

# ODC属性を軸とした欠陥分析パターンの 案出と分析作業の標準化



B1-3【経験発表】

エプソンアヴァシス株式会社

小島 義也

# 目次

1. SW欠陥分析作業の課題
2. ODC:直交欠陥分類
3. 活動の現状と当発表の位置づけ
4. 有効な欠陥分析軸の検討
5. 欠陥分析パターンの適用
6. 改善対策の検討
7. 実施結果（定量・定性）



# 1. SW欠陥分析作業の課題

# 現場でのSW欠陥分析作業の現状

- 分析作業の実施は、開発プロセスに組み込まれているが…
  - 具体的な作業手順になっていない
  - 作業内容も非定形であり、レベルがまちまち
- 基本的にリーダーの力量に依存している
  - 分析作業の標準化ができていない
  - 分析教育の実施がなく、OJTが中心

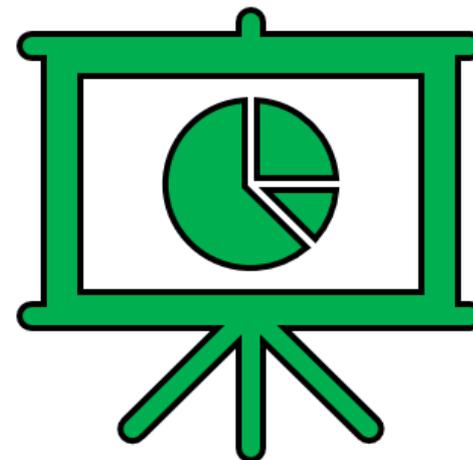




# SW欠陥分析の難しさ

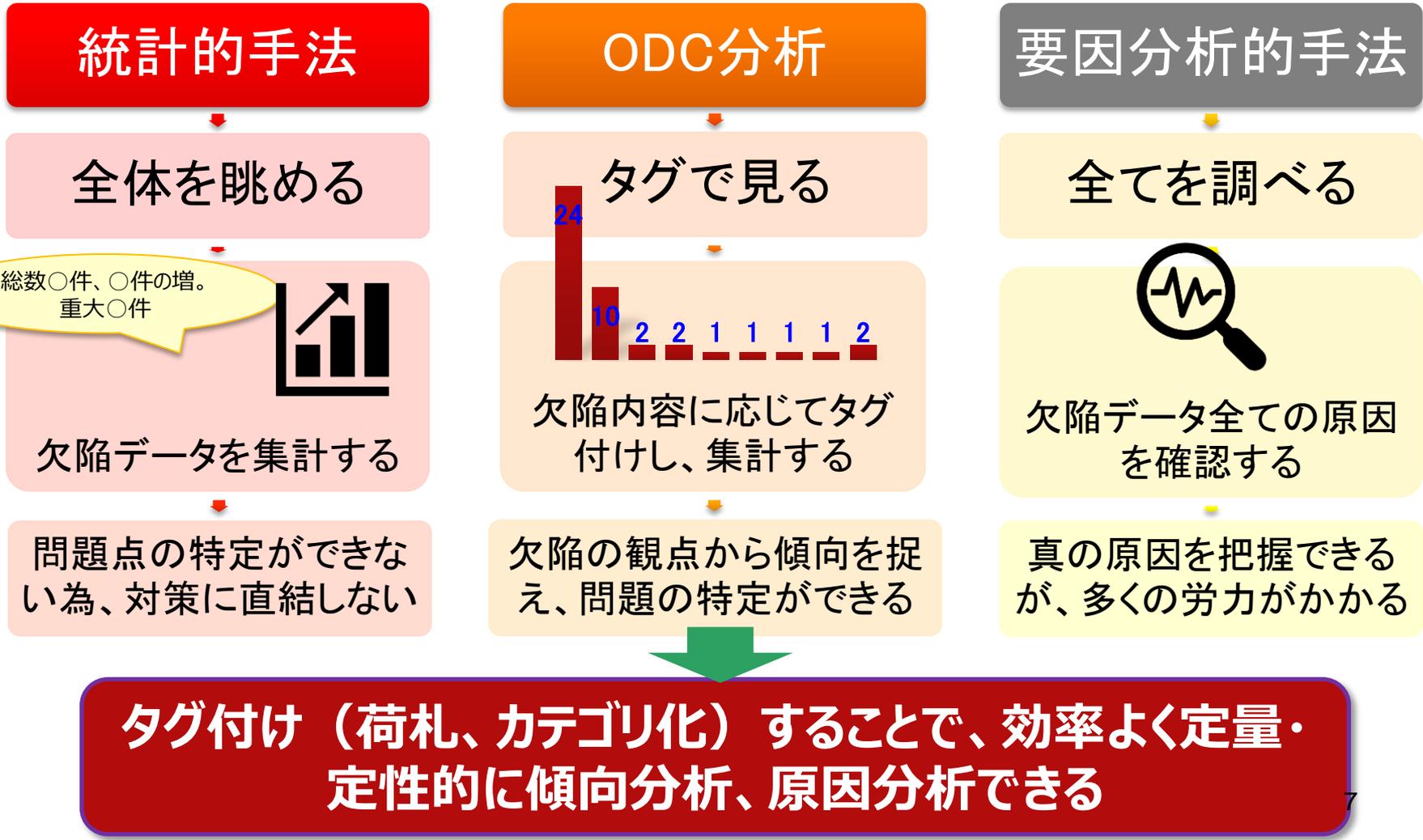
- 欠陥の傾向分析により、品質の弱点箇所を判断する
  - 全般的に同様な傾向がある場合、開発プロセスを分析
    - **開発プロセスに関する知識・知見が必要**
- 特定の箇所に集中している場合、該当する観点で分析する
  - 原因別分布、機能別分布、人別分布などで分析
    - **開発知識、機能要件、開発能力などの知識・知見が必要**
- 上記の知識・知見を持った開発者でないと、的確な分析結果がだせない。また、育成には時間がかかる
  - 欠陥を整理するために、**現象と原因の内容理解と適切な分類分けが必要!**
  - **スキル依存から脱却するためには、まず分析作業の標準化が必要!**

➤ 欠陥を整理するために、現象と原因の内容理解と適切な分類分けが必要!



## 2. ODC:直交欠陥分類

# ODC (Orthogonal Defect Classification, 直交欠陥分類) 分析とは



総数〇件、〇件の増。  
重大〇件



# ODCとは?

- 定量的分析と定性的分析の両方の性質を併せ持つ欠陥分析手法
  - **ソフトウェア開発に特化した欠陥分析する**際に必要な情報を一般化した分類の属性を提供
  - 欠陥の性質とその数量から次のアクションを考えるための技法
  - 互いに観点の異なる8つの属性に基づく欠陥分類で構成されている
  - 属性とその取りうるリスト値同士は重なり合わない→「直交」の由来

# ODC (Ver.5.2) の8つの属性

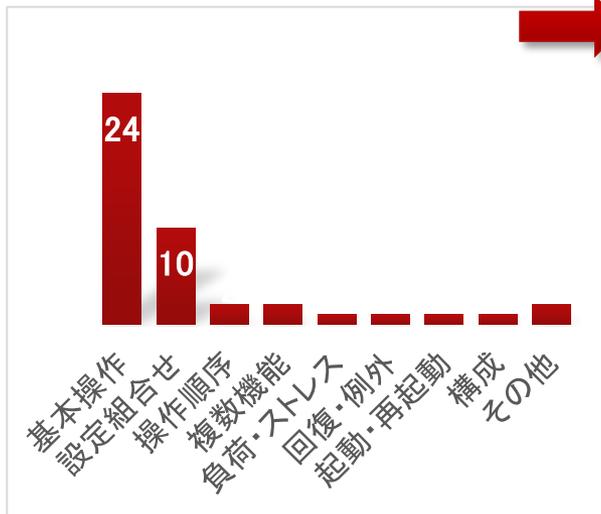
Attributes	属性*	概要
Defect Removal Activities	検出活動	欠陥を発見した活動
Trigger	トリガー	欠陥を発見する観点
Impact	インパクト(品質特性)	お客様に与える影響
Target	修正対象	修正対象の成果物
Defect Type	欠陥タイプ	修正した欠陥の種類
Qualifier	起因	欠陥埋め込みの種類
Age	混入時期	いつ作りこまれたか
Source	発生元	成果物の出所

※日本語名称は現在、ODC分析研究会で定義中。上記は暫定記。

# ODCを用いた欠陥分析

ODC属性で集計

欠陥傾向の分析



欠陥特性の導出

導出された傾向を  
深掘りする

背景や他の属性を  
含め原因を追求する



ODC属性別に欠陥数  
を集計する

欠陥の傾向を定量的に捉え、原因を定性  
的に追究する。

欠陥属性の分析によりバグの傾向を捉え、更に欠陥発生の原因追究をする

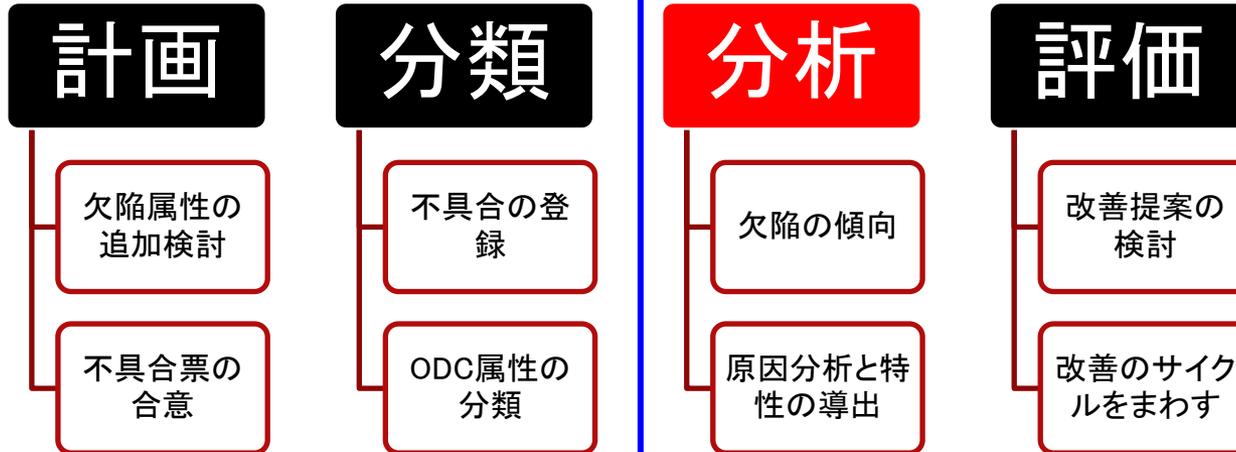


### 3.活動の現状と当発表の 位置づけ

# これまでの改善活動

これまで欠陥の分類が現場で実施できるように、ODCの導入を進めてきた。

ODC属性を付与する分類作業は経験と工夫が必要



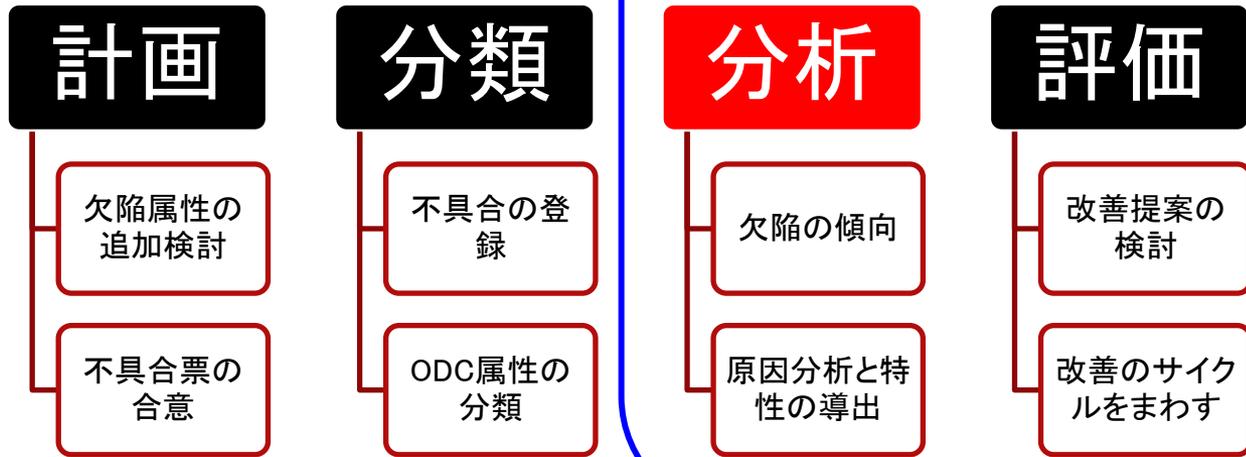
- 精度の高い欠陥分析をするためには、分類の当てはめが的確に行われる必要がある。
- 従って、分析を実施するには、ODC分類のスキルを現場に定着させることが重要。

小島 義也: ODC分析による設計品質改善の取り組み  
 (独立行政法人情報処理推進機構 (IPA)、2016)  
<http://www.ipa.go.jp/sec/reports/20170302.html#73>

# 当発表の位置づけ

当発表ではODC属性等で分類された欠陥群をどのように分析すると効率的かを検討した。

開発知識やプロセス改善に精通していないと分類結果の意味を読み解くのに躓く



高い精度の分類になっていれば、ソフトウェアの開発状況を正確に把握できます。

ODCによる分類付与は導入済みの現場となっていることが前提の説明となっています。

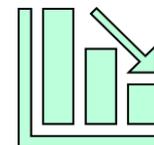
➤ スキル依存から脱却するため  
には、まず分析作業の標準化が  
必要!



## 4. 有効な欠陥分析軸の検討

# 過去の実践による欠陥分析パターン

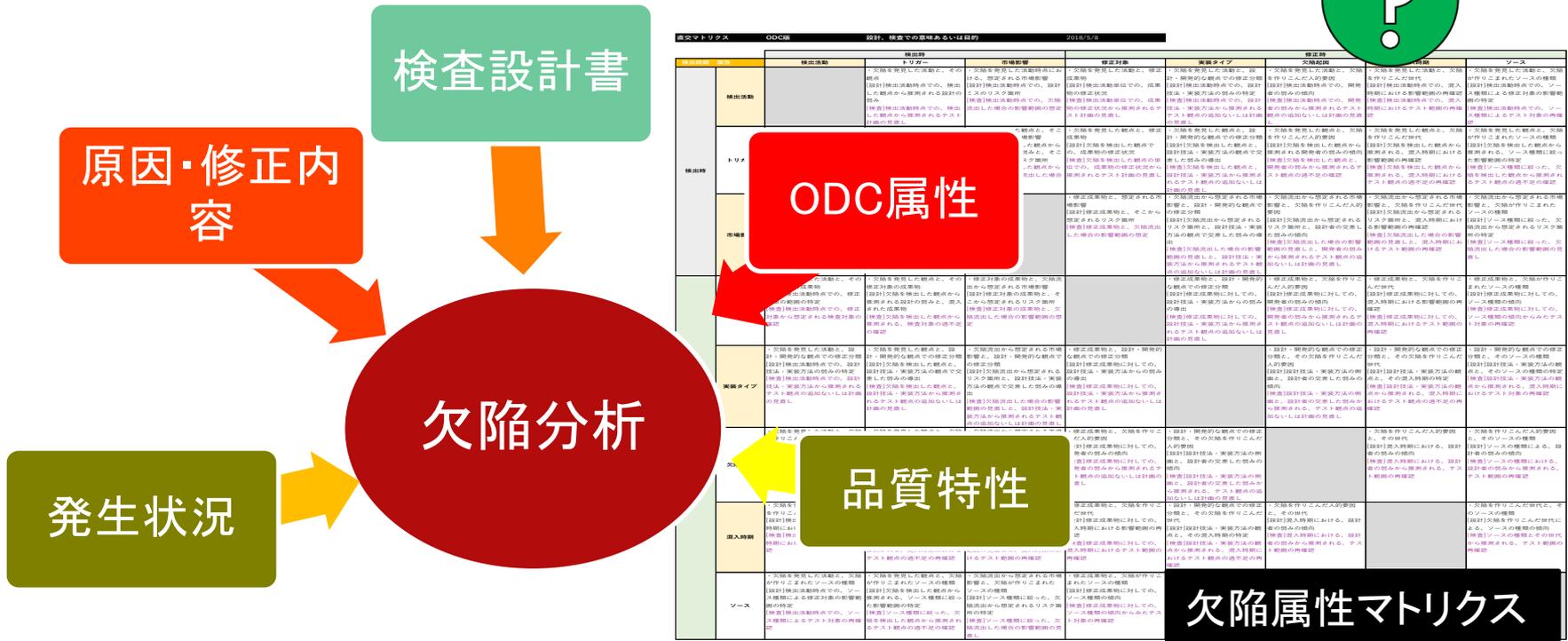
- 欠陥分析手順の標準化には、誰でも同じ方法で作業できることを目的として、実プロジェクト検証から導き出した知見をもとに、ODC属性を使用した観点で案出した欠陥分析パターンを、以下で説明する。



- 欠陥属性の組み合わせ
  - これまで実施してきたODCを導入しているプロジェクトの欠陥データを調査して、有効な属性を分析軸とすることを目的に、バグ票に記載されているデータ(欠陥属性等)の利用性の検証をすすめた。

# 多次元の因果関係のデータを基に分析 AVASYS

ODC8属性を利用した分析軸だけでも、その組み合わせを考えた場合、数多くの視点が存在する。  
その分析結果の選択には、知見・経験が必要となる。



分析結果が多すぎて、どこを見ればよいのかわからない

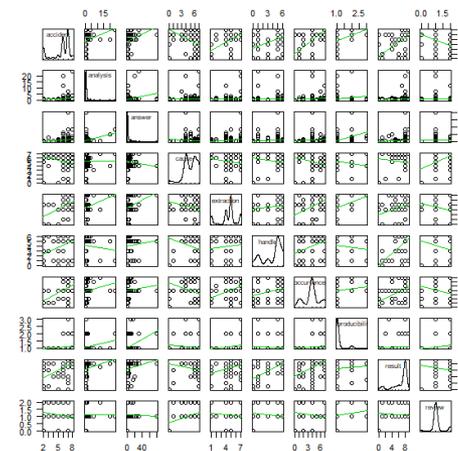
# 適用プロジェクトから導き出した知見→ 貢献度合い

※開発での欠陥分析に際し、どの属性がどれだけ有効であったかを検証した。

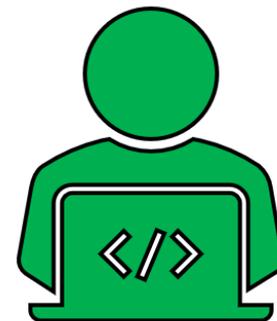
※表中の”◎”は一般的な傾向ではなく、プロジェクト適用の分析レポートおよび会議結果での活用度が高かったものを示している。

欠陥属性	設計チーム	検査チーム
時系列	○	○
トリガー	○	◎
インパクト(品質特性)	○	○
修正対象	○	×
欠陥タイプ	◎	○
起因	◎	○
発生種別	×	△
テスト項目有無	×	◎
....	...	...

※レポート等に記載された属性、属性同士の組み合わせを集計し、標準化を検討した。



※相関関係では、時系列や起因との組み合わせが関係性の強いことなども検討材料に入れた。

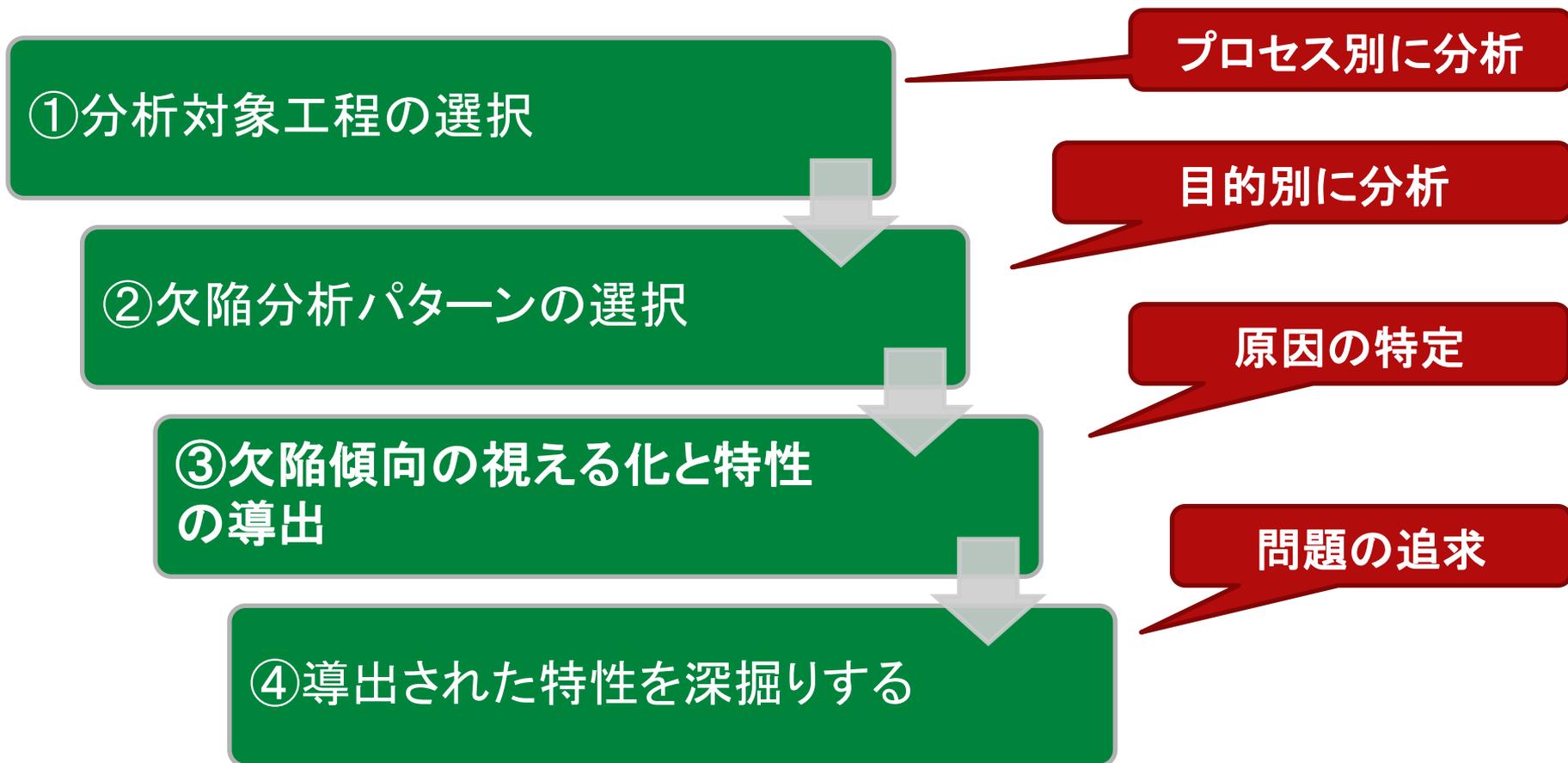


## 5.欠陥分析パターンの適用

# 欠陥分析パターンの案出

- ✓ まず、欠陥を分析するためには開発プロセスを選択対象とすることが有効と考えた後、
  - どのような欠陥傾向があるのか？
  - どういった問題があるのか？
- ✓ を誰でも同じ方法で探れることを目的として、実プロジェクト検証から案出した欠陥分析パターンの活用を説明します。

# 誰でも一定レベルの欠陥分析 ができるようにしたい



# ①分析対象工程の選択

- 分析対象とする工程は、大きく「検出工程」と「混入工程」に分けることが効果的と考えた。

A	検出工程	単体テスト 結合テスト システムテスト	検出活動のうち、テスト工程で検出された流出欠陥を分析する
B	混入工程	要求 設計 実装	混入工程別の欠陥を抽出し、設計欠陥を分析する



## ②欠陥分析パターンの選択

- 目的に応じて、分析パターンを選択する

No	目的	分析軸
A.検出工程(テスト工程)		
A-①	テストの <b>成熟度</b> を確認したいとき	トリガー+時系列
A-②	テストの <b>強化観点</b> を確認したいとき	トリガー+欠陥タイプ
A-③	テストの <b>完成度</b> を確認したいとき	トリガー+テスト項目有無
B.混入工程(設計工程)		
B-①	設計の共通課題を <b>人的要因</b> から確認したいとき	欠陥タイプ+起因
B-②	設計の共通課題を <b>設計技法</b> から確認したいとき	欠陥タイプ+トリガー
B-③	設計の共通課題を <b>網羅性</b> から確認したいとき	欠陥タイプ+品質特性


 例

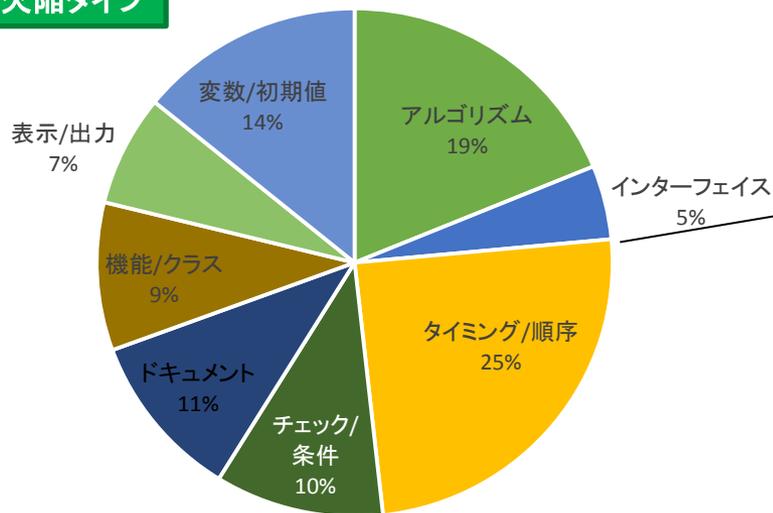
## B-② 設計の共通課題を設計技法から確認したいとき

設計技法と動作観点の関係に着目し、設計技法からみた設計の弱みを確認する(設計・実装スキルで共通した課題は何か?どこを改善すべきか?)を評価できる。

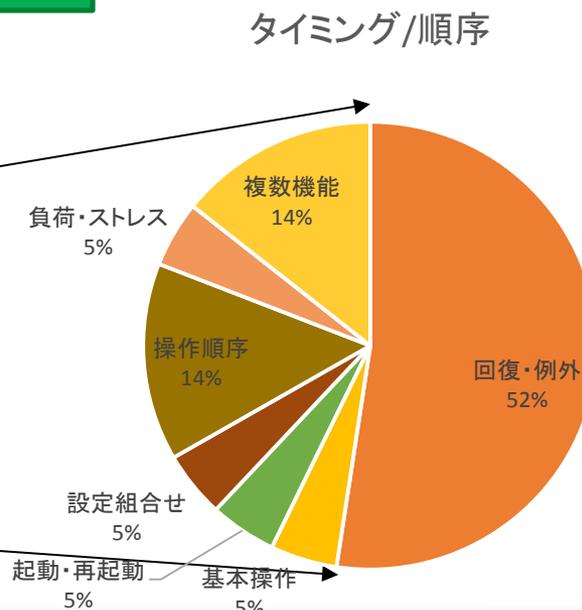
- ▶ 欠陥タイプ+トリガーで、**設計技法から掘り下げた**共通する設計の弱みを確認する。
- ▶ 例えば以下では、「タイミング/順序」は「回復/例外」による検出が多いことがわかる。
- ▶ 次開発ではこの「タイミング/順序」を設計する際に、「回復/例外」の動作観点に注意することを周知し、レビューでのチェックを実施することが対策案として挙げられる。
- ▶ 上記のように各設計技法における弱みを動作観点から推測し、設計改善へのフィードバックとすることが分析のメリットとして考えられる。

### 設計工程(混入工程)

#### 欠陥タイプ



#### トリガー



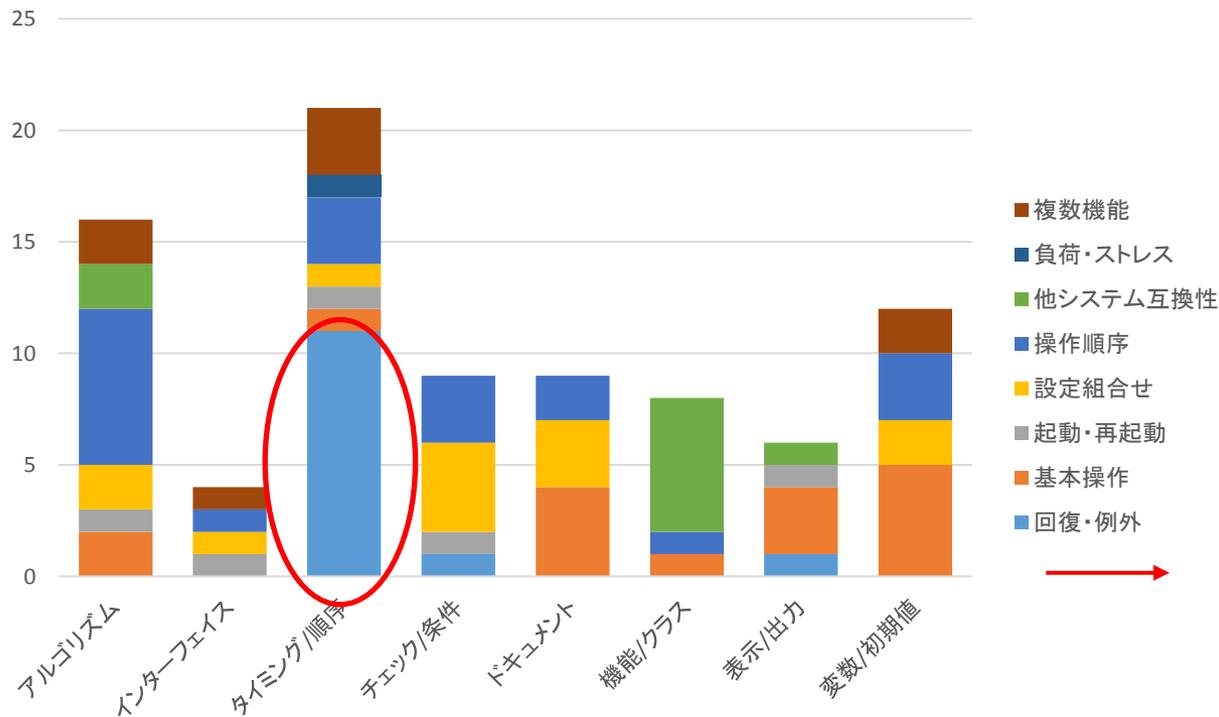
#### タイミング/順序

# ③欠陥傾向の見える化と特性の導出



詳細設計では「タイミング/順序」は「回復/例外」による検出が多いことがわかる。

設計工程(混入工程)



詳細設計のウィークポイントである「タイミング/順序」のミスを検出するには、「回復/例外」のレビュー観点が有効である。

# ④導出された特性を深掘りする



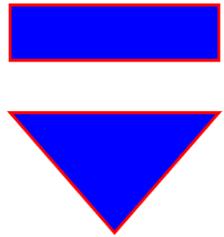
詳細設計

タイミング/順序

回復/例外

詳細設計では「タイミング/順序」は「回復/例外」による検出が多いことがわかる。

## 深掘りする





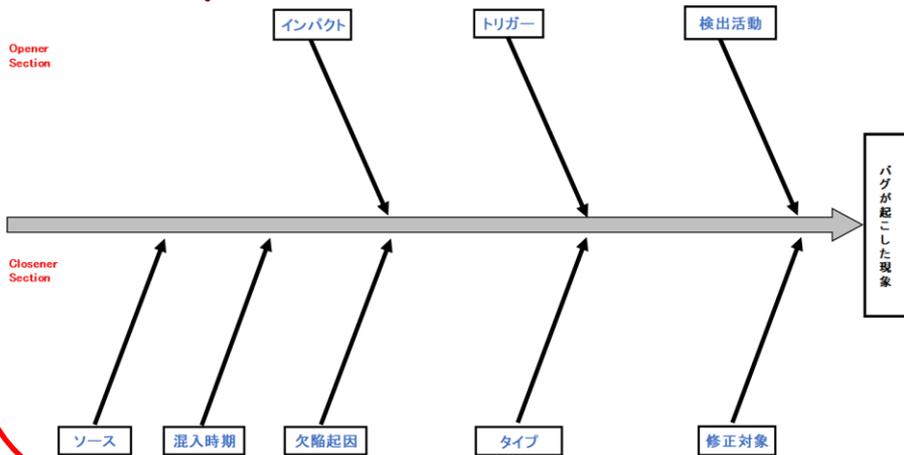
## 6.改善対策の検討

# 欠陥を取り巻くコンテキスト（背景）

欠陥は、要因と結果の関係が複雑に絡み合っている場合が多く、現象が起きた背景情報の全体把握が重要である。

いちから考えなくても、背景情報がすでに与えられている。

特性要因図



いちから背景情報を考えなくてはならない。時間がかかる。

なぜなぜ分析

バグの現象・起きた事象

背景要因①

背景要因①-1

背景要因①-2

背景要因②

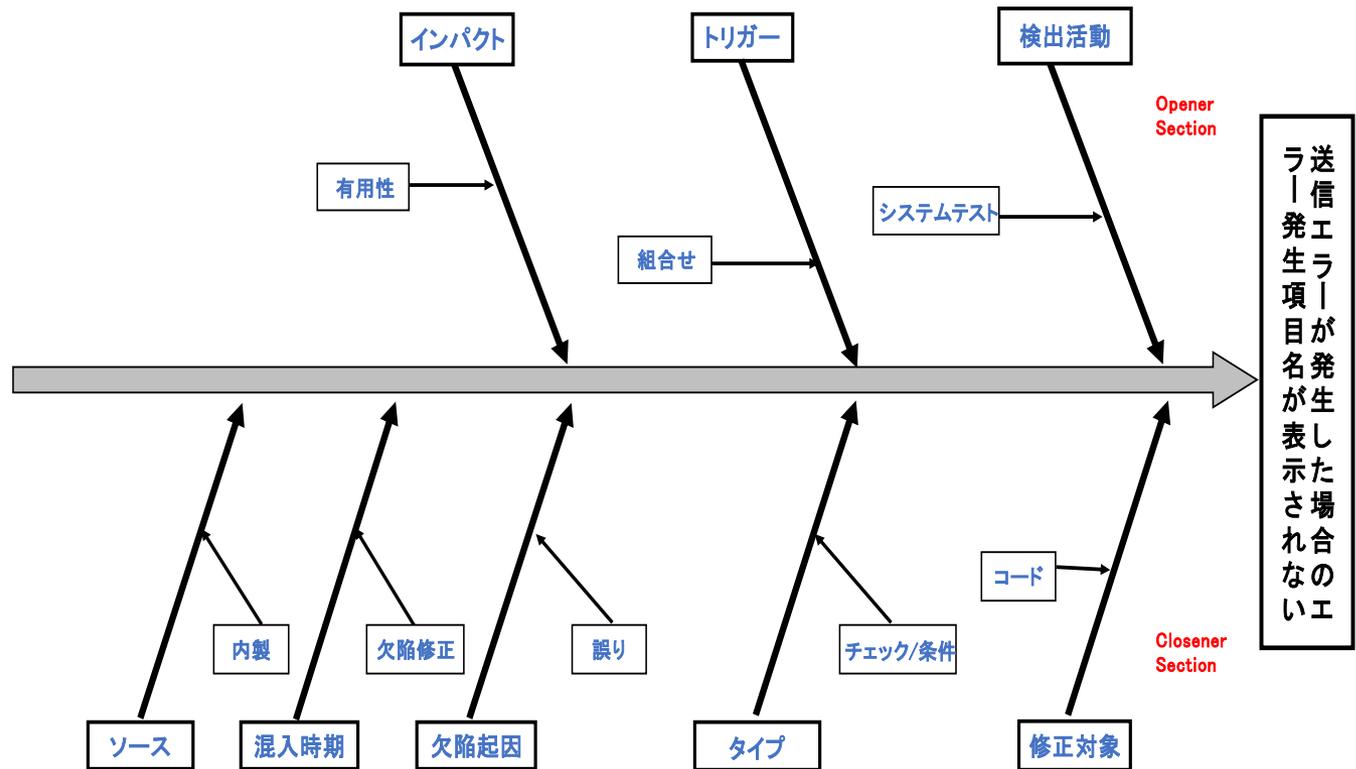
背景要因②-1

特性（結果）とそれに影響を及ぼすと考えられる要因（原因）との関係を系統的な図に示したもの。要因を大、中、小と整理し、特性と要因の関係を明確にすることで、原因を容易に見つけることができる。

# 交差要因図の作成

手順1:大骨はODC属性、中骨は属性の値を機械的に記載する

なぜなぜ分析にかける時間、工数を減らす策として期待ができる。また、標準化するためにODC属性を特性としているため、「交差要因図」と別称することとした。



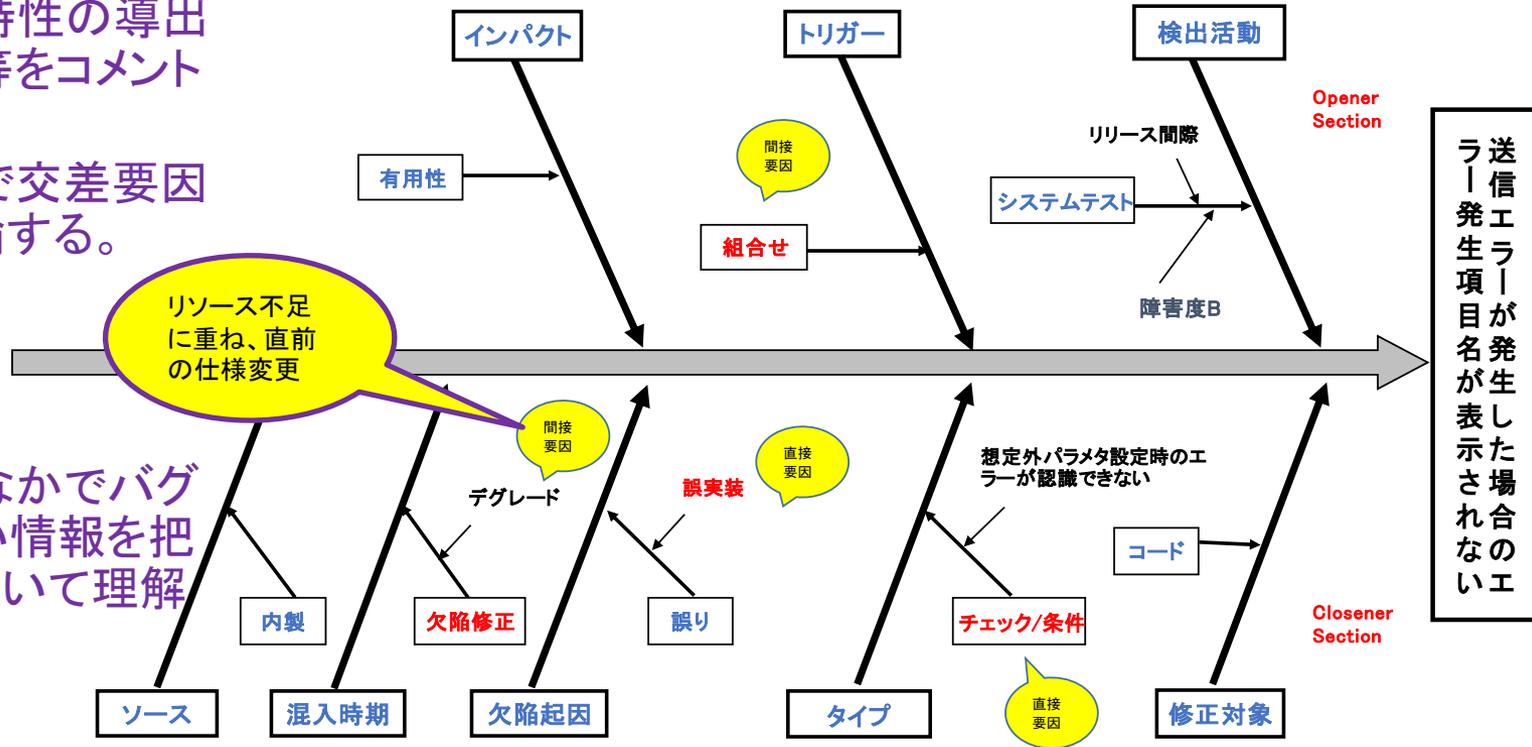
ODC属性を大骨・中骨に配置し、欠陥が生起した相互関係を明らかにできる。問題の構造を明かにしたり、要因・原因を深掘りしたり、欠陥発生全体像を見える化できる。

# バグ票にない「組織に起因する欠陥」要因を把握し、対策の有効性を高める

手順2:小骨に特性の導出で示した推論等をコメントする。

手順3:関係者で交差要因図を見て、議論する。

手順4:議論のなかでバグ票に出てこない情報を把握し、対策について理解を深めていく。



この背景知識(コンテキスト)を整理することで、欠陥の発生要因が関係者のなかで共有できることがわかった。

# 対策案の拾い上げ



- ✓ 交差要因図に記載された要因・原因に対し、対策案を立案します。
- ✓ 対策案は、バグの発生原因に対する対策案と、流出原因に対する対策案を検討してみます。
- ✓ 発生原因に対する対策案は、間違いや漏れを生み出した原因そのものを取り除くための改善を考えます。
- ✓ 流出原因に対する対策案は、間違いや漏れを次の工程に流さないための改善を考えます。
- ✓ 声大きい人の対策の決め打ち、といった**先入観による勘違い**を排除できます。
- ✓ バグ解決の方法論として定型的に蓄積でき、情報資産とすることができます。