



## 大規模ミッションクリティカルシステムにおける トラブル再発防止管理事例

### - トラブル多発時の包括分析手法の提案と適用事例紹介 -

2023年9月8日

株式会社NTTデータ モバイルビジネス事業部

掛川悠

# 01

はじめに

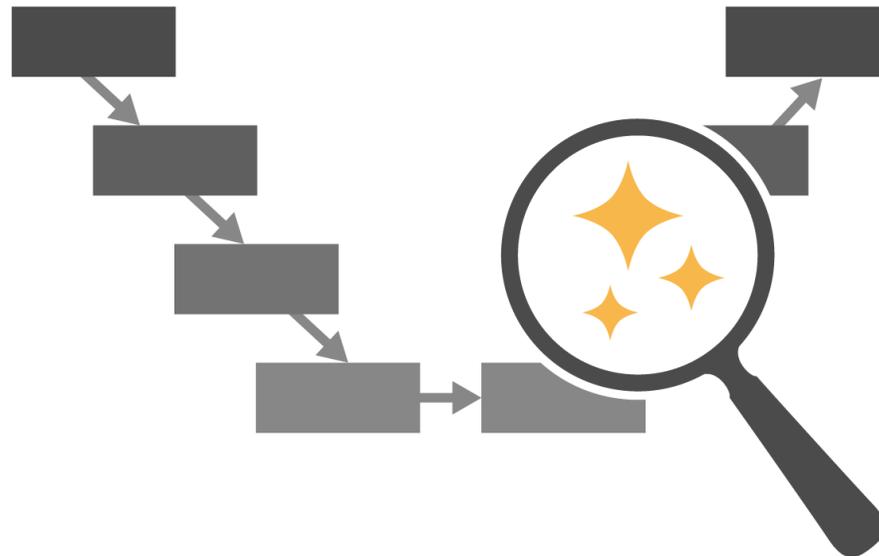
# はじめに

プロジェクトにトラブルはつきもの。

再発防止せよといわれるが、分析は稼働もかかるし意外と難しい。

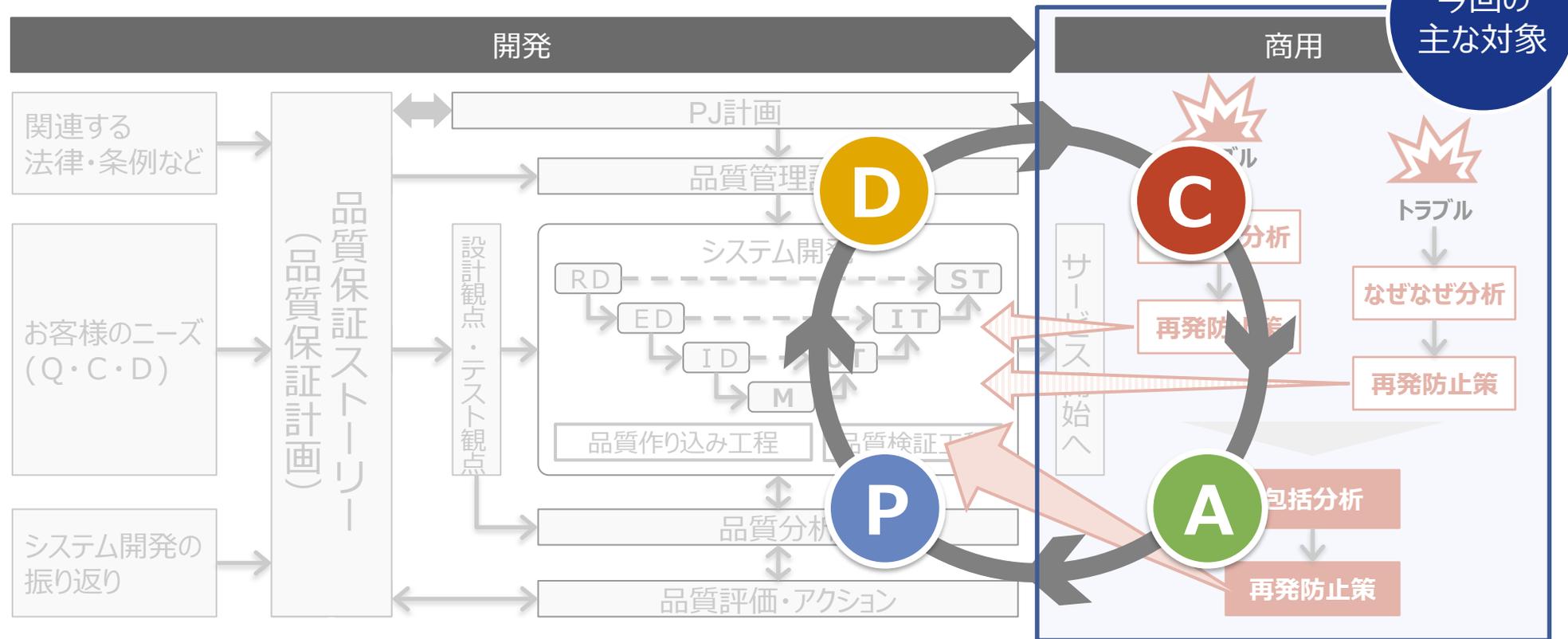
実際、社会的なインフラシステムのトラブルは昨今でもなくなっておらず、その意味でトラブルの再発防止は古くて新しい課題といえる。

そこで、本発表では、大規模ミッションクリティカルシステムの10年以上の開発で培われた再発防止管理のポイントをまとめ、特に難しい**トラブル多発時の原因分析**に焦点をあてた。



# 再発防止管理とは？

トラブルの再発を防ぐためのプロセスおよびノウハウ



※1 本資料で扱うトラブルはAPバグ、設定値誤り、商用作業ミスといったシステム関連のトラブルを指す

※2 本資料では主に商用トラブルを想定した再発防止管理について記載しているが、開発工程中のトラブルに対しても基本的な考えは同じ。

**再発防止管理**  
⇒自律的な品質改善のエンジン！

# 目次

## 1.はじめに

## 2.プロジェクト概要

## 3.再発防止管理について

【基本】個別トラブルの分析 - なぜなぜ分析（テーラリングあり） -

【応用】トラブル多発時の分析 - 包括分析 - ← **メインはここ**

### ■トラブル多発時における再発防止管理の事例紹介

- 1.トラブル多発時の原因分析しくじり
- 2.包括分析における原因分析 - ポイント -
- 3.包括分析における原因分析 - ポイントを例とともに見ていく -
- 4.包括分析における原因分析 - 進め方（SEPG主体） -
- 5.包括分析のコストに関する考察

## 4.まとめ

# 02

## プロジェクト概要

※以下、プロジェクトをPJと略記

# PJ特性

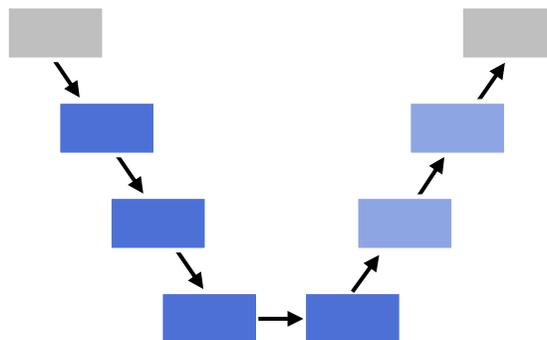
## 対象システム

### 通信基盤システム



## 開発手法

### Waterfall型



## 開発周期

### 年四回以上の開発



## 開発の特徴

- 1つのバージョンで複数のサービスを開発
- 1つのサービスは複数の機能で構成



## 組織構造

マトリックス型（強）

## 要員数

常時1000人以上

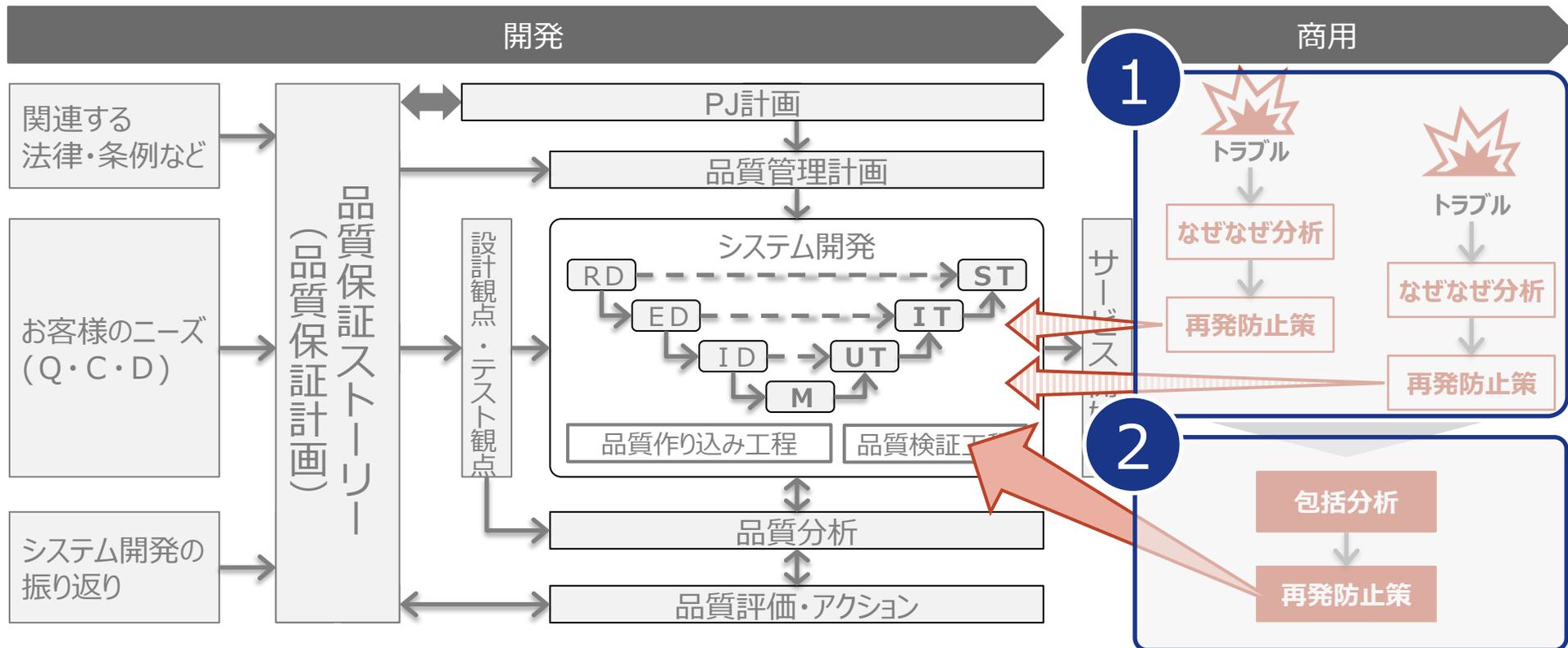
大規模かつ繰り返し開発が継続するPJ

# 03

## 再発防止管理について

# 大規模ミッションクリティカルシステムにおける再発防止管理の課題

- 【課題1】全てのトラブルに対してスピードと質を両立した分析を徹底し続けるのが難しい**  
 → **【基本】個別トラブルの分析 -なぜなぜ分析 (テーラリングあり) -**
- 【課題2】トラブル多発の原因分析をスピードと質を両立した形で実施するのが難しい**  
 → **【応用】トラブル多発時の分析 -包括分析- ← **メインはここ****

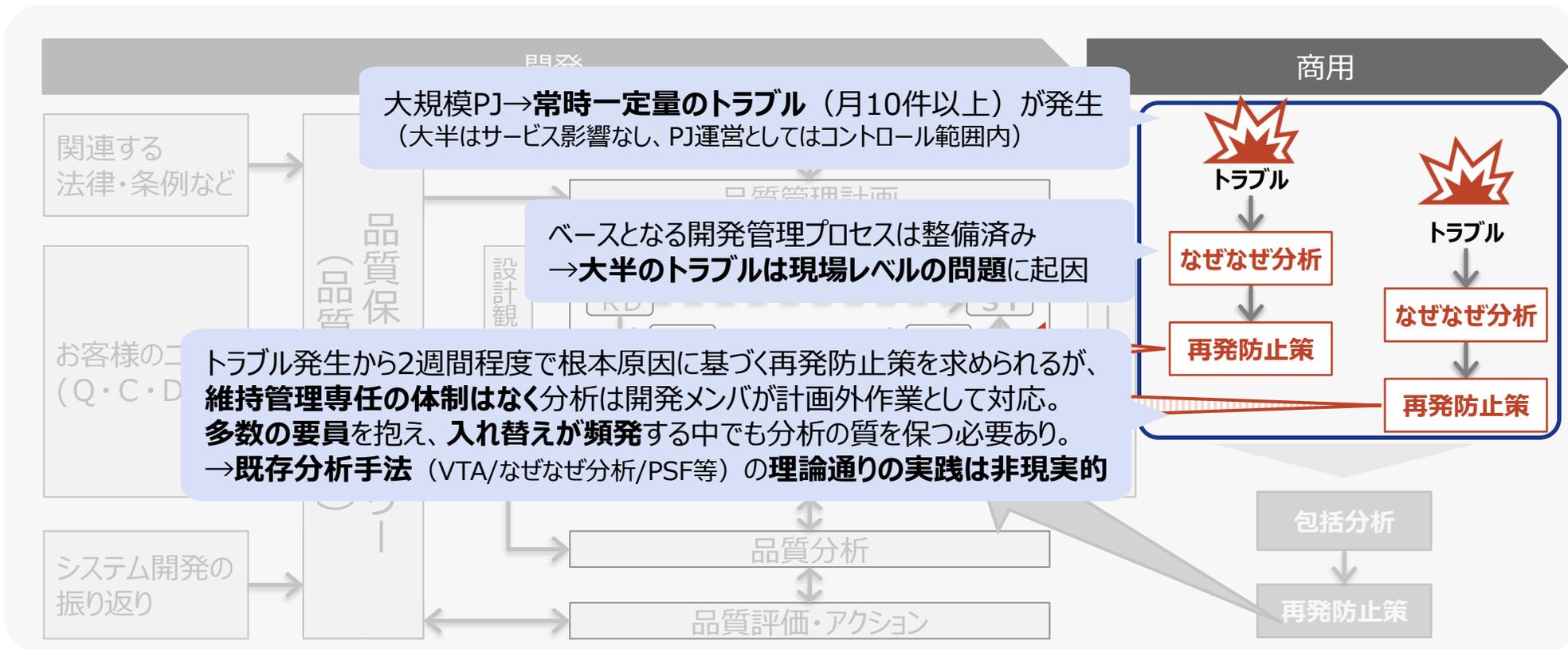


# 再発防止管理の構造 ①

## 【基本】個別トラブルの分析

### - なぜなぜ分析（テーラリングあり） -

# 当PJにおける個別トラブルの分析課題



## 課題1

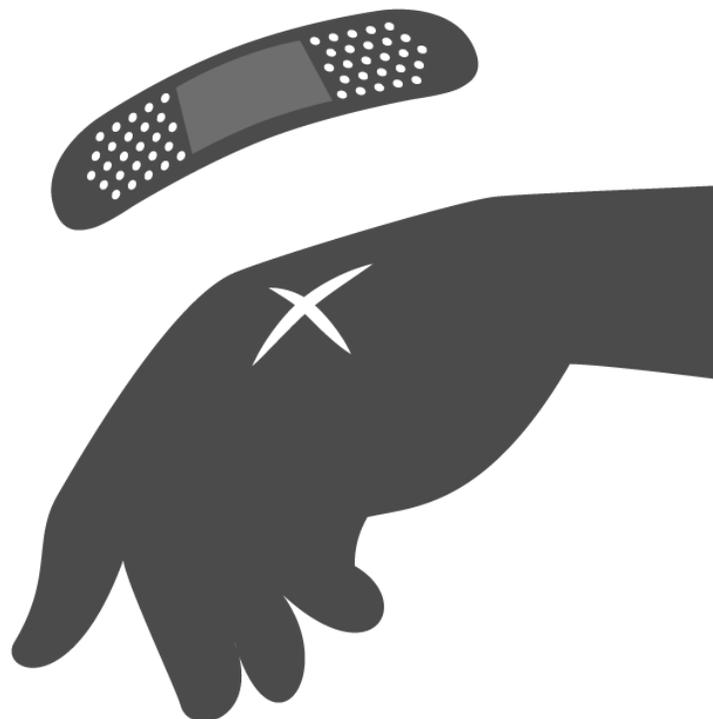
多数の要員/頻発する要員入れ替え/限られたリソースといった**体制制約**の中で、**全てのトラブル**に対して**スピードと質を両立**した**分析**をいかに**徹底し続ける**か

# なぜなぜ分析の目的

## 目的

なぜなぜ分析の目的は、  
限られた期間・コストで最大効果を生む対策を見つけること

絆創膏が  
傷より小さいと  
覆えない



絆創膏が  
傷より大きいと  
無駄ができる

**傷にあったサイズの絆創膏なら無駄なく覆える**

## なぜなぜ分析を実践する際の観点

混入		トラブル事象	処置内容
要摘			
Ver			

設計/製造担当	レビューア	テスト担当
↓なぜ？	↓なぜ？	↓なぜ？
↓なぜ？	↓なぜ？	↓なぜ？

プロセス改善	適用範囲	スケジュール

横展開	対象範囲	スケジュール/結果

## Step1 基礎情報の整理

機能、発生日、トラブル事象、処置内容、混入工程/要摘出工程/混入バージョン等

## Step2 ヒアリングによる原因追及

- 作り込みレビューすり抜け/テストすり抜けごとになぜを繰り返して**根本原因**（プロセス・仕組み/ツール/材料の問題）を探る。
- 人のせいにせず、**プロセス問題**に帰着
- **3現主義**で分析

## Step3 プロセス改善と横展開

根本原因の裏返しで、**再発防止策**（プロセス改善）、**横展開**を立案・実行

# なぜなぜ分析を実践する上でのポイント①

- 「人」以外の原因（プロセス・仕組み/ツール/材料）を見つけてプロセス改善！
- NG事例集として、注目ポイントごとに、誤った分析例 / なぜ誤りか？ / 正しい考え方を整理

NG事例集の例)

注目ポイント	誤った分析例	なぜ誤りか？	正しい考え方
曖昧って 何がどう曖昧なの？	<p>◆なぜ、設計時にバグを作り込んだ？ 担当者がセルフチェックでPD書とDD書を 突き合わせたが検出できなかった。 ↓</p> <p>◆なぜ、セルフチェックをすり抜けた？ DD書に処理シーケンスやエラー時の動 作が記載されている必要があったが、その <b>記載が曖昧</b>であった。</p>	<p>「記載が曖昧」では、何 が悪かったのかが分から ないため、有効な再発 防止策に繋がらない</p>	<p>問題点を具体的に特定する</p> <p>【正しい分析】 ◆なぜ、セルフチェックをすり抜けた？ DD書が文章だけの記述でシーケンス やマトリックスで整理されていなかった ため、条件の分岐や多重に対して抜 け漏れに気付けなかった。</p>

# なぜなぜ分析を実践する上でのポイント②

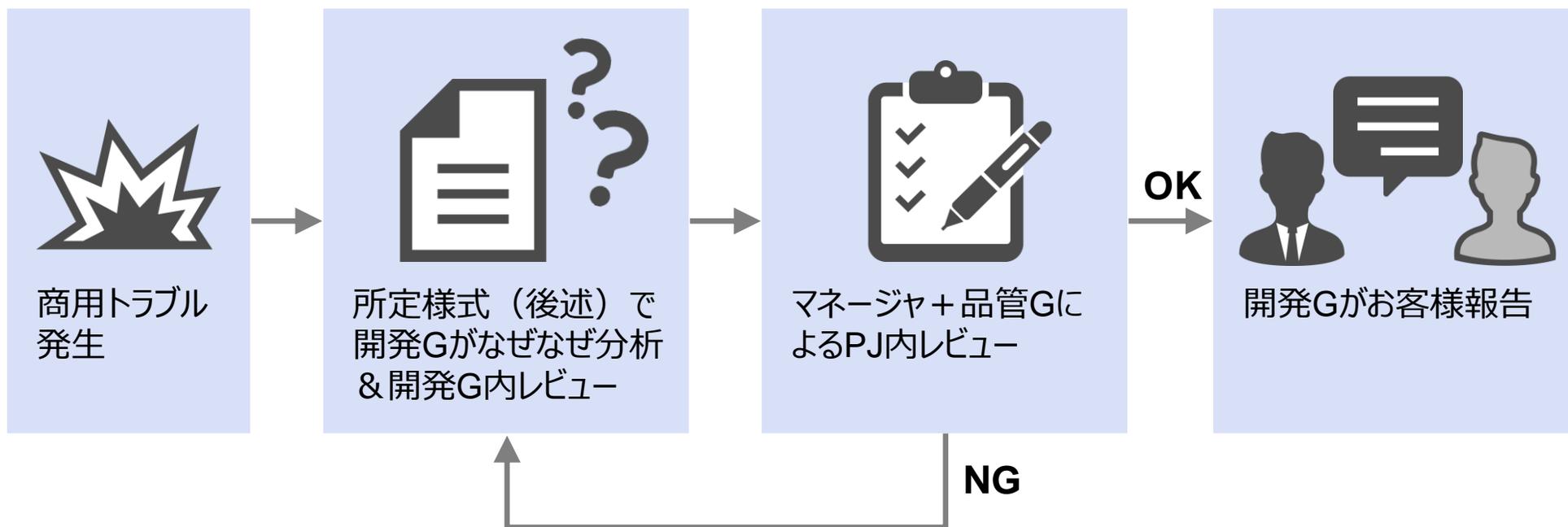
NG事例集での注目ポイント例)

カテゴリ	なぜなぜ分析の注目ポイント例	
作り込み	<ul style="list-style-type: none"> <li>曖昧って何がどう曖昧なの？</li> <li>なぜ思い込んだの？</li> <li>やるべきことが決まってないのでは？</li> <li>そもそもプロセスを守って作業したの？</li> </ul>	<ul style="list-style-type: none"> <li>対策を決めてから原因に結び付けてない？</li> <li>担当者目線での追求を。想定はダメ</li> <li>当たり障りのない言葉で逃げてはダメ</li> <li>今までうまく行っていたのには理由があるはず</li> </ul>
レビューすり抜け	<ul style="list-style-type: none"> <li>正しいレビューとは何？</li> <li>何を根拠にOKしたの？</li> <li>レビューイの問題にしてはいけない</li> </ul>	
テストすり抜け	<ul style="list-style-type: none"> <li>考慮不足で終わらせてはいけない</li> <li>試験項目が抜けていた理由は？</li> </ul>	
再発防止策	<ul style="list-style-type: none"> <li>対策が曖昧</li> <li>現実的にできて、続けられる？</li> <li>禁句（周知、徹底）</li> <li>チェックを重ねてもムダ</li> </ul>	<ul style="list-style-type: none"> <li>グループ間の責任分担を決めよう</li> <li>対策は量ではない。原因分析に基づきピンポイントで対策を</li> </ul>
横展開	<ul style="list-style-type: none"> <li>横展開って何？</li> <li>原因分析ができてないから範囲が絞れない</li> <li>他のグループにも横展開すべき問題</li> </ul>	

# トラブル分析会議

- 商用トラブルが発生した場合、「トラブル分析会議」を開催し、  
作込/摘出遅延の根本原因、再発防止、横展開実施内容の確認を行う  
主催 マネージャ + 品質管理担当
- 全トラブルの分析状況をチケット管理システムで一元管理。定期的な督促で運用徹底

## トラブル分析会議のフロー



# なぜなぜ分析のまとめ

## 目的

限られた期間・コストで最大効果を生む対策を見つけること

## 実践する上でのポイント

### ■ 分析

分析担当者によらず、全トラブルに対してスピードと質を両立した分析を行うために、

- **作り込み、レビュー、テストすり抜けごとになぜを繰り返して根本原因を探る**  
→PJ標準の分析様式
- **根本原因は「人」以外の原因**（ツール/プロセス・仕組み/材料）に着目して**プロセス改善**  
→NG事例集

### ■ 運用

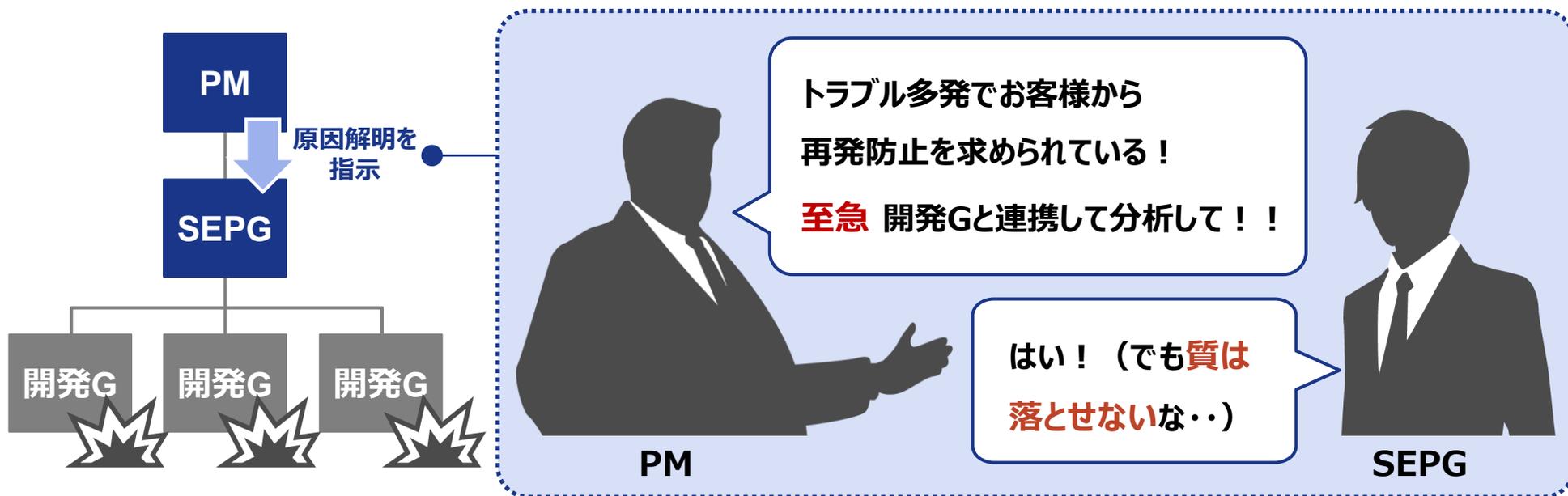
- 開発グループのトラブル分析に対するマネージャレビュー（トラブル分析会議）を義務化し、チケット管理で実行徹底

## 再発防止管理の構造 ②

# 【応用】トラブル多発時の分析 - 包括分析 -

# 背景 -トラブル多発時の分析課題-

## トラブルが多発すると多発の原因分析が必要になる



### 状況

- 一定規模（数十人以上 / 数十ks以上 / 複数ベンダーetc）のPJで、特定の案件や時期に偏ったトラブル多発により何らかのQCD影響が出て、お客様から再発防止を求められている。

例

- ・ 商用APバグが多発し、商用APバグ密度が目標値超過
- ・ 結合試験でインシデント多発し、そのリカバリーで納期遅延

- 各トラブルごとのなぜなぜ分析と再発防止は、トラブル発生都度実施し、顧客承認済み。

### 課題2

大規模かつ多様な開発手法、グループ、案件が共存する開発で発生した**トラブル多発**の原因分析を、**限られた体制**の中で**スピードと質を両立**した形でいかに実施するか

# 包括分析とは？

## 包括分析

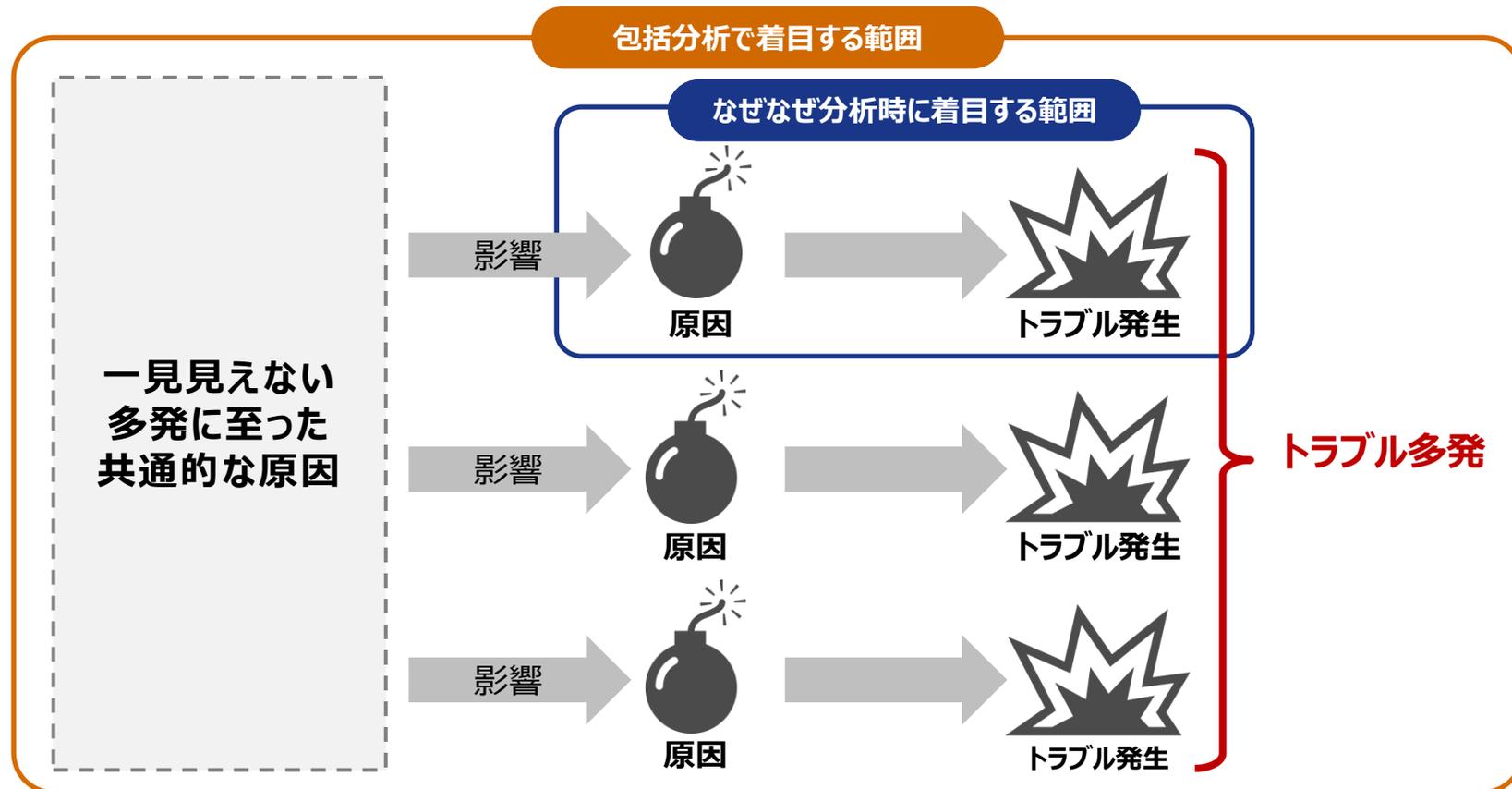
### 複数のトラブルの関係性に着目した包括的な分析

個々のトラブルのなぜなぜ分析では、基本的に該当トラブルに閉じて分析を行う。

しかし、**トラブル多発時**は、多発に至った共通的な原因が潜んでいることが多い。

そのため、個々のトラブルのなぜなぜ分析とは別に**複数のトラブルの関係性に着目した包括的な分析**を行う必要がある。以降、このような分析を包括分析と呼ぶ。

トラブル多発により何らかのQCD影響が出て、お客様から再発防止を求められている状況



# トラブル多発時の包括分析の目的

## 目的

トラブル「多発」の原因を明らかにして、  
多発を防ぐために最も費用対効果の高い再発防止策（※）を見つけること。

※開発工程中や商用リリース後間もない場合は、該当開発の潜在バグを刈り取る横展開も必要

### なぜなぜ分析

個々のトラブルに対する  
再発防止策をみつける



### 包括分析

トラブル「多発」に対する  
再発防止策をみつける



# 包括分析の大まかな流れと本資料の対象

包括分析は以下の**3ステップ**で進める。

本資料では最も大事な「**step2 : 原因分析**」にフォーカスをあてる。

## step1 : 分析対象トラブルの特定

報告相手と対象トラブルの目線合わせを行い、分析の全体感を共有する

## step2 : 原因分析

トラブル「多発」の原因を分析する

次スライド以降で説明

## step3 : 再発防止策の立案

多発を防ぐために最も費用対効果の高い再発防止策（※）立案する

※開発工程中や商用リリース後間もない場合は、該当開発の潜在バグを刈り取る横展開も必要

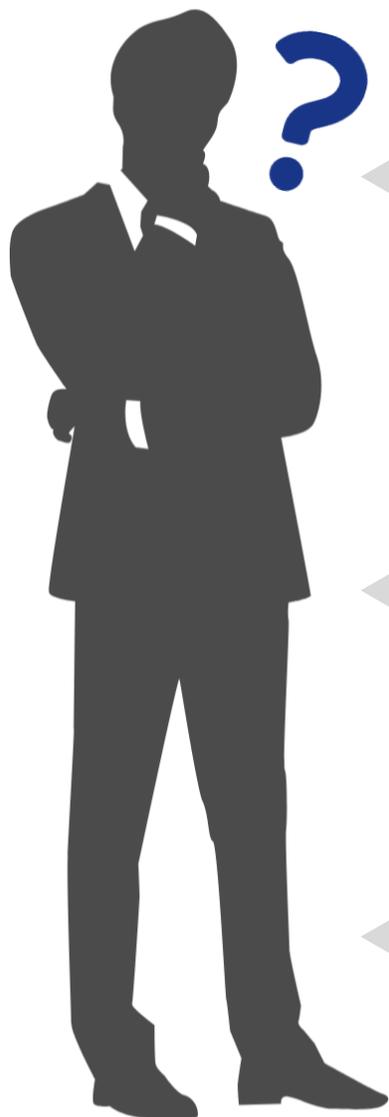
# トラブル多発時における再発防止管理の事例紹介

1. トラブル多発時の原因分析しくじり
2. 包括分析における原因分析 - ポイント -
3. 包括分析における原因分析 - ポイントを例とともに見ていく -
4. 包括分析における原因分析 - 進め方（SEPG主体） -
5. 包括分析のコストに関する考察

# トラブル多発時における再発防止管理の事例紹介

1. **トラブル多発時の原因分析しくじり**
2. 包括分析における原因分析 - ポイント -
3. 包括分析における原因分析 - ポイントを例とともに見ていく -
4. 包括分析における原因分析 - 進め方（SEPG主体） -
5. 包括分析のコストに関する考察

## トラブル多発時の原因分析しくじり① 根本原因に共通性なし



お客様

やっぱり今回はトラブル多いね。  
で、原因は？

各トラブルごとの分析結果は既に報告受けているから知ってるけど、  
何か共通性はないの？

他案件は大丈夫なのに、なんで案件Aに限ってトラブルが多発したの？  
案件Aに何か問題があったんじゃないの？

そこも疑ったのですが、各トラブルの**根本原因**に案件特性に関する  
ものは見当たらなかったです。

商用トラブルの一覧はこちらです。

トラブル1の**根本原因**は資材管理のXXXで、  
トラブル2の**根本原因**は母体設計書のYYYで、…、  
**いずれもプロセス改善による再発防止策を実施済みです。**

表面的には案件Aに偏ってはいますが、  
**根本原因には共通性は見当たりませんでした。**



報告者

# トラブル多発時の原因分析しくじり① 根本原因に共通性なし



お客様

え？じゃあトラブル多発の原因はわからないってこと？

各トラブルはいずれもなぜなぜ分析でプロセスの問題を見つけ、改善を行っているので再発は防止できます。

答えになってないよ。  
それは個々のトラブルの話でしょ。  
なぜ今回多発したの？  
我々も経営層に報告しなくちゃいけないのに、  
たまたまですなんていえないよ。  
やり直し！

…（根本原因に共通性ないんだから仕方ないじゃん）

## しくじり①

個々のトラブルに対しての根本原因は把握済みだが、  
各トラブルの根本原因に共通性を見つけられないケース

### このケースに陥りやすい人の特徴

- 基本的に忠実になぜなぜ分析をしっかりとってきた
- 根本原因はプロセス問題に帰着させることも重々わかっている



報告者

# トラブル多発時の原因分析しくじり①：何が足りなかった？

## できていること

個々のトラブルに対して、根本原因は把握。  
根本原因に共通性がないことは確認済。

**ボトムアップ分析** は実施済み

### ボトムアップ分析

**概要** トラブル起点で根本原因の共通性を探る

**分析観点** トラブルの傾向（混入・すり抜け原因/案件/人etc）



## できていないこと

トラブル起因の分析しかせず  
**トラブル「多発」の原因分析**ができていない。

1件のトラブルには複数の原因が絡むことが多く、  
間接原因により顕在化する根本原因もある。  
つまり、ボトムアップ分析だけでは、多発原因に  
たどり着かないことが多い。

**間接原因** にも着目した原因分析が必要

※間接原因  
トラブルに直結しないが、根本原因の顕在化に影響する  
間接的な原因

## 改善ポイント

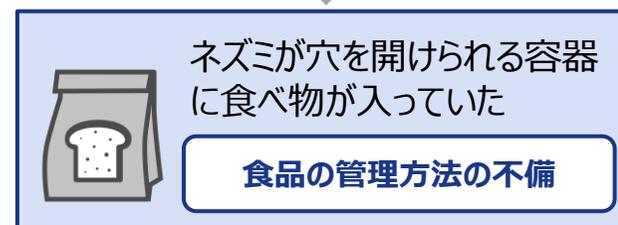
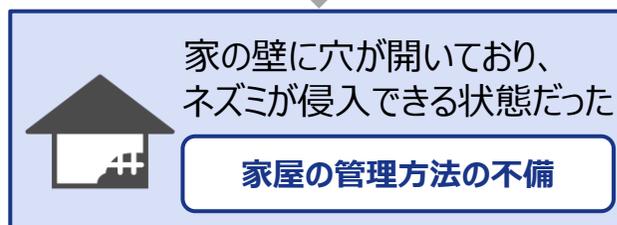
トラブル多発の原因を探るときは、  
各トラブルの間接原因まで含めた分析が必要！

## 【喩え話】直接原因、根本原因、間接原因の関係性

## 家の「食べ物が消える」トラブルが多発



直接原因



根本原因

Q. 管理不備は前からあったのになぜ最近ネズミが増えたのか？



ネズミ狩のエキスパート

## 猫がいなくなった

これまでは、猫がネズミに気づき、事前に退治していたため食べ物が  
ネズミに食べられることがなかった。

しかし、猫がいなくなったことで、ネズミは家の中に入っても退治されるこ  
とがなくなり、食料に手を付けることができるようになった。

間接原因

状況を俯瞰することで  
今回だけ問題が  
多発した原因を探る

# トラブル多発時の原因分析しくじり② 効果の薄い壮大な対策



お客様

で、なんでトラブル多発したの？

各トラブルの共通的な原因を探るために、  
**開発Gに対して案件特性、当初のリスク、体制、スケジュール等をヒアリングした結果、該当案件に必要な業務有識者の稼働ひっ迫で新規参画者のフォローができていなかったことがわかりました！**

それで再発防止はどうするの？

業務有識者を増やすためにXXX、また、新規参画者の立ち上げを早くするために設計書の補足資料を作成してXXX。

すごい大変そうだけど、そんなこと**本当にできるの？**

がんばります！！



報告者

商用トラブルの一覧はこちらです。  
各トラブルごとの根本原因分析および再発防止策立案は  
全て完了して報告済みです。

# トラブル多発時の原因分析しくじり② 効果の薄い壮大な対策

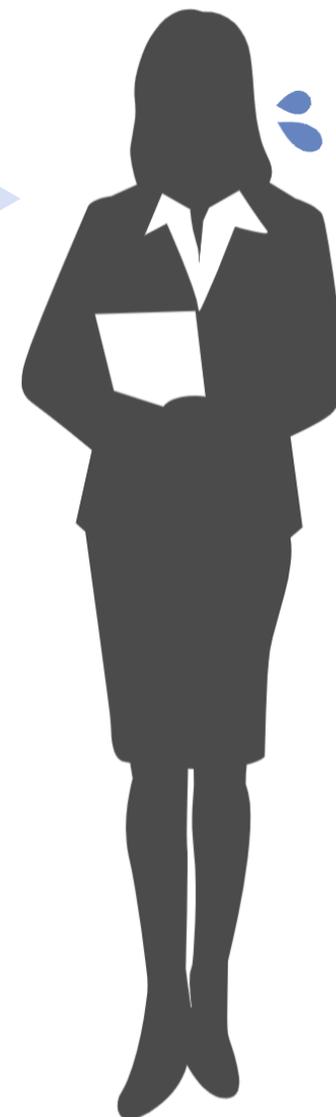


お客様

ちなみに、この対策やったら今回のトラブルのどれが防げるの？

(あわてて一覧で関連トラブルを探しながら)  
ええと・・・、**20件中4件**です・・・ね。

え？ たった2割？  
じゃあ**全然対策**になってないじゃん。  
やり直し！



報告者

## しくじり②

トラブルの根本原因とは別に  
開発時の問題を調査し多発の原因を探るが  
的外れな対策を打ち出すケース

### このケースに陥りやすい人の特徴

- 比較的大きな組織でマネジメントよりの業務が多い
- 現場担当者と話す機会はそれほどない

# トラブル多発時の原因分析しくじり②：何が足りなかった？

## できていること

トラブルの根本原因とは別に  
開発時の問題を調査し多発の原因を探ること。

**トップダウン分析** は実施済み※

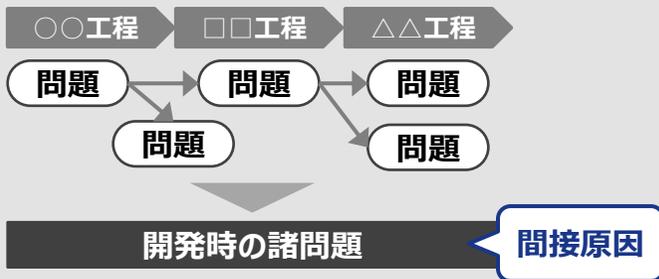
### トップダウン分析

#### 概要

時系列等、多角的な観点で開発中の問題を探る

#### 分析観点

開発時の問題（案件特性/リスク認識/体制/改修内容/マネジメントetc）



## できていないこと

トップダウン分析で判明した開発時の問題と  
トラブルとの**因果関係の検証**ができていない

トップダウン分析で開発時の問題（間接原因）  
が明らかになっても、それが各トラブル発生（根  
本原因）にどう影響したのかがわからないため、  
**分析が仮説で止まっている。**

根本原因と間接原因の

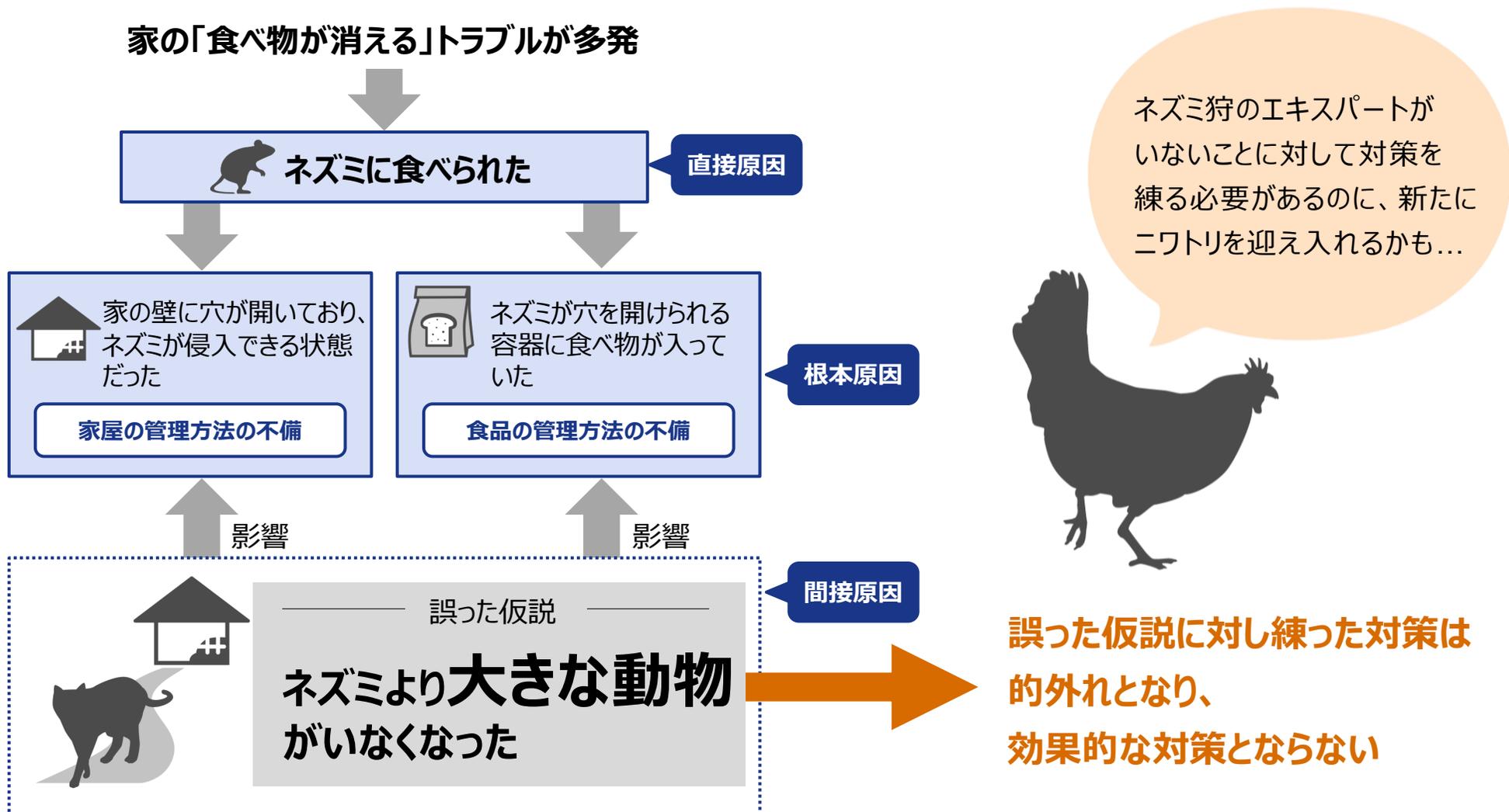
**因果関係** を明らかにする必要がある

## 改善ポイント

- トップダウン分析とボトムアップ分析**両方の視点からの分析が必要。**
- トップダウン分析とボトムアップ分析の**因果関係を検証**することが必要。

## 【喩え話】因果関係の検証の重要性

Q. 誤った仮説のまま間接原因の対策を練るとどうなるか？



# トラブル多発時の分析しくじりのまとめ

トラブル多発時の原因分析で  
陥りやすい失敗

改善ポイント

ケース  
1

トラブル起因の分析しかせず  
トラブル多発の原因分析ができていない

- 各トラブルの間接原因まで含めた分析

ケース  
2

仮説ベースで分析した開発時の問題と  
トラブルとの因果関係の検証ができて  
おらず、的外れな対策を立てる

- トップダウン分析、ボトムアップ分析  
両方の視点からの分析
- トップダウン分析とボトムアップ分析  
双方の因果関係の検証

# トラブル多発時における再発防止管理の事例紹介

1. トラブル多発時の原因分析しくじり
2. **包括分析における原因分析 - ポイント -**
3. 包括分析における原因分析 - ポイントを例とともに見ていく -
4. 包括分析における原因分析 - 進め方（SEPG主体） -
5. 包括分析のコストに関する考察

## 包括分析の原因分析イメージ

ポイント

例

実践

## 包括分析

視点 PMと開発G/SEPG

## トラブル多発時

概要 複数のトラブルの関係性に着目した包括的な分析

分析観点 ①ボトムアップ分析、②トップダウン分析、③それらの紐づけ

## トップダウン分析

視点 PM

概要 時系列等、多角的な観点で開発中の問題を探る

分析観点 開発時の問題（案件特性/リスク認識/体制/改修内容/マネジメントetc）



開発時の諸問題

間接原因

影響

影響

影響

## ボトムアップ分析

視点 開発G/SEPG

概要 トラブル起点で根本原因の共通性を探る

分析観点 トラブルの傾向（混入・すり抜け原因/案件/人etc）

Aプロセス不備

Bプロセス  
□□不備Cプロセス  
××不備

多発

## なぜなぜ分析

視点 開発G

## 個々のトラブル

概要 トラブルの根本原因を探る

分析観点 「人」以外の原因（ツール/プロセス・仕組み/材料）

Aプロセス  
○○不備Aプロセス  
△△不備Bプロセス  
□□不備Cプロセス  
××不備

根本原因

手順誤り

設定ミス

ロジック誤り

処理ルート漏れ

直接原因



# 包括分析における原因分析のポイント

ポイント

例

実践

包括分析における原因分析はトラブル「多発」の原因を明らかにするために行う。

## Point ボトムアップ分析、トップダウン分析、その紐づけの3つすべてが必要

### ①ボトムアップ分析

⇒何が起こったのか

**分析観点** トラブル傾向

### ②トップダウン分析

⇒どんな背景でトラブルが起きたか

**分析観点** 開発時の問題

### ③トップダウン分析と ボトムアップ分析の紐づけ

**分析観点** 因果関係

#### 納得感のない 分析の例

- **ボトムアップ分析しかない**  
→個々のトラブルの根本原因はわかるが、多発原因（≒共通要素）は不明。たまたま・・・。
- **トップダウン分析しかない or トップダウンとボトムアップの紐づけが欠けている**  
→本当に対策の効果が出るのかあやしい。しかも、無駄に壮大な対策になりがち。

## ボトムアップ分析観点例

ポイント

例

実践

視点

開発G/SEPG

概要

トラブル起点で根本原因の共通性を探る

ポイント

「人」以外の原因（ツール/プロセス・仕組み/材料）を見つけてプロセス改善

分析観点	観点例
トラブルの発生傾向	案件、開発グループ、機能
トラブルの混入原因傾向	混入工程、混入プロセス、設計・製造担当者、混入原因
トラブルの レビューすり抜け原因傾向	レビューア、レビューすり抜け原因
トラブルの 試験すり抜け原因傾向	工程、観点、試験項目作成者、試験担当者、摘出不可原因

P41-45に  
例ありP41-45に  
例ありP41-45に  
例あり

# トップダウン分析観点例

ポイント

例

実践

視点

PM

概要

時系列整理を含む多角的な観点で開発中の問題を探る

ポイント

できたこと/できなかったこと、各事実の因果関係を明らかにするのが大事

分析観点	観点例
当初のリスク認識と対策	リスク認識と対策状況（会議体資料やリスク管理シートに記載した観点など）
案件背景	要件確定時期、仕様変更発行量、短期開発、並行開発、他Gとの調整や影響の受けやすさ
開発体制	担当者/レビューアの業務知識、スキルレベル、稼働状況（逼迫や割り込みなど）、自G/他Gの担当分担
改修内容 技術的負債	製品や新規採用方式など技術的な実装難易度、改修による母体影響範囲の広さ
マネジメント	想定外作業の現場影響調整、および経験の浅い担当者フォロー状況、開発担当者のプロセス不徹底
開発中品質	開発期間中の品質定量値の妥当性（他案件と比較して不足がないか、異常値でないか）、当該機能/担当者のレビュー密度、エラー密度、レビューすり抜け率、要摘試験の試験密度、バグ密度、試験すり抜け率etc 開発期間中に発覚した品質問題、及び実施した品質強化との関係性

P41-45に  
例ありP41-45に  
例ありP41-45に  
例あり

# 包括分析における原因分析のポイントのまとめ

ポイント

例

実践

## ■ 原因分析には3つの要素が必要

### ① ボトムアップ分析

⇒何が起こったのか

**分析観点** トラブル傾向

### ② トップダウン分析

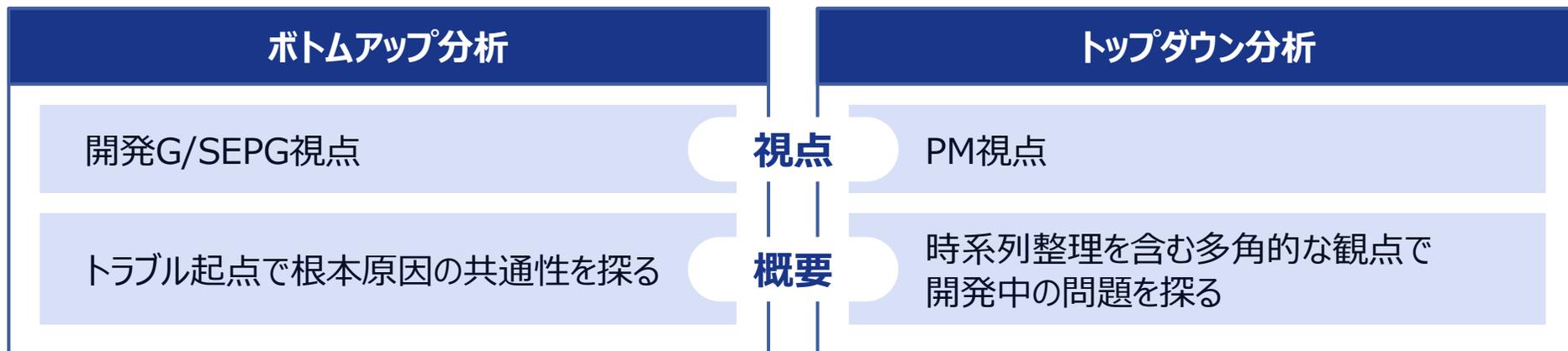
⇒どんな背景でトラブルが起きたか

**分析観点** 開発時の問題

### ③ トップダウン分析と ボトムアップ分析の紐づけ

**分析観点** 因果関係

## ■ ボトムアップ分析とトップダウン分析と包括分析の関係性



これらを紐づけトラブル多発の原因を明らかにするのが**包括分析**

# トラブル多発時における再発防止管理の事例紹介

1. トラブル多発時の原因分析しくじり
2. 包括分析における原因分析 - ポイント -
- 3. 包括分析における原因分析 - ポイントを例とともに見ていく -**
4. 包括分析における原因分析 - 進め方 (SEPG主体) -
5. 包括分析のコストに関する考察

# 原因分析例（1/5）

ポイント

例

実践

以下の順番で包括分析における原因分析の事例を見ていく

ボトムアップ分析

トップダウン分析

トップダウン分析と  
ボトムアップ分析の  
紐づけ

※なぜなぜ分析は終えた状態と仮定

## 原因分析例 (2/5)

ポイント

例

実践

## 1. ボトムアップ分析

※観点例はP38

視点 開発G/SEPG

概要 トラブル起点で根本原因の共通性を探る

ボトムアップ分析で複数のプロセス不備が判明。しかし、過去開発でも存在した不備なのに  
**なぜ今回に限ってそれらが同時多発的に顕在化したのか、つまり「多発の理由」が説明できない。**

ボトムアップ分析で検出した問題  
(SEPG視点の問題)

傾向A	傾向B	傾向C	傾向D
設計時の新規 参画者フォロー	PT期間中に判明した 性能課題への対応	検証環境 変更管理	試験の 不具合管理

なぜなぜ分析で判明した根本原因を  
4つの傾向ごとに分類

発生した トラブル	1			
	2			○
	3			○
	4			○
	5		○	
	6			○
	7		○	
	8			
	9			
	10	○	○	
	11			
	13	○		
	14		○	

※○がゼロのトラブルは個別要因

## 原因分析例 (3/5)

ポイント

例

実践

## 2. トップダウン分析

※観点例はP37

視点 PM

ポイント 時系列整理を含む多角的な観点で開発中の問題を探る



**予定と実績に乖離があったことが判明。  
どういった背景で乖離が生まれたのが深堀。**

# 原因分析例 (4/5)

## 2. トップダウン分析

※観点例はP37

視点 PM

ポイント 開発中の出来事を時系列で整理し問題を探る

**ボトムアップ分析で検出した問題**

- 傾向A 設計時の新規参画者フォロー
- 傾向B PT期間中に判明した性能課題への対応
- 傾向C 検証環境変更管理
- 傾向D 試験の不具合管理

2020年			2021年				
10月	11月	12月	1月	2月	3月	4月	5月
開発線		DD	PD/M/UT		SI1	SI2	PT
予定 (性能系)		新規参画者フォロー		ログ管理/分析製品評価期間	性能検証期間		
実績 (性能系)		新規参画者フォロー (規定プロセスのみ)		ログ管理/分析製品評価期間	性能検証期間		

当初の予定は  
どうなっていたか?

実績は?

開発時に  
何が起きていた?

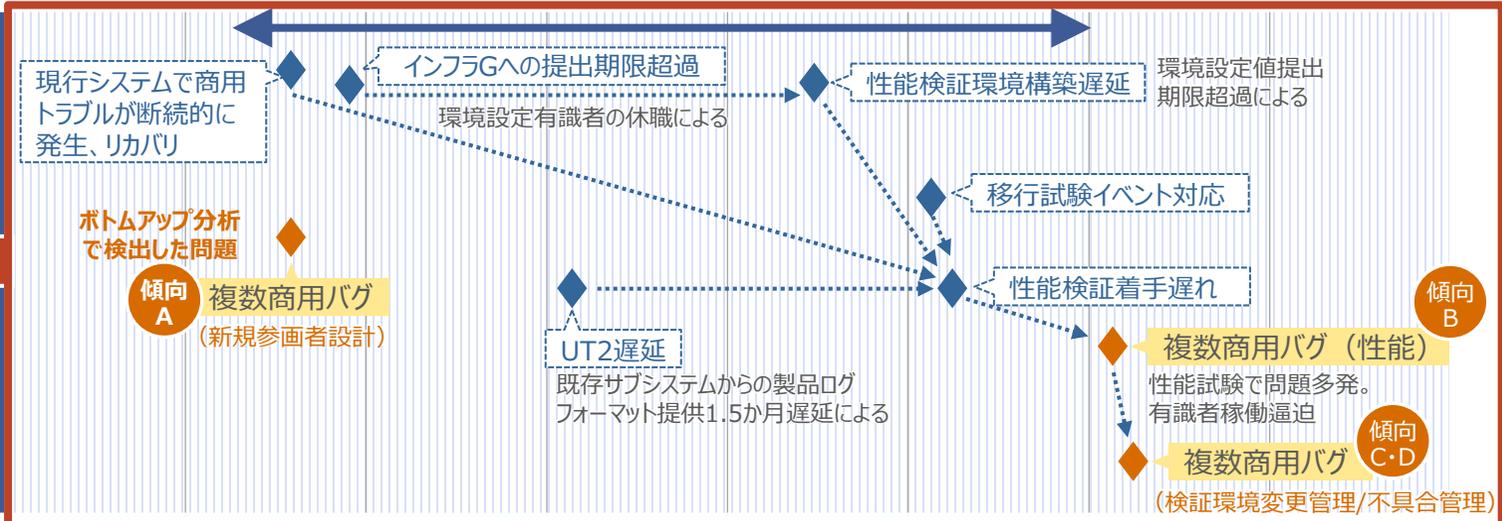
**想定外の問題多発**

マネジメント面の  
プロセス/体制の不備

有識者の稼働逼迫

経験の浅い担当者の  
プロセスが不徹底

**トラブル発生**



一部有識者の稼働逼迫と、それに伴う経験の浅い担当者へのフォロー不足によるプロセスの脆弱化

つまり、「マネジメント面のプロセス/体制の不備」が間接原因

## 原因分析例（5/5）

## 3. トップダウン分析とボトムアップ分析の紐づけ

視点 PMと開発G/SEPGの両方

概要 仮説ベースで分析した開発時の問題とトラブルとの因果関係の検証

	ボトムアップ分析で検出した問題 (SEPG視点の問題)				トップダウン分析で検出した問題 (PM視点の問題)	根本原因 × 間接原因 で顕在化したトラブル
	傾向A	傾向B	傾向C	傾向D	マネジメントプロセス/体制不備	
	設計時の新規 参画者フォロー	PT期間中に判明した 性能課題への対応	検証環境 変更管理	試験の 不具合管理		
1					○	
2			○		○	●
3				○	○	●
4			○		○	●
5		○				
6				○	○	●
7		○				

発生した  
トラブル

複数のプロセス不備の顕在化に影響

トップダウン分析&因果関係検証によりボトムアップ分析だけでは出せない対策を打てた

- 例)
- ・マネジメントプロセス改善（メンバ稼働可視化等）
  - ・複数G間でのマネジメントプロセスの集約・標準化
  - ・PMOの体制強化..etc

といったマネジメント面のプロセス/体制強化により、業務有識者が担っていた管理業務を切り出し稼働平準化を図る。  
また、稼働状況監視を強化し、問題発生時は速やかに上位層エスカレーションし対策につなげる。

→ 2021年の再発防止策適用以降、該当グループの商用バグは0件。包括分析の有効性を示せた。

※○がゼロ

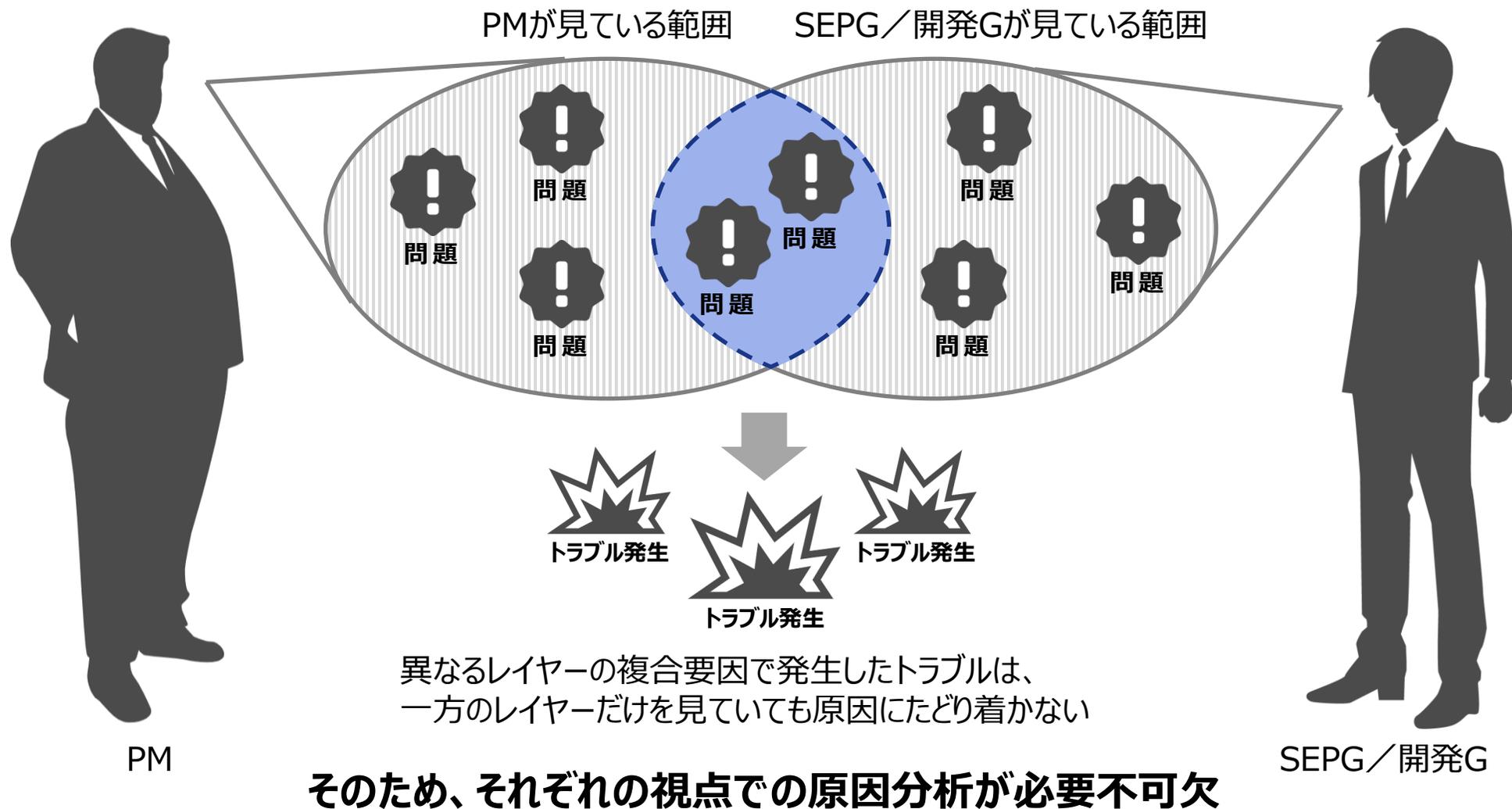
## 【考察】トラブル多発の原因構造①

ポイント

例

実践

トラブルの多発は全く異なるレイヤーの複合要因で発生することが多い。



## 【考察】トラブル多発の原因構造②

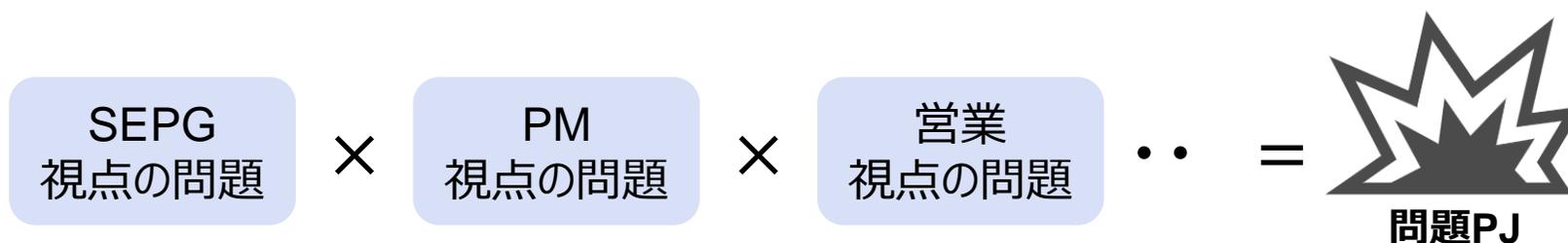
ポイント

例

実践

実は、不採算案件等、大きく問題化するPJの原因構造は基本的に同じ。

**全く異なるレイヤーの複合要因で問題化**している（3つ以上もざら）。



## 原因分析例と考察のまとめ

ポイント

例

実践

各トラブルごとの根本原因だけではなく  
視点を変えることで見えてくる問題も含めて考えなくては  
多発の原因にはたどりつけつかない



**開発関係者、それぞれの視点に立って  
包括的に原因を分析していくことが重要**

# トラブル多発時における再発防止管理の事例紹介

1. トラブル多発時の原因分析しくじり
2. 包括分析における原因分析 - ポイント -
3. 包括分析における原因分析 - ポイントを例とともに見ていく -
4. **包括分析における原因分析 - 進め方（SEPG主体） -**
5. 包括分析のコストに関する考察

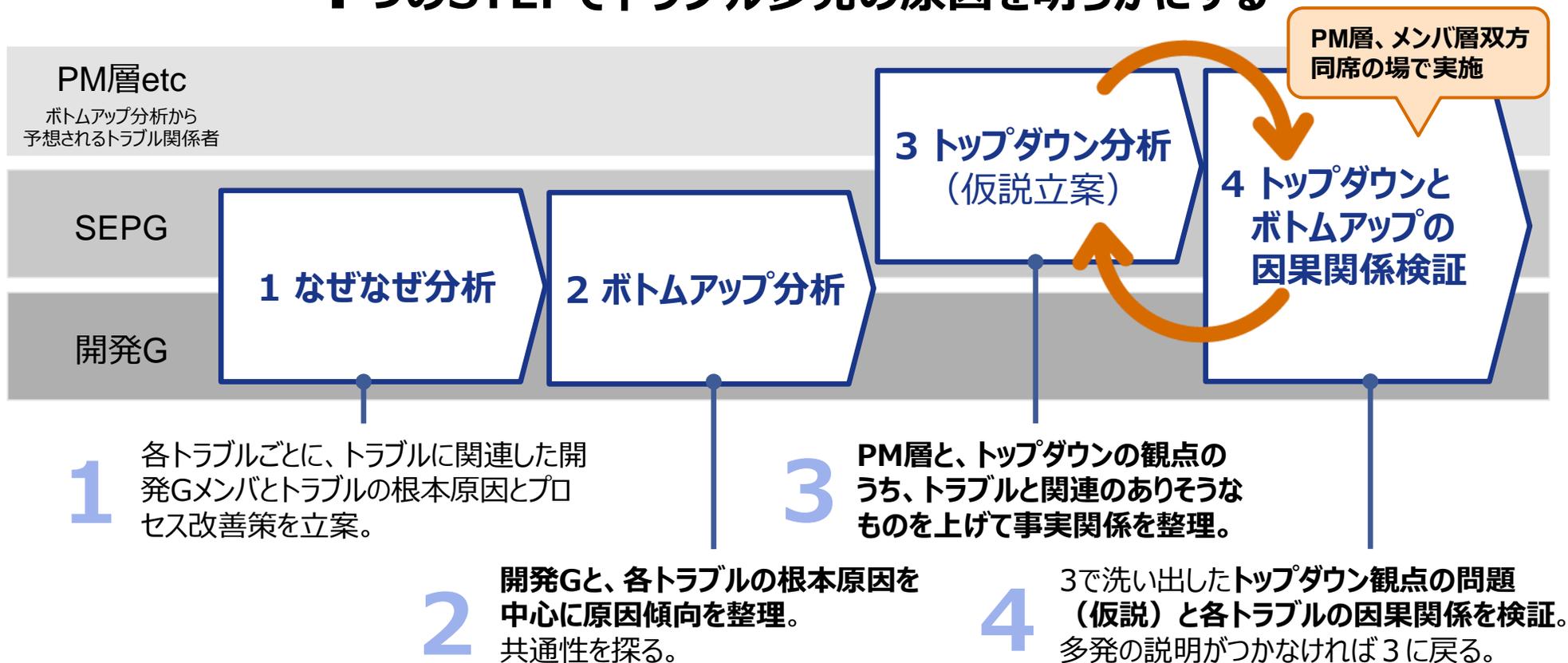
# トラブル多発の原因分析の進め方（SEPG中心に進める場合）

ポイント

例

実践

## 4つのSTEPでトラブル多発の原因を明らかにする



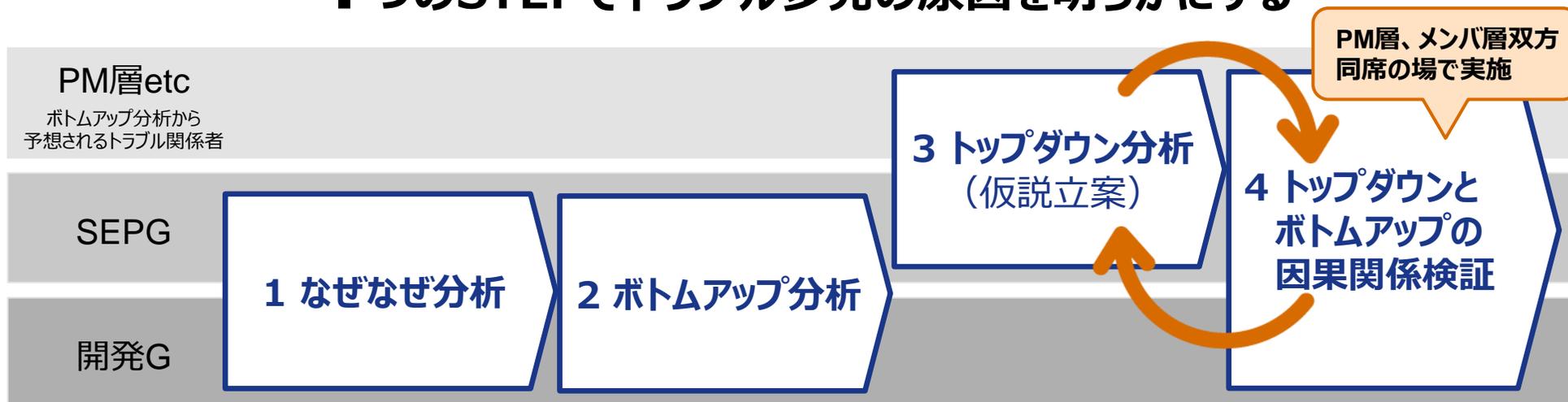
# トラブル多発の原因分析の進め方（SEPG中心に進める場合）

ポイント

例

実践

## 4つのSTEPでトラブル多発の原因を明らかにする



### ポイント

- **分析観点に応じた適切な相手にヒアリングする**  
関係者としていぐらいのヒアリングと議論を繰り返して、分析を進める
- **トップダウン分析→因果関係検証は仮説検証サイクルがおすすめ**  
(トップダウンはボトムアップと視点が大きく異なるためボトムアップからの積み上げは非効率になりがち)

### 注意点

初めて分析を行う場合は、SEPG→開発者へいきなりヒアリングを行って  
的外れな回答多数で暗中模索になりがち。

→**包括分析の目的、進め方を開発Gに説明し、理解してもらうのも大事**

# 一番大事なものは・・・

ポイント

例

実践

各トラブルのなぜなぜ分析は十分に実施したのに  
根本原因に共通性を見つけられない状況は  
心が折れそうになるかもしれません。

でも、トラブル多発を解消するためには

『 **これだけトラブルが多発しているからには、  
新しい問題が絶対あるから、  
何としてもそれを見つけるんだ！！！！** 』

という**諦めない気持ち**。そして、

『 **要因に対しての粘り強い検証** 』

が大事。



# トラブル多発時における再発防止管理の事例紹介

1. トラブル多発時の原因分析しくじり
2. 包括分析における原因分析 - ポイント -
3. 包括分析における原因分析 - ポイントを例とともに見ていく -
4. 包括分析における原因分析 - 進め方（SEPG主体） -
5. **包括分析のコストに関する考察**

# 包括分析のコストに関する考察

間接原因を包括分析となぜなぜ分析（※）で分析した場合のコストモデルを作成し、包括分析のコストを評価した。

※当PJのなぜなぜ分析は根本原因追及に特化しているが、ここでは間接原因まで分析対象を広げた場合を想定

## 前提

- ① 個々のトラブルの根本原因（プロセス問題）は判明済み
- ② 包括分析の対象トラブル数はn件、関連組織（開発/インフラ/統制系/PM/営業等）数はa
- ③ コストモデルに用いる係数（基礎工数）は筆者の分析経験に基づく肌感覚値



## 包括分析

※間接原因分析

### 分析の進め方

① 関連組織ごとの開発背景整理 → ② 間接原因とトラブルの因果関係の検証

### 基礎工数

① 1組織あたり10人時 ② トラブル1件あたり1人時程度

### コストモデル

分析工数（人時）  $\doteq 10 \times a + 1 \times n \dots (C1)$

## なぜなぜ分析

※間接原因分析

### 分析の進め方

トラブル1件ずつに対して関連組織ごとの網羅的なヒアリング

### 基礎工数

1トラブル1組織あたり2人時程度

### コストモデル

分析工数（人時）  $\doteq 2 \times a \times n \dots (C2)$

仮にn=20、a=3とすると、C1=50人時、C2=120人時 → **包括分析の方が低コスト。**

コスト構造として**aとnの関係がC1は和に対して、C2は積になる**点がポイント。

**aやnが大きい程C1とC2の差は広がり、包括分析のコストメリットが大きくなる。**

# 04

## まとめ

# 再発防止管理のポイントまとめ

## 1. 個別トラブルの なぜなぜ分析

## 2. トラブル多発時の 包括分析

目的	限られた期間・コストで 最大効果を生む対策の立案	<ul style="list-style-type: none"> <li>● トラブル「多発」の原因を究明する</li> <li>● 多発を防ぐために「最も費用対効果の高い」再発防止策を見つけること</li> </ul>
分析 ポイント	<ul style="list-style-type: none"> <li>● 作り込み、レビュー、テスト すり抜けごとになぜを繰り返 して根本原因を探る</li> <li>● 「人」以外の原因（ツール/ プロセス・仕組み/材料）に 着目してプロセス改善</li> <li>● トラブル分析会議、チケット 管理による実行管理徹底</li> </ul>	<ul style="list-style-type: none"> <li>● トラブル多発は、全く異なるレイヤーの複合要因で発生することが多い PM視点の問題 × 開発者視点の問題 × SEPG視点の問題etc</li> <li>● トラブル「多発」の原因分析には下記の3つすべてが必要。 A. ボトムアップ分析 = 各トラブルの根本原因の共通性を探る B. トップダウン分析 = トラブルとは別に、開発時の問題から間接原因を探る C. AとBの紐づけ</li> <li>● 分析観点に応じた適切な関係者との綿密な議論で仮説検証（BとC）</li> <li>● 一番大事なものは諦めない気持ちと要因に対しての粘り強い検証</li> </ul>
分析 タイミング	トラブル発生都度。 当PJでは100件以上/年。	トラブル多発によりQCD影響が出て、PJレベルの対応が必要な時。 当PJでは数年に1回程度。
分析主体	開発グループ	PM/SEPG/開発グループ
再発防止	局所的・短期・低コスト	広範囲（PJ）・中長期・高コスト
既存分析 （※）と比 較した場合 の特徴	<ol style="list-style-type: none"> <li>① 分析の精度と効率の両立</li> <li>② 分析観点の絞り込み</li> <li>③ プロセス問題追及に特化</li> </ol>	<ol style="list-style-type: none"> <li>① 分析の精度と効率の両立</li> <li>② トップダウン分析はPM目線</li> <li>③ トップダウン分析は仮説検証</li> <li>④ トップダウン分析は組織ごとの分析</li> </ol>

※ 既存分析例）VTA/いきさつダイアグラム/なぜなぜ分析等の経緯整理、原因分析手法

# 参考文献

- [1] 黒田勲、「安全性とヒューマンファクター」、日本信頼性学会編・信頼性ハンドブック、日科技連、pp.345-352、1997
- [2] 行待武生監修、ヒューマンエラー防止のヒューマンファクターズ、テクノシステム、2004
- [3] 小倉仁志、「なぜなぜ分析徹底活用術『なぜ?』から始まる職場の改善」、日本プラントメンテナンス協会、1997
- [4] 永田学/行待武生、再発防止対策における PSF 緩和の評価とヒューマンエラー未然防止への提言、ヒューマンファクターズ、Vol.9, No.2, 2005
- [5] 掛川悠/高橋秀行/市岡亜由美、自律的な品質改善のエンジン！大規模ミッションクリティカルシステムにおけるトラブル再発防止管理事例、JISA 会報 (JISA Quarterly) No.149, 2022
- [6] 全日本航空株式会社 総合安全推進委員会／事務局、「ヒューマンファクターズの実践的アプローチ」、全日本航空株式会社、1993

**NTT DATA**