品質を創造する
VERISERVE

日科技連ODC分析研究会向け講演資料

第4回活動報告(ODC分析基礎編)

特別講義「インパクト・ソース属性」とワークショップの活動報告

2020/10/1

株式会社ベリサーブ
森 龍二

自己紹介

VERISERVE

➢ 経歴

- ソフトウェアエンジニア：25年
  - 前半15年：開発者
  - 後半10年：レビュー組織リーダー
- 2016～：ベリサーブ
  - 検証サービスの品質保証活動

➢ ODCとの関わり合い

- 前職時代に親会社から技術導入
- ODC公開資料(IBM)を翻訳しネット公開
- いろんな講演に呼んでいただけるように
  - SQuBOKユーザ会(2013)
  - JaSST 2015
- 日科技連ODC分析研究会 運営委員

なぜこの講義を作ったか

VERISERVE

- ODCの中でもメジャーな「タイプ」「トリガー」と比較して、あまり活用されていない感のある「インパクト」「ソース」に光をあてる
- 欠陥分析の歴史をふりかえり、ODCにおける「インパクト」の意味をあらためて考え直してみる

目次

VERISERVE

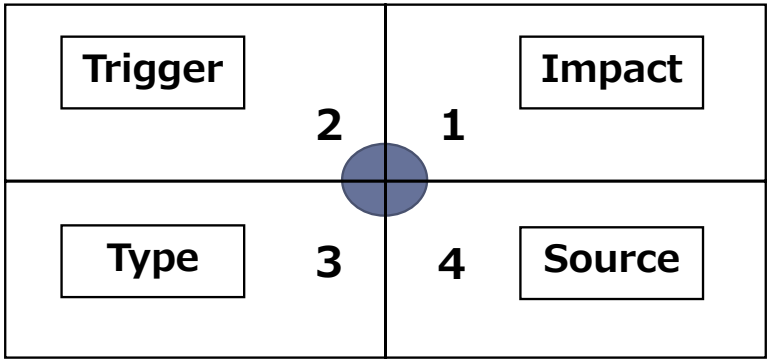
- **インパクト**
  - インパクトとは
  - インパクトの付け方と利用法
- **欠陥分類の歴史**
  - Beizer's Taxonomy(1983)
  - The Errors of TeX(1989)
  - IEEE 1044-2009
- **混入元(ソース)**
  - ソースとは
  - 混入元(ソース)の付け方と利用法
- **第4回ワークショップでの議論について**
- **まとめ**

➢ インパクト

あらためてODCとは

VERISERVE

- **ソフトウェアの「レントゲン」「CT」「MRI」**
- **以下の手順で行う**
  1. 4つの「基本属性」
     - トリガー
     - タイプ
     - インパクト
     - ソース
  2. 4つの情報の分布を見る
     - 製品全体の割合
     - 時間的変化
  3. 最終的に診断し、処方する



インパクトとは

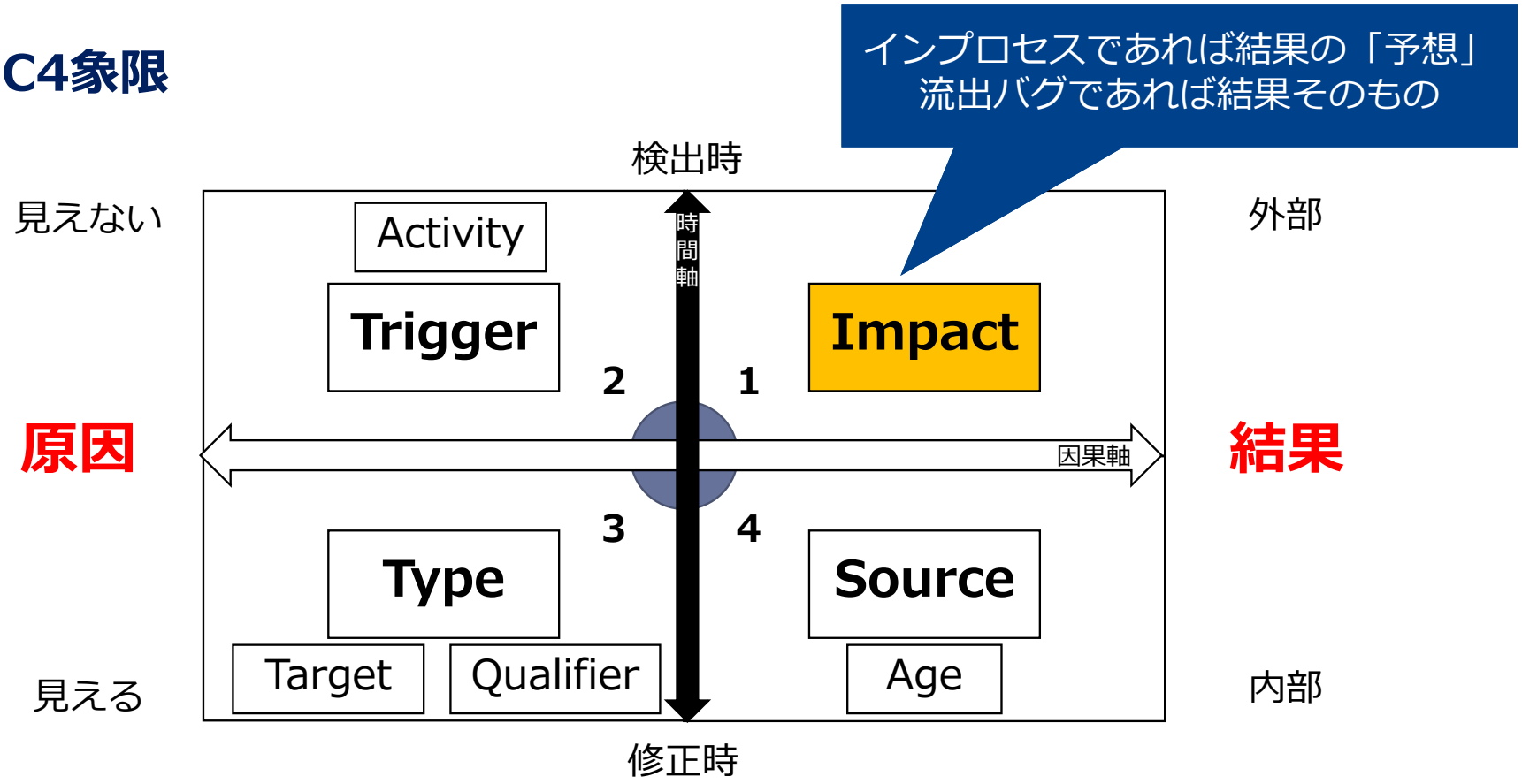
VERISERVE

- **V5.2の説明文**
  - For in-process defects, select the impact which you judge the defect **would have had upon the customer** if it had escaped to the field. For field reported defects, select the impact the failure had on the customer.
- **「プロセス内の欠陥であれば、もしそれが市場流出していたとしたらどういうインパクトを顧客に与えそうかを選べ。市場で報告された欠陥であれば、顧客に与えた障害のインパクトを選べ」**
- **注目ポイント**
  - 「顧客」とはだれか？
  - 市場流出前は被害を「予測」する必要があること
    - 現場が苦手とするところ

インパクトの位置づけ

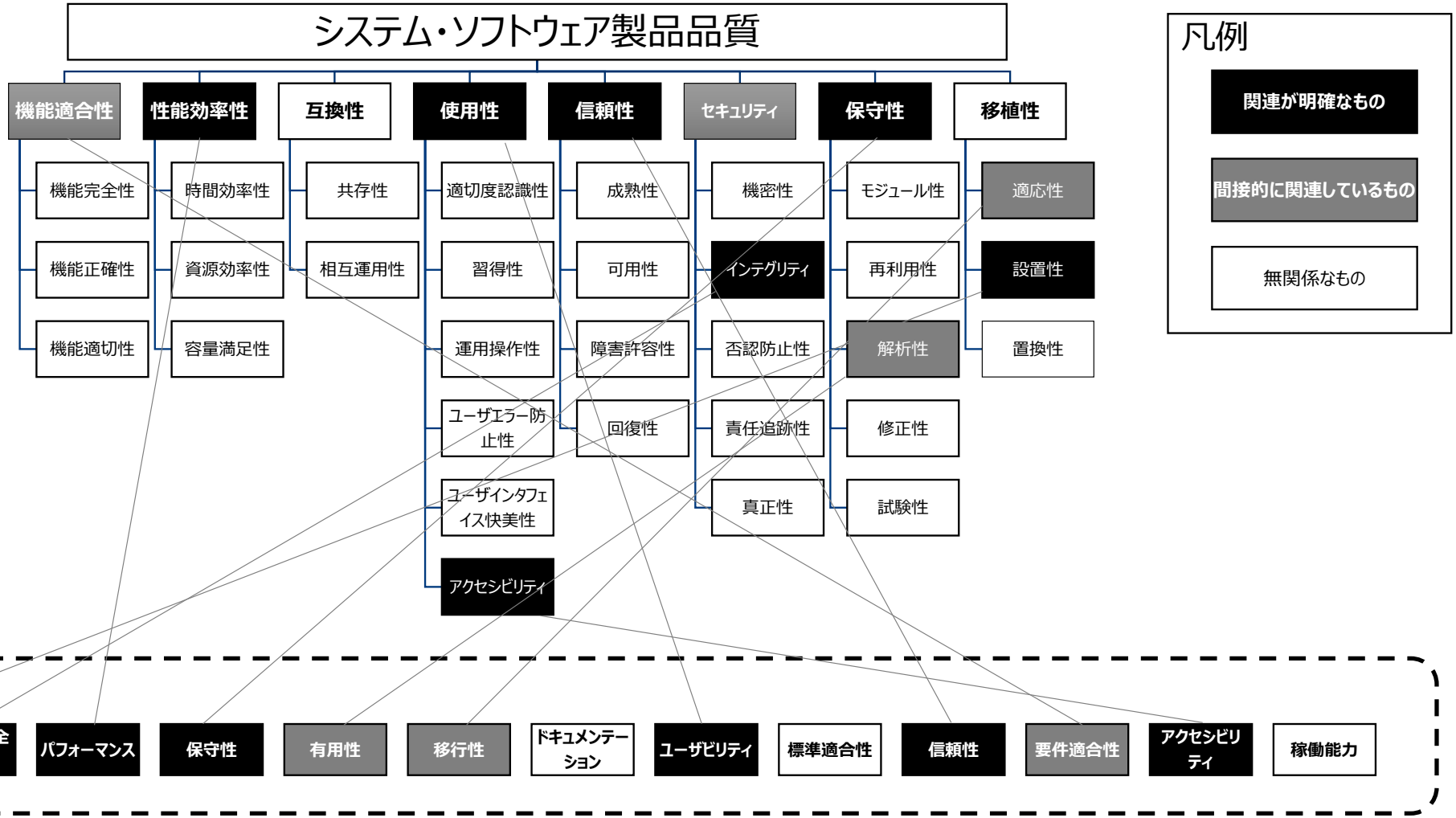
VERISERVE

➢ ODC4象限



品質特性とインパクトとの対応関係

VERISERVE



インパクト or 品質特性

VERISERVE

- **もともとはインパクト→品質特性**
  - 標準化の過程＋時間の経過により変化した
- **インパクトの代わりに品質特性を使うことの是非**
  - 品質特性は「正」の特性
    - 製品・システムの「良さ」を表す
  - インパクトは「負」の特性
    - 欠陥が積み上がることで「～性」が欠けていることを表す
  - 「稼働能力」「標準適合性」など該当するものがない属性がある
- **インパクトでしか表せないものがあるのでこのまま使うのがオススメ**
  - 補助として品質特性を使うのはあり

➢ 欠陥分類の歴史

Beizer's Taxonomy(1983)

VERISERVE

- 1xxx:機能不良(要求そのものの誤り)
- 2xxx:開発された機能不良(間違って実装された)
- 3xxx:構造不良(プログラムの作りが悪い)
- 4xxx:データ
- 5xxx:インプリメント(規約違反など)
- 6xxx:システム統合(インターフェイスに関わるもの)
- 7xxx:システム、ソフトウェア構成(OS,アーキテクチャに関する)
- 8xxx:テストの定義、実行に関わる不良
- 9xxx:その他不良記述されていないもの

【ポイント】
○要求/仕様そのものの欠陥と、実装欠陥を区別している
○テスト自体の欠陥が含まれている（ODCにはない）
×漏れと誤りは混同されている

The Errors of TeX(1989)

VERISERVE

- The Art of Computer Programmingで知られるクヌース博士がプログラム進化の事例研究として発表した1989年の論文
- 自身が手がける組版システム「TeX」の10年にわたる開発を通じたプログラムの進化と、開発中に得られた850項目のエラーログを分析し、15のカテゴリ分類を提示した

The Errors of TEX*

DONALD E. KNUTH
Computer Science Department, Stanford University, Stanford, California 94305, U.S.A.

クヌース15のカテゴリ

VERISERVE

- A.アルゴリズムの不首尾
- B.不手際または台無し
- C.一貫性または明快さのための美文書化
- D.データ構造の崩壊
- E.効率の増強
- F.忘れられた機能
- G.一般化または能力の成長
- I.対話性の改善
- L.言語の責任
- M.モジュール間の不適当な組み合わせ
- P.移植性の増進
- Q.品質の追求
- R.堅牢性の強化
- S.想定外のシナリオ
- T.ちょっとした誤植

【ポイント】
○要求・仕様・データの欠陥と実装の欠陥を区別している
×誤りと漏れの区別はない
×個人で作っているため、仕様書の概念やデータとプログラムの区別はない

IEEE 1044-2009

VERISERVE

- IEEE Standard Classification for Software Anomalies
- バグ票の項目とその属性値を規定したもの
- 主な属性

属性名	説明	ODCとの比較
Defect ID	欠陥ID	ー
Priority	優先度	ー
Severity	重大度	ー
Probability	発生頻度	ー
Effect	欠陥によって影響を受ける要求の分類	インパクトに相当
Type	欠陥が見つかった箇所の分類	欠陥タイプに相当
Mode	誤り、漏れ、余計	タイプ識別子に相当
Insertion Activity	欠陥「混入」活動	ー
Detection Activity	欠陥「検出」活動	欠陥検出活動に相当

欠陥分類の歴史まとめ

VERISERVE

- **Beizer's Taxonomy**
  - テストの誤りを分けている
  - 直交という概念はなく、各分類に混在している
- **Knuth's Errors of TeX**
  - 時間変化を見ると対策の優先順位を決めることができる
  - これも直交という概念はない
- **IEEE 1044-2009**
  - かなりODCに近くなっているがODCよりも粗い
- **他の分類法にないODC(1992)の特徴**
  - インパクトという概念はODC固有→品質の定義に沿ったもの
  - 漏れと誤りを分けたこと
  - きめ細やか(逆につけにくくはなっているが)

私が考えるインパクトの意味

VERISERVE

- **データの寿命問題**
  - 重大度/優先度はコンテキスト/ステークホルダにより変化する
  - データの寿命を考えたとき何を残すか
    - ユーザが受けるインパクトの「性質」を残す
- **重大度・優先度が共有されていない場合**
  - 例えば第三者検証会社が報告する場合など
- **層別のパラメータとして**
  - パレート図でどんぐりの背比べを避けるため
  - 1，2位が決まれば対策の優先度をつけやすい

➢ インパクトの付け方と利用法

インパクトの付け方

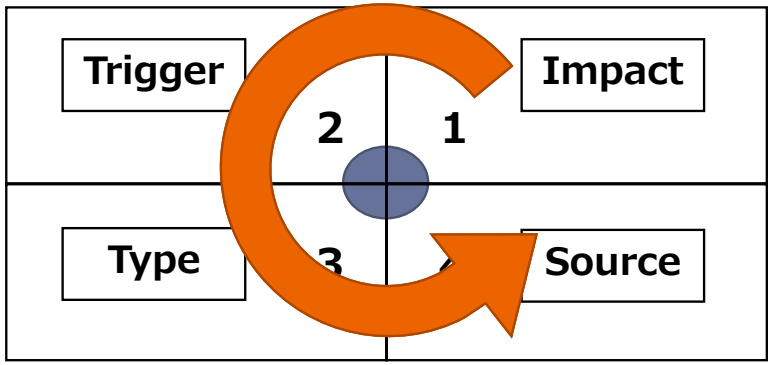
VERISERVE

- **「だれ」に対するインパクトかを最初に明確にする**
  - 製品に対するエンドユーザが明確に定義できない場合
    - 最終的に組み立てられる製品・システム上で考える
      - 例：車載ECU→ドライバー、決済システムの認証部分→ペイメントユーザ
- **市場での使い方を把握する**
  - ヒアリングしたり、実際に見にいく
- **判定基準を明確にする**
  - 報告先、分類チームメンバ同士で話し合い、合意する
- **実際につけてみる**
  - つけながら判定基準にフィードバックする

インパクトの利用法

VERISERVE

- **分析の「インプット」の層別として使う**
  - 全体傾向も知りたいが分析結果をすばやく出したい
  - 入力データの層別キーとして使う
    - 例えば信頼性が求められる製品なら「信頼性」に限定する
- **次にトリガー＋αでV字モデルの右側(テスト工程)を検証する**
- **次にタイプ＋αでV字モデルの左側(設計実装工程)を検証する**
- **最後に施策の向かう先としてソースを使う**



➢ 混入元（ソース）

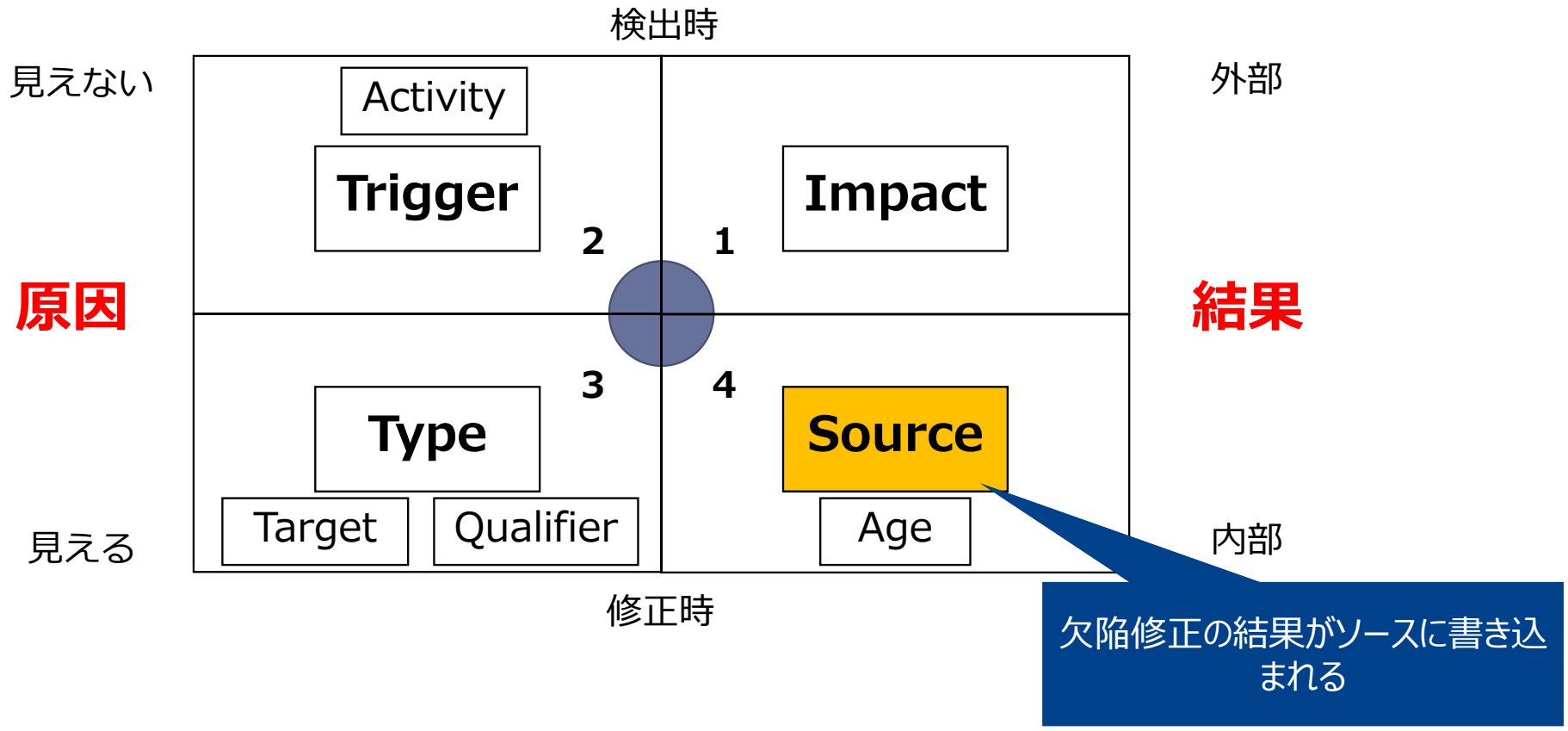
混入元(ソース)とは

VERISERVE

- **V5.2の説明文**
  - Choose the selection which best defines the fixed Requirements/Design/Code in terms of its **developmental history**.
- **「修正した要件定義書/設計書/ソースコードを、その開発の歴史の観点から、最もよく表すものを選べ」**

混入元(ソース)の位置づけ

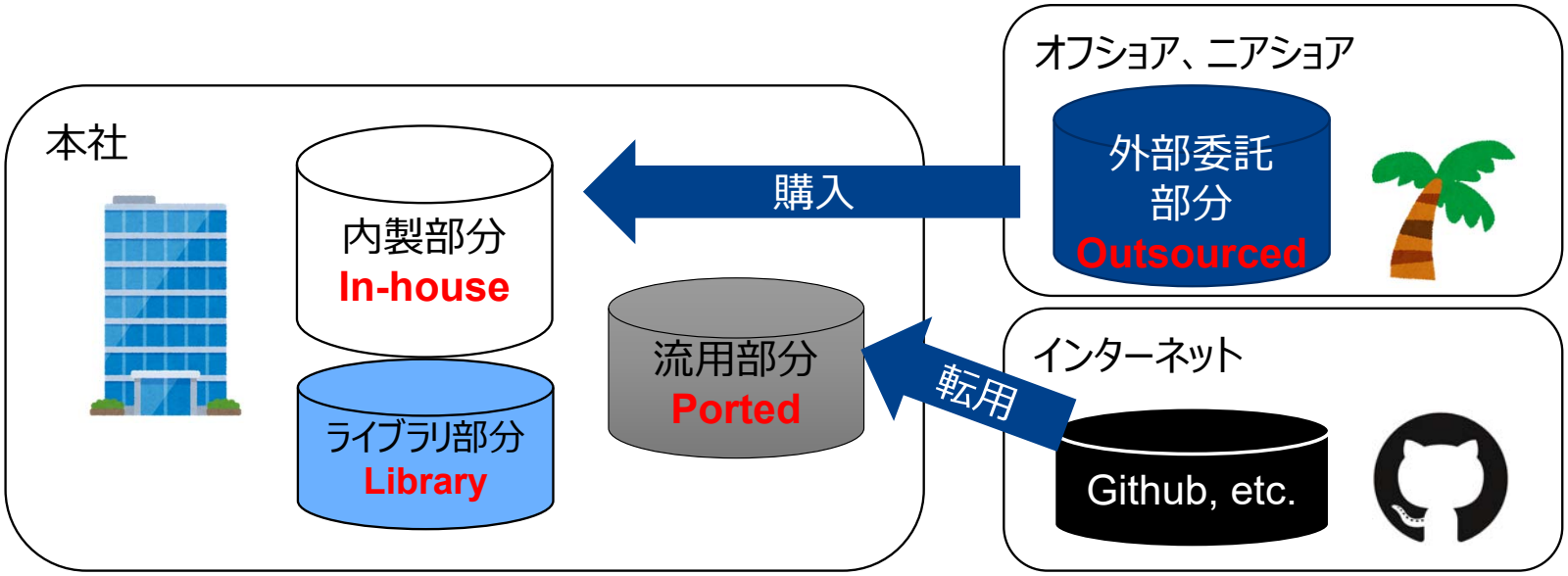
VERISERVE



混入元(ソース)

VERISERVE

- 「だれ」が混入させたかをチームレベルまで薄めたもの
- 犯人捜しにならないように注意が必要
- 現場の状況に応じてカスタマイズするのが普通
  - 例：開発チーム・オフショア先が複数拠点、複数企業である、など

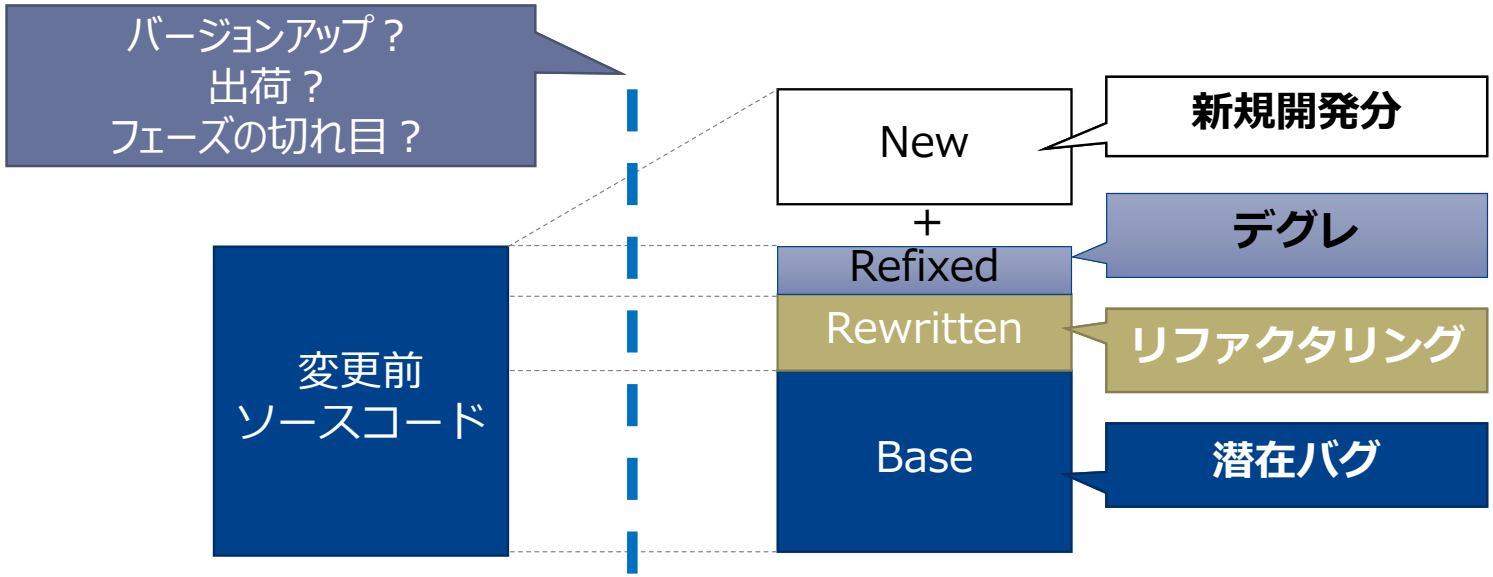


➢ 混入時期(Age)について

混入時期(Age)

VERISERVE

- **欠陥を「いつ」埋め込んだか**
- **V5.2では「ある時期」からの変更の種類を記録する**
- **「ある時期」は明確には決まっていない**
  - いつからかは決める必要がある



➢ 混入元(ソース)の付け方と利用法

混入元(ソース)の付け方

VERISERVE

➢ 属性値について合意を得る

- 目的：何のために属性をつけ, 何を見たいか
  - ベンダーごとの偏りはないか
  - 受け入れチェックが十分か

➢ 必要なカスタマイズの検討

- 属性値を現場に合わせる
  - 例：内製→X社(日本), X社(中国)、アウトソース→A社, B社
- 非ODC属性と組み合わせる
  - 混入元：標準ODCのまま(内製、アウトソース…)
  - 「開発元」：X社(日本), X社(中国), A社, B社

混入元(ソース)の利用法

VERISERVE

- **改善の向かう先として使うのが本来の使い方**
  - 裏を返すと犯人捜しですが、最初にソースから分解しないというのがミソ
  - 信頼性に関わる欠陥のタイプは代入/初期化が多い
    - →静的解析やコードレビューが足りていない
      - →A社担当部分から始め、受け入れ基準とするなど
- **最初にソースから見てしまうと犯人捜しになる**
  - A社は200件/500件→A社は次から使わない(犯人捜し)

➢ 第4回ワークショップでの議論について

インパクト・ソースへの質問と回答

VERISERVE

インパクトについて

- インパクトの使い方
  - 他の属性を層別するためにある
- 品質特性を引用した理由
  - 外部品質・利用時品質と関係があるから
- インパクトにODC固有属性がある理由
  - 品質特性では表せない「負」の概念を表現
- インパクト分類における「想定」の扱い
  - ターゲットとなるユーザを定義
- ドキュメンテーションの意味
  - インパクトはユーザ提供文書に用いる

ソースについて

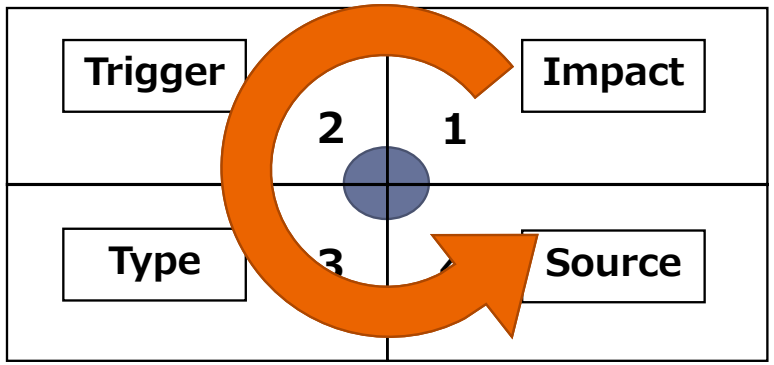
- カスタマイズ方法
  - 見たいこと、知りたいことで属性値を変える
- 犯人捜しにならない工夫
  - ポジティブな使い方をすること
- 流用とライブラリの違い
  - ソースコードを持ってきたかどうかで分ける

➢ まとめ

まとめ

VERISERVE

- **インパクトはODCに特徴的な属性**
  - 分析のインプットの層別に使うと効果的
- **ソースは「だれ」をチーム単位に薄めたもの**
  - 改善施策の適用先を絞るのに効果的
- **講義で説明しきれなかった疑問について研究会で議論し回答**



参考文献

VERISERVE

- 「Orthogonal Defect Classification/Orthogonal Delta Categorization」, Ram Chillarege, 2019
- Orthogonal Defect Classification v 5.2 for Software Design and Code, IBM, 2013
- Orthogonal Defect Classification v 5.2 Extensions for Defects in GUI, User Documentation, Build and Natural Language Support(NLS), IBM, 2013
- ODC属性 日本語翻訳原稿 Opener Section, 日科技連ODC研究会適訳化チーム, 2019
- ODC属性 日本語翻訳原稿 Closer Section, 日科技連ODC研究会適訳化チーム, 2019
- 「これからの“バグ”の話をしよう」、細川宣啓、JaSST'13 Kyushu、2013
- 「ソフトウェアテスト技法」、ボーリスバイザー、日経BP、1994
- The Errors of TeX, DONALD E. KNUTH, SOFTWARE-PRACTICE AND EXPERIENCE, VOL. 19(7), 607-685 (JULY 1989)
- IEEE1044-2009: IEEE Standard Classification for Software Anomalies, IEEE Computer Society, 2009
- 「インパクトとソース」、日科技連ODC分析研究会、2019