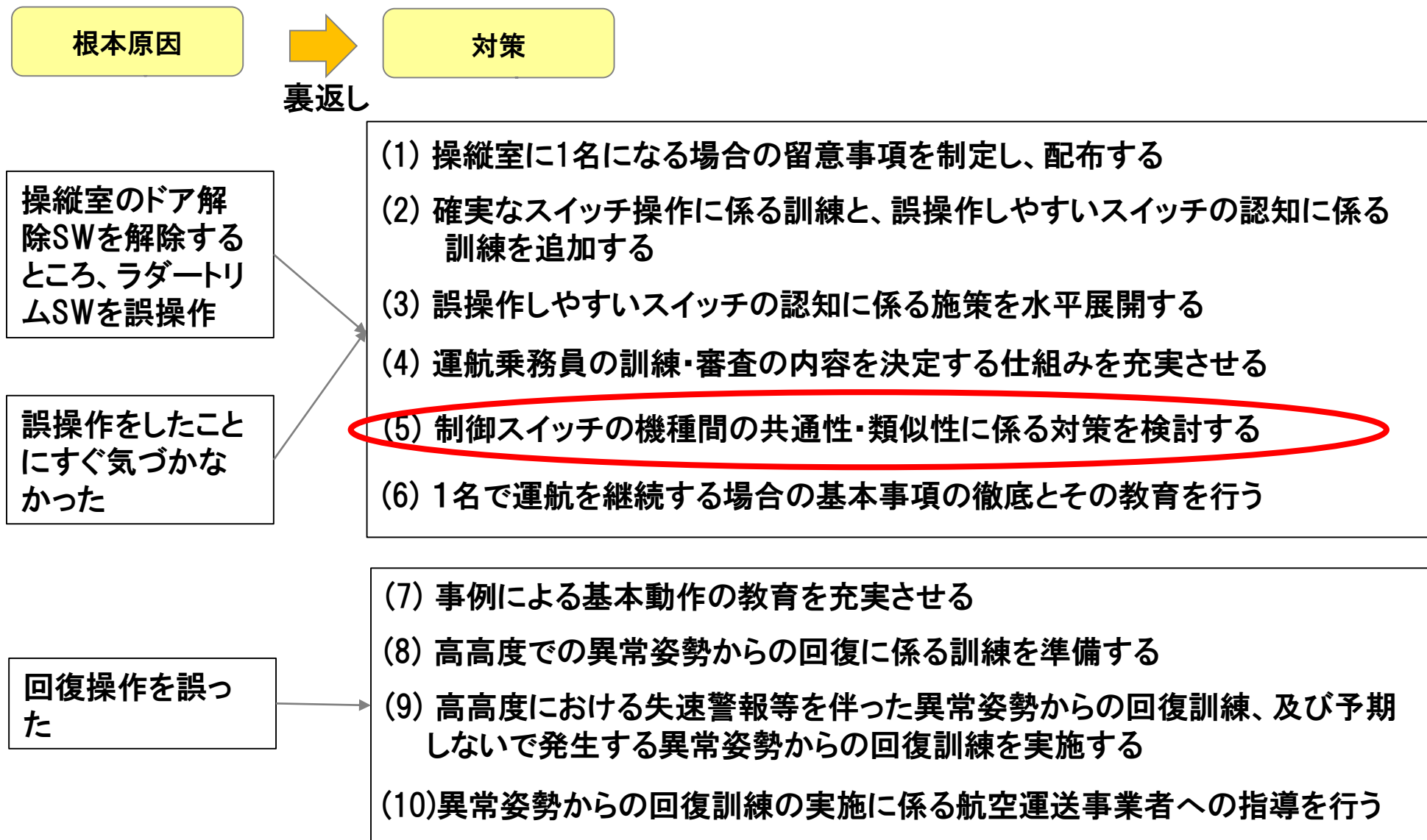
(参考)河野龍太郎 「医療におけるヒューマンエラー第2版」医学書院2014

2-4. なぜなぜ分析事例



注)SW:スイッチ、AP:オートパイロット

2-4. なぜなぜ分析事例



(3) その後のボーイング

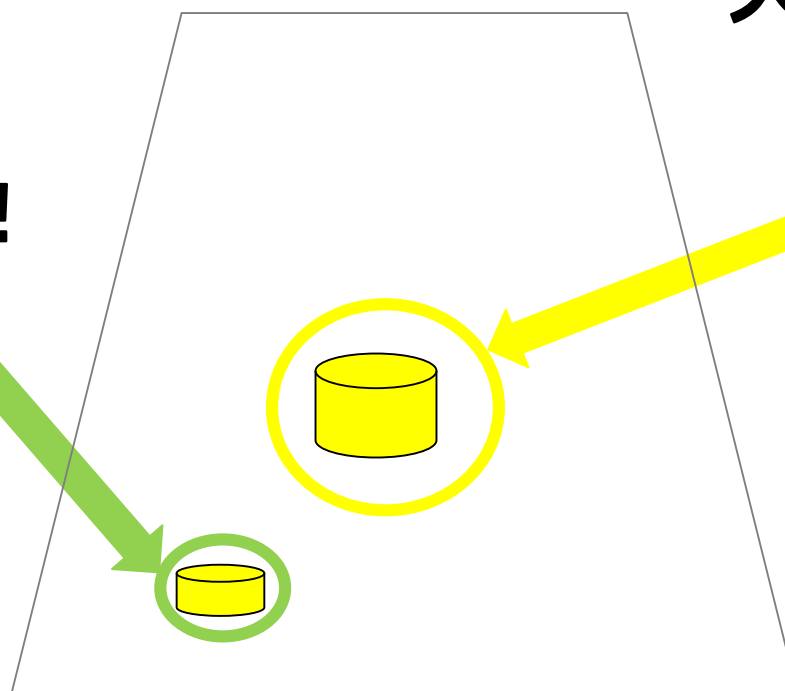
ボーイングB787 コックピット

重要なので
大きく中央に！

今までの
大きさを隅に！

ドアロックセレクター

ラダートリム

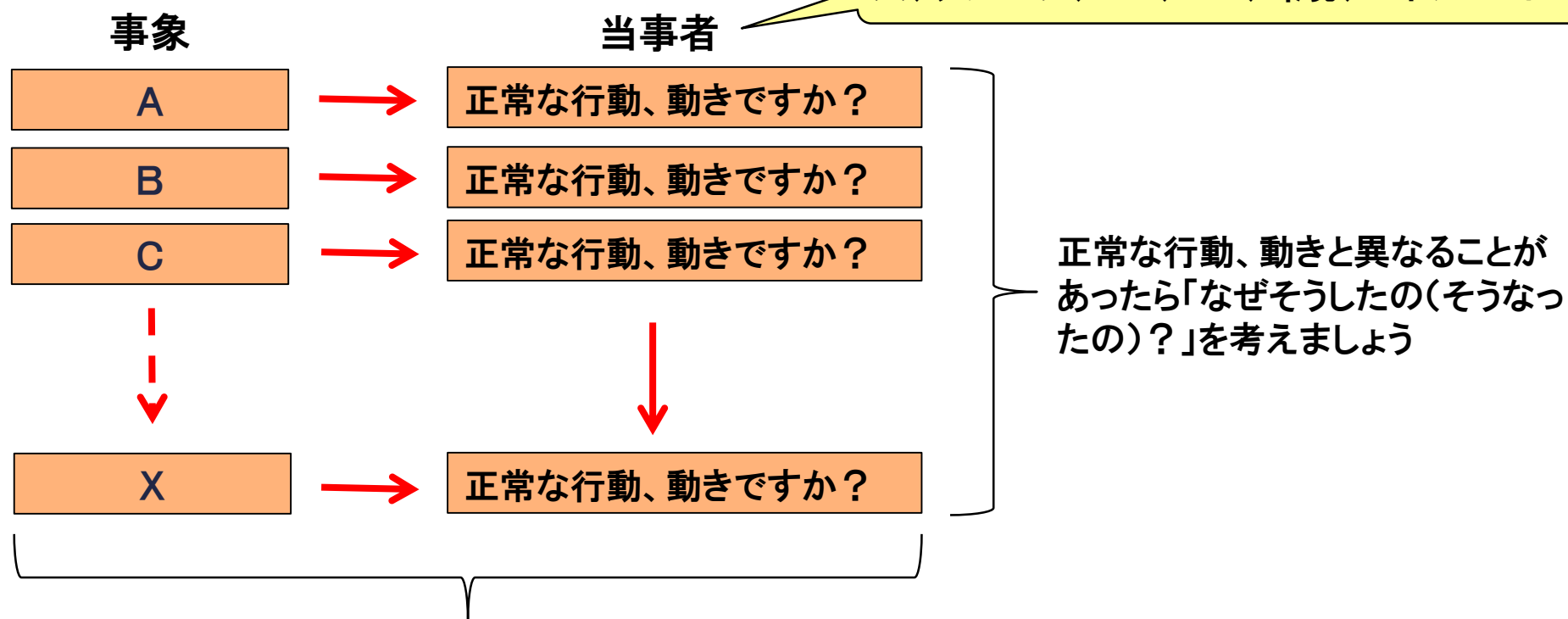


2-5. なぜなぜ分析・留意点

◆なぜなぜ分析が行き詰ったら、ここでちょっと確認しましょう

1. 分析前に事象を正確に把握していますか

人、グループ、SW、HW、環境、マネジメント...



正常な行動、動きと異なることがなかったら「**事象を正確に把握していない**」か、
「**正常な場合の事象に問題がある**」こととなります

2-5. なぜなぜ分析・留意点

2. 主語(当事者)を明確にしていますか

対策の対象(当事者)がはっきりしないと、対策を誤解します。

「作業者が〇〇
を間違えた」

なぜ？

「大きな騒音
が起きた」

なぜ？

騒音の当事者は誰？確認者？外部？

「壁が薄かった？」
「確認者が電話していた？」
「作業者が道具を落とした？」

3. 内容の繰り返しになっていませんか

「作業者が〇〇を忘れていた」

なぜ？

「作業者が思い出さなかった」



言い換えでなく、深堀をしていきましょう

(参考) 鍋木俊暁 鉄道総研人間科学部安全性解析研究室

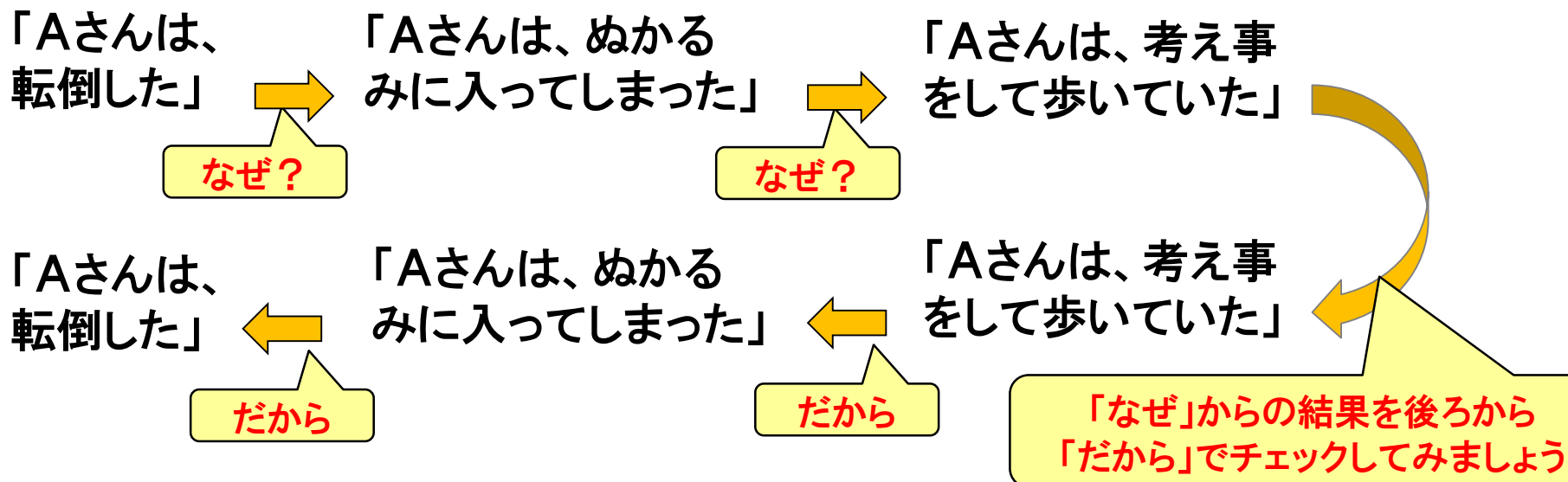
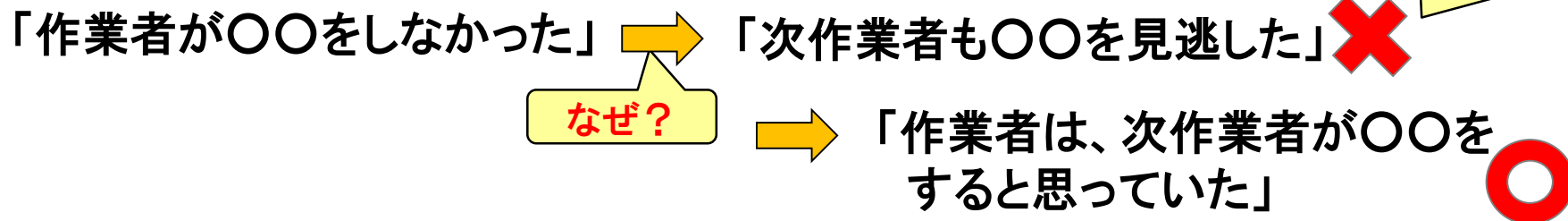
「ヒューマンファクター分析(なぜなぜ分析)の留意点と分析ツールの開発」 当事例用に変更しています。

<http://bunken.rtri.or.jp/PDF/cdroms1/0040/2017/0040002652.pdf>

2-5. なぜなぜ分析・留意点

4. 時間軸が逆になっていませんか

後に起きる行動が、前に起きる行動の理由にならない



(参考) 鑄木俊暁 鉄道総研人間科学部安全性解析研究室

「ヒューマンファクター分析(なぜなぜ分析)の留意点と分析ツールの開発」 当事例用に変更しています。

<http://bunken.rtri.or.jp/PDF/cdroms1/0040/2017/0040002652.pdf>

5. 「なぜ？」を諦めていませんか

「作業者が〇〇をしなかった」

なぜ？

「作業者は、初めての作業だった」

初めてじゃしょうがない

「リーダーは、作業者が初めての作業かどうか、事前確認をしていなかった」

6. 一般化した表現になっていませんか

「作業者が〇〇をしなかった」

なぜ？

「準備不足だった」

具体的に

「リーダーは、事前に作業者用のマニュアルを用意していなかった」

(参考) 鏑木俊暁 鉄道総研人間科学部安全性解析研究室

「ヒューマンファクター分析(なぜなぜ分析)の留意点と分析ツールの開発」 当事例用に変更しています。

<http://bunken.rtri.or.jp/PDF/cdroms1/0040/2017/0040002652.pdf>

2-6. 障害対策を検討する

対策の検討

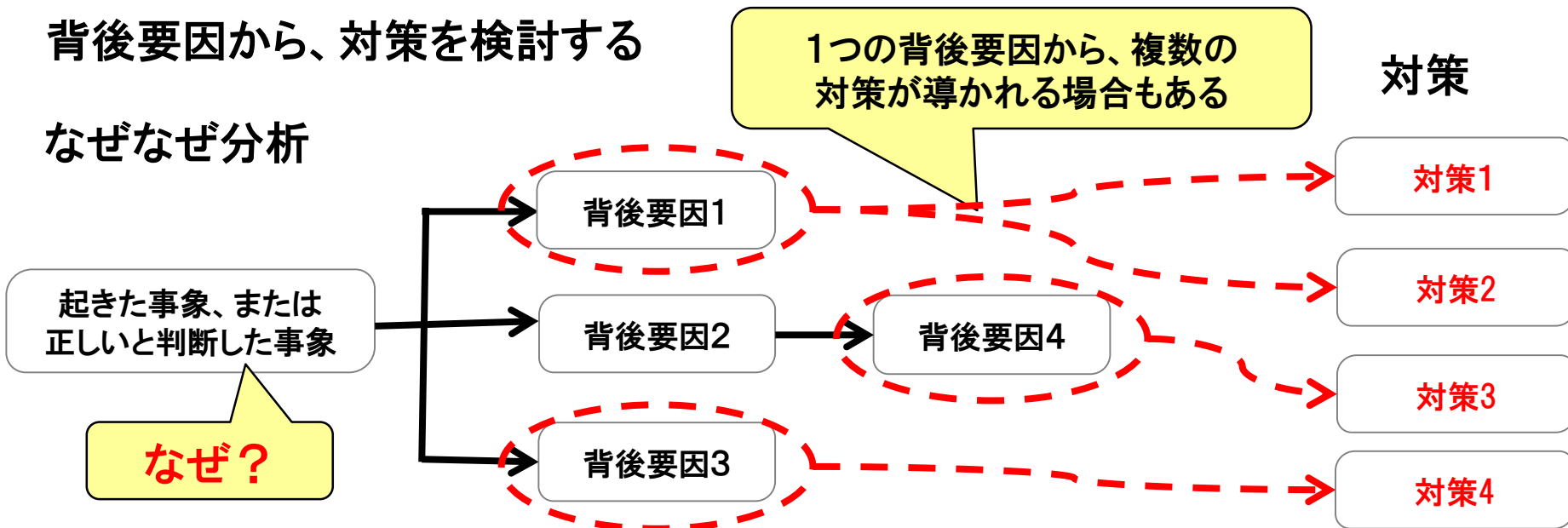
対策を検討してください。
過去の障害事例や教訓、技法などを参考にするのも有効です。

1. 直接原因→裏返し→対策
2. 根本原因→裏返し→本質対策(上位概念)
 - ・当事者がいる場合:徹底的に「改善」を追求
 - ・報道、報告書からの情報:あらゆる想定(想像力)

2-6. 障害対策を検討する

背後要因から、対策を検討する

なぜなぜ分析



m(マネジメント)の観点から分析する

S(ソフトウェア)、運用の観点から分析する

H(ハードウェア)の観点から分析する

E(環境)の観点から分析する

L-L(人)当事者、支援体制の観点から分析する



m(マネジメント)、組織を変える

S(ソフトウェア)、運用を変える

H(ハードウェア)を変える

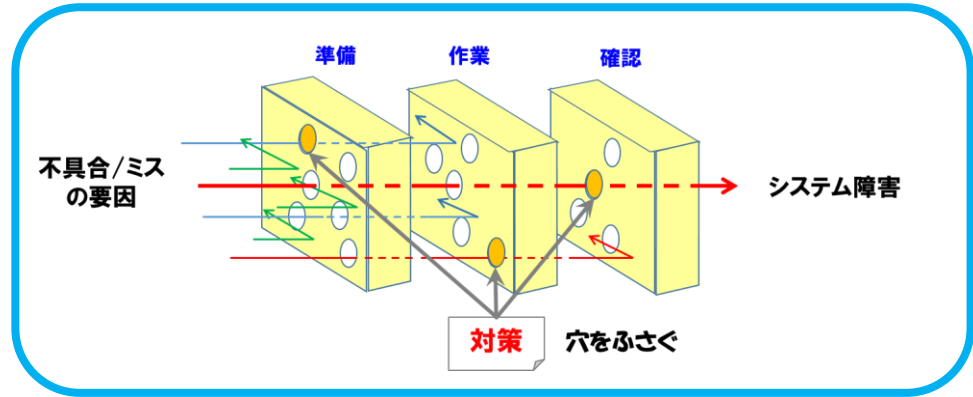
E(環境)を変える

L-L(人)支援体制を整える

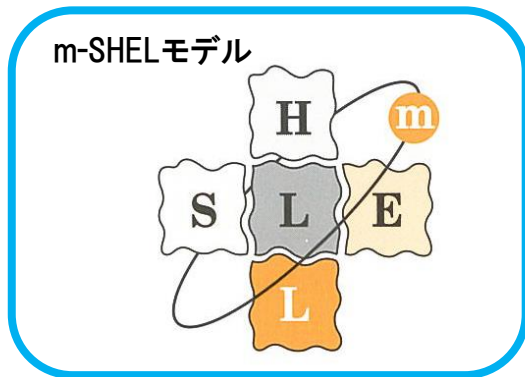
2-6. 障害対策を検討する

◆ **3つの観点**の組み合わせによる対策

(1) IT運用/開発プロセスの観点



(2) 関係性の観点



(3) 人間特性の観点

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				
	(なくす、減らす) やめる 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
IT対策	作業軽減		チェック機能	未然防止	能力向上				

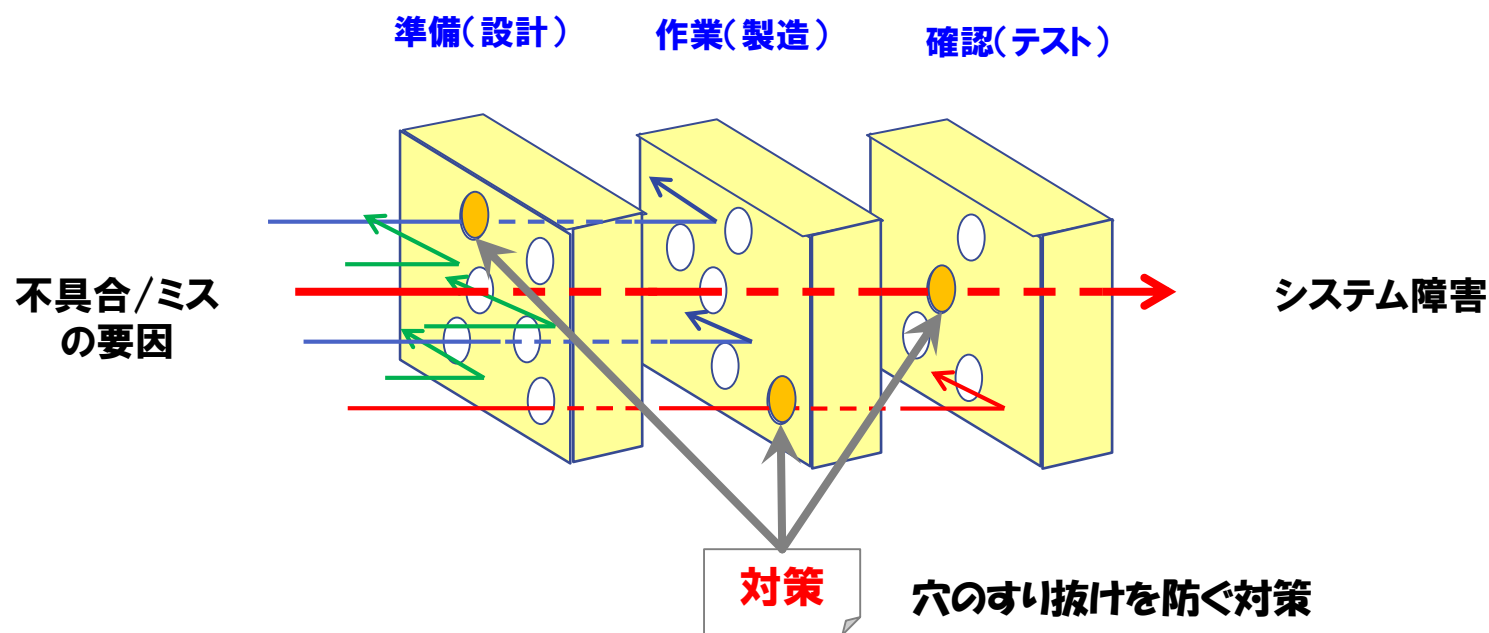
障害対策は、多層防御で防げ！

(参考)河野龍太郎 m-SHELモデル、ヒューマンエラー

2-6. 障害対策を検討する

(1) IT運用/開発プロセスの観点

誤りを防ぐ対策(プロセス)と誤りを見逃さない対策(前後のプロセス)を考える

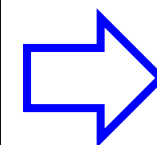


- ①どのプロセスで誤りを防ぐか、を考える。
- ②誤りを見逃さない対策を考える。

2-6. 障害対策を検討する

(2) 関係性の観点 と (3) 人間特性の観点 のマトリックス

優先度を比較し、対策を決定



検討順序



観点

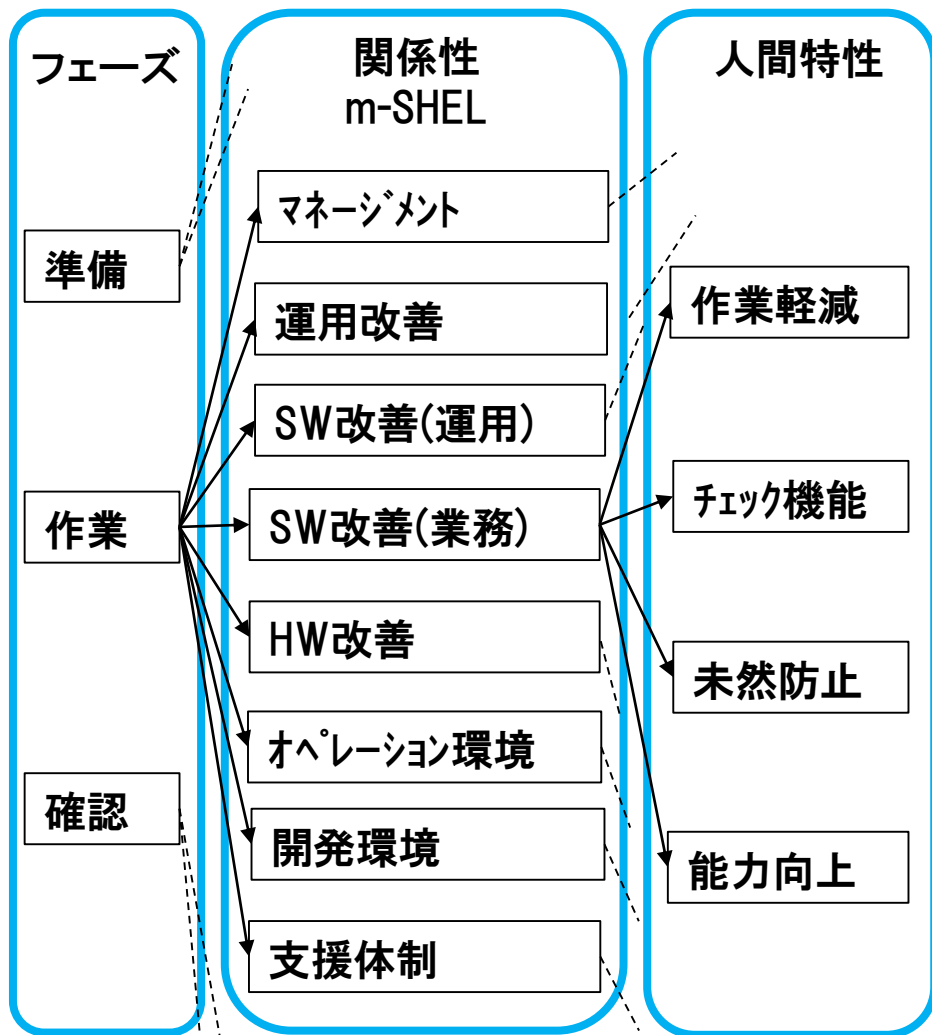
戦術的エラー対策の 発想手順	自動化(システム化)					確認/チェック・教育・訓練					
	作業環境への対策					作業者自身への対策					
	やめる(なくす、減らす)	できないようにする	分かりやすくする	やりやすくする	検出する	備える	知覚能力を持たせる	認知・予測させる	安全を優先させる	できる能力を持たせる	自分で気づかせる
m-SHELモデル											
m(マネジメント)、組織を変える											
S(ソフトウェア)、運用を変える											
H(ハードウェア)を変える											
E(環境)を変える											
L-L(人)支援体制を整える											

対策をマッピングし、対策の観点に漏れがないことを確認する

- ①作業軽減
- ②チェック機能
- ③未然防止
- ④能力向上

(参考)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

2-6. 障害対策を検討する



◆運用対策を考える

3つの言葉で対策を考えよう

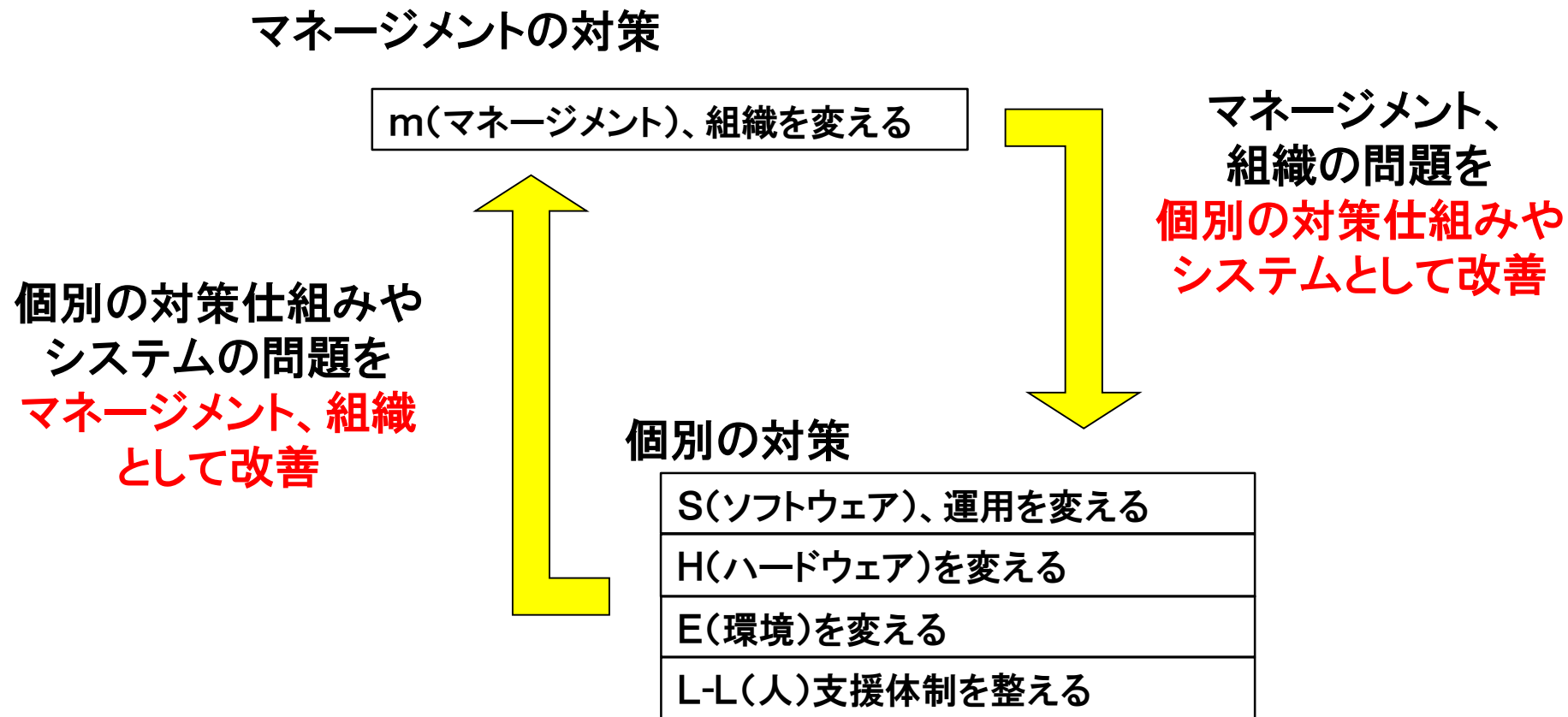
フェーズ 段階で、関係性 面での
人間特性 対策はないだろうか？

- ・準備段階で、体制面での未然防止策はないだろうか？
- ・準備段階で、運用面での作業軽減はできないだろうか？
- ・作業段階で、SW改善でのチェック機能はないだろうか？
- ・作業段階で、オペレーション環境での能力向上策は？
- ・確認段階で、支援体制のチェック機能はなにがあるか？
- ・確認段階で、開発環境での未然防止策はないだろうか？

開発フェーズでは、設計、製造、テスト等で、考えよう

2-6. 障害対策を検討する 補足①

◆ m-SHELモデルを基にITシステム障害対策



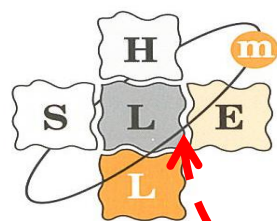
(参考) 野龍太郎 「医療におけるヒューマンエラー第2版」医学書院2014

2-6. 障害対策を検討する 補足②

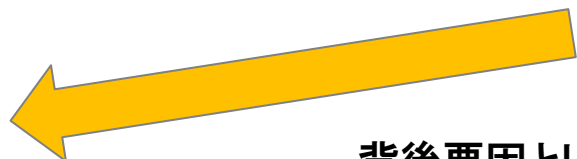
◆L毎に、m-SHELモデルが作られた場合は、それぞれの対策を検討

戦術的エラー対策の 発想手順			
m-SHELモデル			
m、組織を変える			
S、運用を変える			
Hを変える			
Eを変える			
L-L支援体制を整える			

戦術的エラー対策の 発想手順			
m-SHELモデル			
m、組織を変える			
S、運用を変える			
Hを変える			
Eを変える			
L-L支援体制を整える			



背後要因
ソフトウェア開発者



背後要因として
ソフトウェアの問題



直接原因
運用作業者

(参考)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

2-6. 障害対策を検討する 補足③

このセミナーでは、再発防止対策と未然防止対策を、以下のように整理しています。

◆再発防止対策

自社事例から背後要因を**事実に基づいて**、なぜなぜ分析から導き出す

◆未然防止対策

他社事例から背後要因を**知見と想像によって**、なぜなぜ分析から導き出す
さらに、リスク分析をおこない、未然防止対策を立てる

1. 直接原因、根本原因→**裏返し**→再発防止対策
2. 再発防止策→他社事例から自社の**リスク**を考える
→未然防止対策

2-7. 教訓化する

教訓化

最も影響があり注意すべきと感じた点を教訓にしてください。

- 1. 忘れない言葉(キャッチコピー)**
- 2. 短く、ストレートな表現**
- 3. 否定形でなく、肯定形**
- 4. 具体的な行動へ導く表現**

3. 自分で教訓を作成する

- 3-1. 過去のセミナー参加者の教訓作成事例から
- 3-2. 過去のセミナー参加者教訓作成事例

説明 17:10-17:15
(5分を想定)

自分の教訓を作る

演習 17:15-17:30
(15分を想定)

あなたが今までに経験したITシステム関連の障害事例をあげ、問題・原因・対策・効果・教訓にわけて考えてみてください。

(事例は抽象化していただいて結構です。失敗だけでなく、成功事例でも結構です。ご自身の経験以外の報道事例、教訓集を活用されても良いです。)

【ご提案】

職場に持ち帰って、是非取り組んでみて下さい。
→ あなたがリーダーとなって、
職場のみなさんと議論するとより効果的です。

3-1. 過去のセミナー参加者の教訓作成事例から

- ・何事も丸投げしない。
- ・ベンダを過度に信頼するな。
- ・性能試験においては、可能な限り本番環境と同じシステム構築、データ、負荷シナリオを用いること。
- ・開発スケジュールの遅れは、プロジェクトマネジメントの観点だけでなく、システムアーキテクチャの適否を検討すべきである。
- ・テスト・検証対象は列挙するだけでなく、システムの全体構成と対比して漏れがないかチェックすべし。
- ・テストケースの網羅性を担保するためには、業務に精通した複数の有識者でレビューを行なう。
- ・ソフトウェアの稼動検証は、実行環境の条件・違いを理解した上で行うべし。
- ・バックアップシステムを寝かせておくのはもったいない！！
- ・障害は必ずある。最悪のケースに備えよう。出来ることを順番に考えて備えよう。
- ・検証環境について、甘く考えない！！
- ・システム本番を延期する勇気を持つ。
- ・品質向上に抜け道なし。地道にコツコツやるべし。
- ・旧システムからのデータ移行は慎重に！！
- ・エンドユーザを苦勞させないシステムを作る。
- ・何かを変更する時は、何かが及ぼす影響を調査する。
- ・現場にやさしく丁寧に。

3-1. 過去のセミナー参加者教訓作成事例から

- ・解除や延期の変更点管理に注意しよう。できれば、対処策も合わせよう。
- ・障害は発生する。次の対応を考えて行動。
- ・忙しくても担当者任せにせず、クロスチェックできるように体制を組む。
- ・本番作業は、経験と独断だけに頼るな！
- ・ソースコードを適切に保つ努力は、システムの品質を安定させる。
- ・あたり前と思うことでも記録に残す。伝達する。
- ・安易にリリースを追加するな。
- ・先輩といえども得手、不得手がある。
- ・見えない情報にも注意しよう。
- ・常に危機感を忘れない。定期的にはリスクマネジメントを実施する。
- ・更新履歴は重要だ！
- ・体調が悪い者は作業しない。必ず確認者と2人以上で。
- ・通常おこなっているリモートメンテナンスも、必ずおこなう。
- ・テストする際は、十分なデータを用意すること。
- ・システムが停止できるタイミングで、勇気をもってファームウェアのアップをおこなうようにする。優先順位を考える。
- ・システム間でデータの取決めをしていたとしても、データチェックをおこなう処理は入れること。
- ・環境の変化に過敏になろう。
- ・指示者と作業者の意識共有は十分にすべし。

3-1. 過去のセミナー参加者教訓作成事例から

- ・複数装置に対して、一度に「設定」「解除」を繰り返し送信する場合は、一部がエラーとなることを忘れないようにする。
- ・変更管理の仕組みを整備し守ること。
- ・情報・機能については、アーキテクトが責任をもつ。
- ・要件定義・設計は、裏付けを取って詰める。
- ・顧客の目線で業務を考え、システムの設計をおこなうこと。決して仕様書に記載されている内容のみに頼らないこと。
- ・本番・テスト環境差分をプロジェクトで管理する。
- ・突発事象もあわてずに！1つ1つの作業をきっちり終わらせる。
- ・お客様にもシステム安定運用に協力してもらおう。
- ・事前確認は必ず。不測の事態を想定する。
- ・システムの(ユーザからの)利用範囲を明示的に線引きする。不適切な利用は、システムの的に制限する。
- ・システムの全体構成は、全関係者に明らかにする。
- ・要求仕様を固めるのが成功のカギ。
- ・リスクを極小せよ。

3-2. 過去のセミナー参加者教訓作成事例①

・問題	商用環境において、検証時には見られなかった処理状況が発生し、ファイル更新が止まってしまった。
・原因	検証環境と商用環境で実装されている機能ライセンスに差分があったためにこのような状況が発生していることが判明。
・対策	検証環境と商用環境の差分を事前に確認して環境を揃える様にする。
・効果	検証環境でライセンスに絡む問題は全て確認が可能となる。
・教訓	検証環境について甘く考えない！！

具体的な行動を生むような教訓が役に立つ？

3-2. 過去のセミナー参加者教訓作成事例②

・問題	システム本番稼働後の不具合が多数発生。
・原因	・厳しい受注条件でプロジェクト進行 ・品質より納期を優先
・対策	・リリース判定を正しく実施 ・品質条件を満足しない場合、システム本番しない。
・効果	長い目で見て、顧客満足度向上
・教訓	システム本番を延期する勇気を持つ

個人で対応は難しい？
組織としてどう教訓とするか？

3-2. 過去のセミナー参加者教訓作成事例③

・問題	業務システムが停止し、支払業務が1日停止した。 バックアップシステムの切替えも失敗し、全業務が停止した。
・原因	・十分なシステムテスト不足。 ・想定外のオペレーション。 ・バックアップ回復処理の失敗。
・対策	・障害対応フローの文書化。 ・予行訓練 ・定期的な保守。情報共有。
・効果	不測時にも慌てず、次策の対策で対応できる。
・教訓	障害は必ずある。最悪ケースに備えよう。 できることを順番に考えて備えよう。

備えあれば憂いなし。
マネジメントにどう反映するか

3-2. 過去のセミナー参加者教訓作成事例

参加者のみなさまの教訓で気になった事例は・・・

- ◆テスト環境と本番環境が異なっていたため、テストでは発見されず、本番環境で起きてしまった障害
- ◆開発が順調に行かず、本番で障害が多発した
ベンダ任せ、納期遅れ、テスト漏れ、有識者不参加・・・
- ◆バックアップ切替え失敗
- ◆データのバックアップが不十分
データのバックアップを取っていなかった、
バックアップデータが使えるか不安・・・

教訓集にも類似の事例が出ています

結構、「データを消してしまった」報道事例が出ています

サイバー攻撃(ランサムウェア)が脅威です

現場の皆さまの声を拾うと・・・

「日常の運用、開発が忙しく、障害対策まで手が回らない・・・」のでは？

「情報処理システム高信頼化教訓集(ITサービス編)」を
より有効にご活用いただくためのメールマガジンの登録について

未だ登録されていない方は、是非登録を！



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)
<https://www.ipa.go.jp/sec/system/lesson.html>

（お問合せは）
社会基盤センター（IKC）
産業プラットフォーム部
コネクテッドインダストリーズグループ
ikc-info@ipa.go.jp