

イテレーティブなプロセスと欠陥モデル による要因分析法 (Root Cause Analysis) の改善

アジャイルRCAの提案



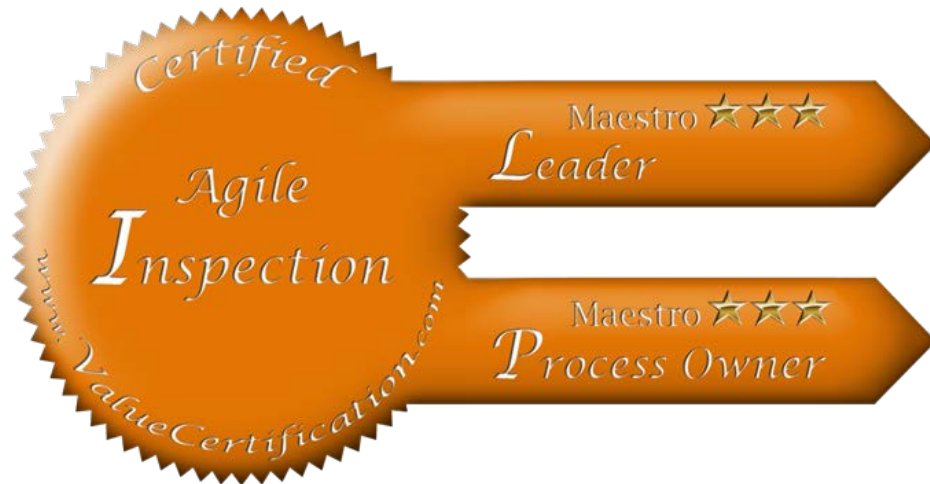
永田 敦
ソニー株式会社

自己紹介

ソニー株式会社
品質エンジニアリングマネージャ
IP&S PSG 品質保証部門
ソフトウェア開発品質改善

SQIP 研究会第7分科会副主査
派生開発推進協議会運営委員

JSTQB Advanced Level Test Manager



謝辞

**真のアジャイル開発とその改善プロセスおよび
Agile Inspectionを教えていただいた
Tom Gilb 氏と Kai Gilb氏に感謝いたします。**

RCA : Root Cause Analysis

- **Root Cause Analysis (RCA) は、品質改善に非常に有効な情報を与えてくれるツールである.**
- **設計者に対して**
 - **バグを埋め込まない**
 - **プロセスを改善する**
- **テストエンジニア**
 - **Error Prone : エラーのありかと傾向**
 - **テスト戦略とテスト設計.**
- **これまでは、なぜなぜ分析を使っていた**

なぜなぜ分析テンプレート

[illegible]

なぜなぜ分析の問題点

1. 時間がかかる

2. 人を攻めてしまい、分析がうまくいかない.

3. 分析のスキルが向上しない

1. 時間がかかる

- 準備

- 説明資料を担当設計者に作らせる

2時間以上

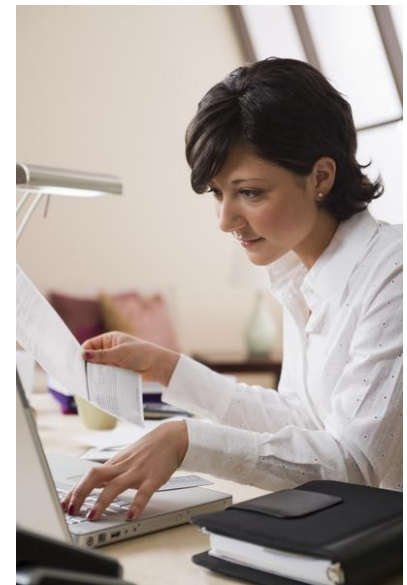
- 分析ミーティング

- なぜなぜ分析のテンプレートを使いながらレビューする

2 時間

- 効率

- 1回のミーティングでは終わらない。
もう一回：追加資料の依頼



<http://info.legalzoom.com/documents-required-preparation-4187.html>

結局トータルでおよそ8時間以上かかっていた

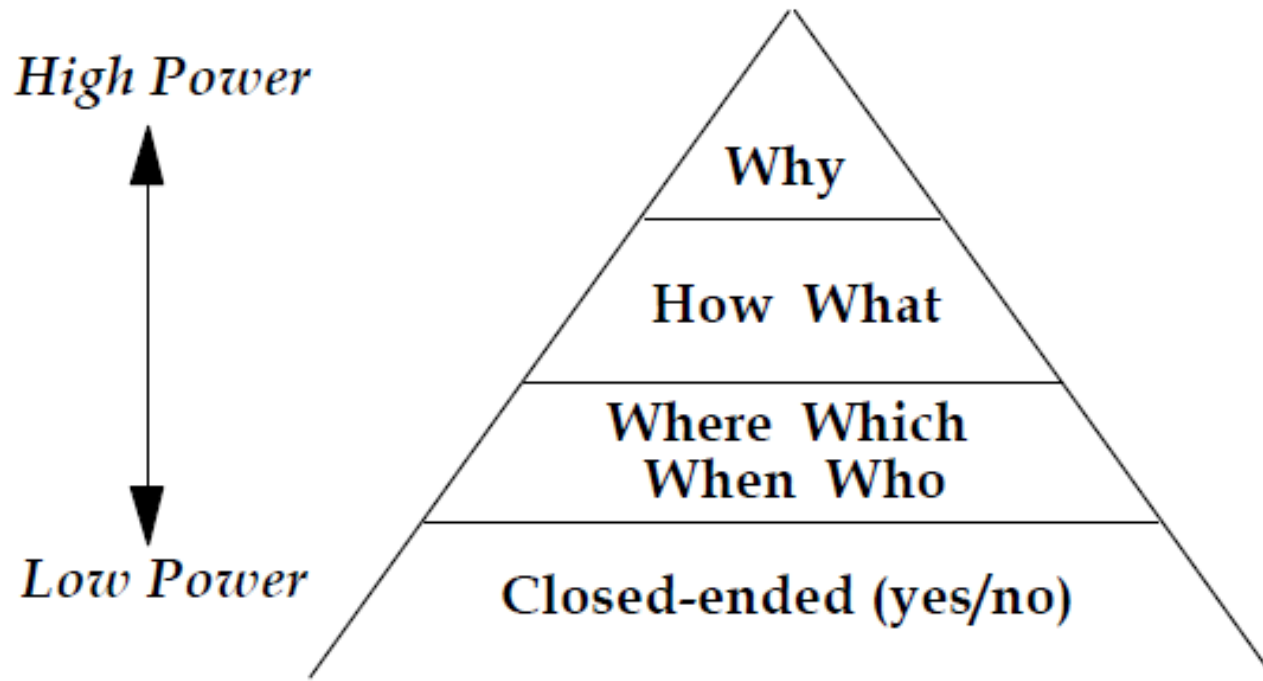
2. ストレスを感じる分析

- 「なぜ」は最強の質問
過ちを犯した人を**攻撃**してしまう
たとえ攻撃するつもりはなくても



2. ストレスを感じる分析

- 「なぜ」は最強の質問
過ちを犯した人を**攻撃**してしまう
たとえ攻撃するつもりはなくても



The Art and Architecture of Powerful Questions, Eric E Vogt, Juanita Brown and David Isaacs, 2003

2. ストレスを感じる分析

- 「なぜ」は最強の質問
過ちを犯した人を**攻撃**してしまう
たとえ攻撃するつもりはなくても
たとえば、部長から

なぜこのようにしたの？

- どう感じます？





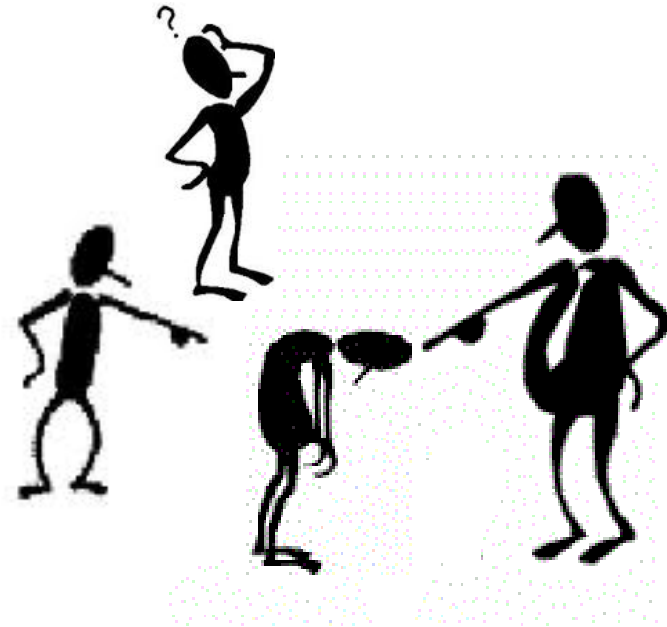
<https://www.flickr.com/photos/ruthhb/2917888819/in/photostream/>

「なぜ」：最強の質問

TPSオリジナル：製造ライン ソフトウェア開発現場



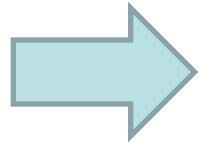
欠陥：物理的な問題
対象：現場のモノ



欠陥：バグ
対象：エンジニア

2. ストレスを感じる分析

- 「なぜ」は最強の質問
過ちを犯した人を**攻撃**してしまう
- 詰問に対する反応,

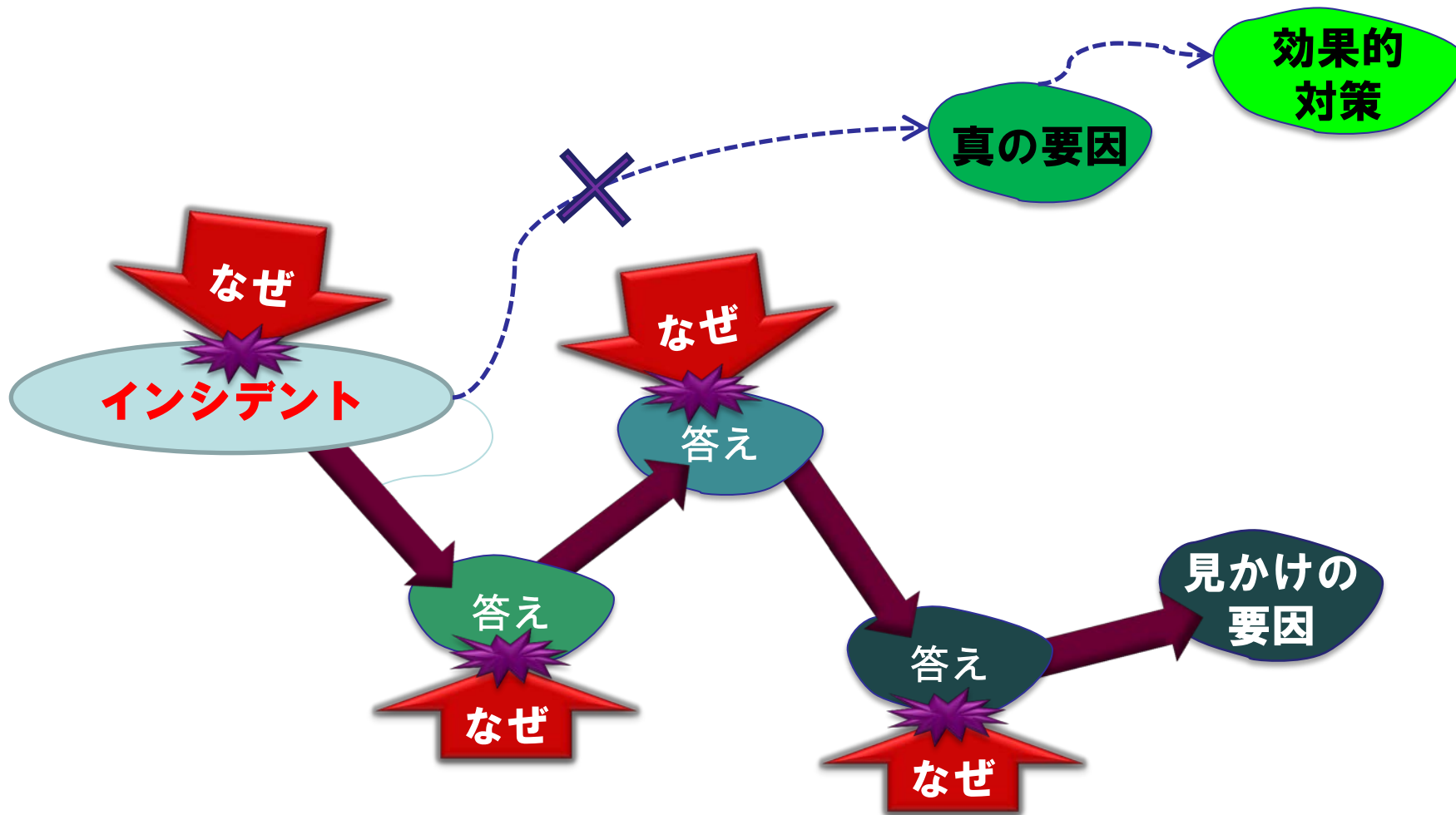


[防御モード]

人は、真因を探ることよりも、
自分の答えを正当化しようとする
Eric E. Vogt



防御モードでのなぜなぜ分析



適切な質問

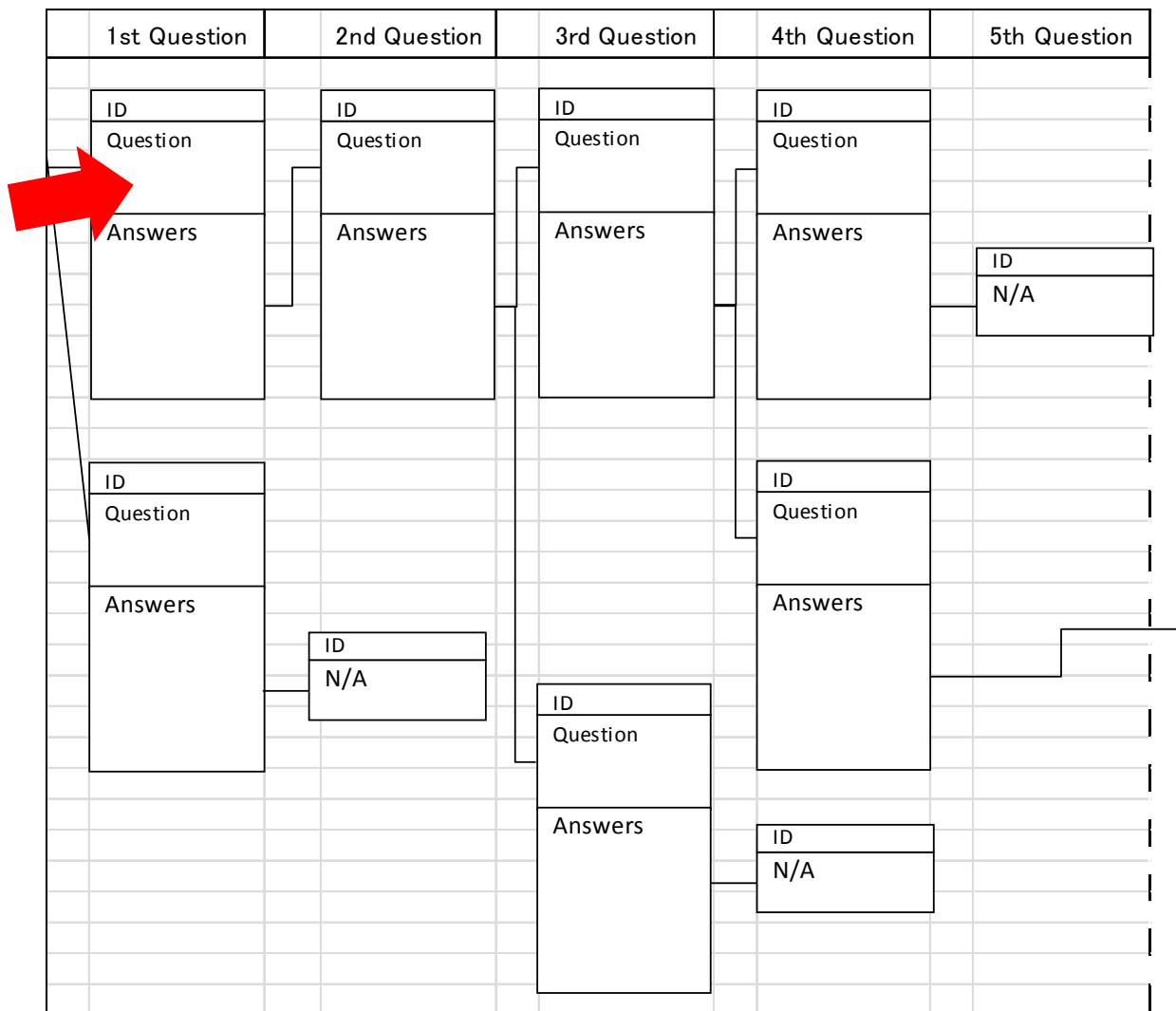
強力な質問



適切な質問

では、どんなものが適切な質問か

改善 1 : 質問を明示的に表現する



3. 分析のスキルが向上しない

- **時間がかかる**

なぜなぜ分析を何回もすることができない



分析やファシリテーションのスキルが上がらない



- **適切な質問を考えだすことができない**

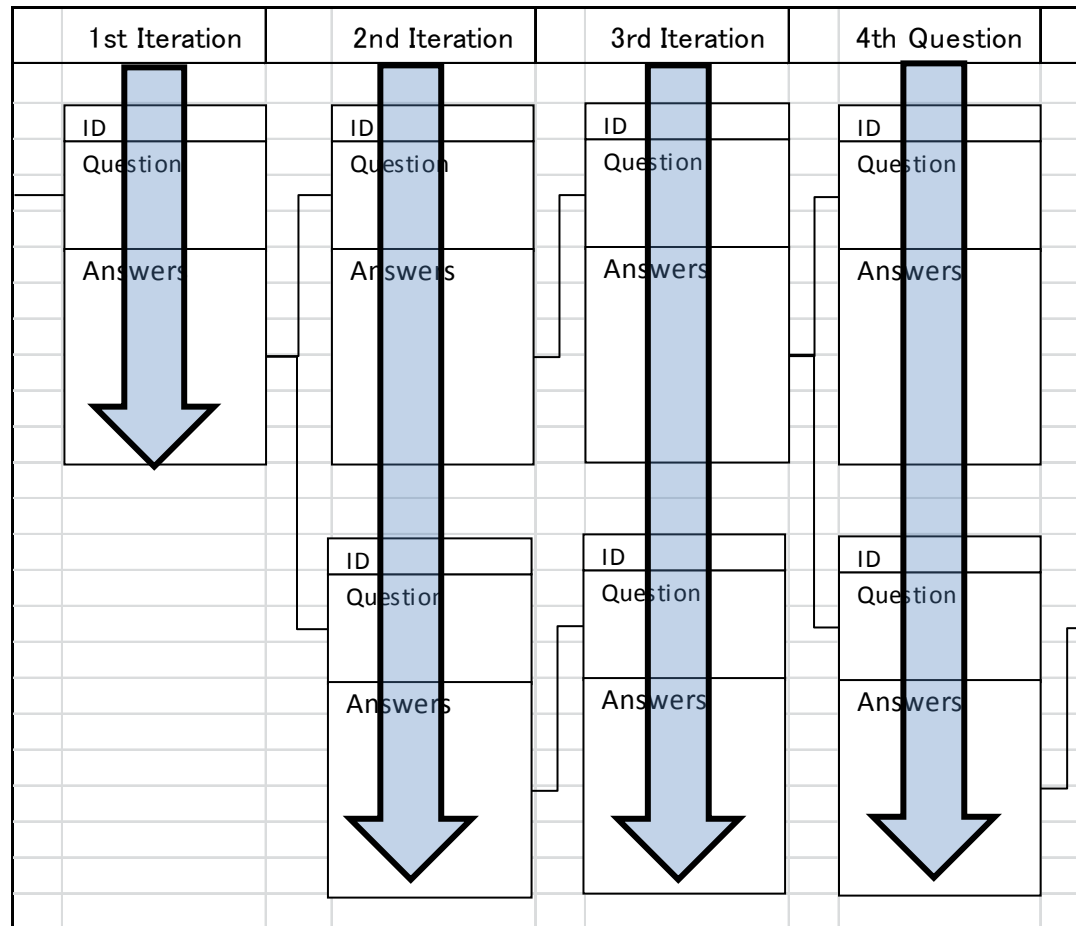


真の要因にたどり着かない。

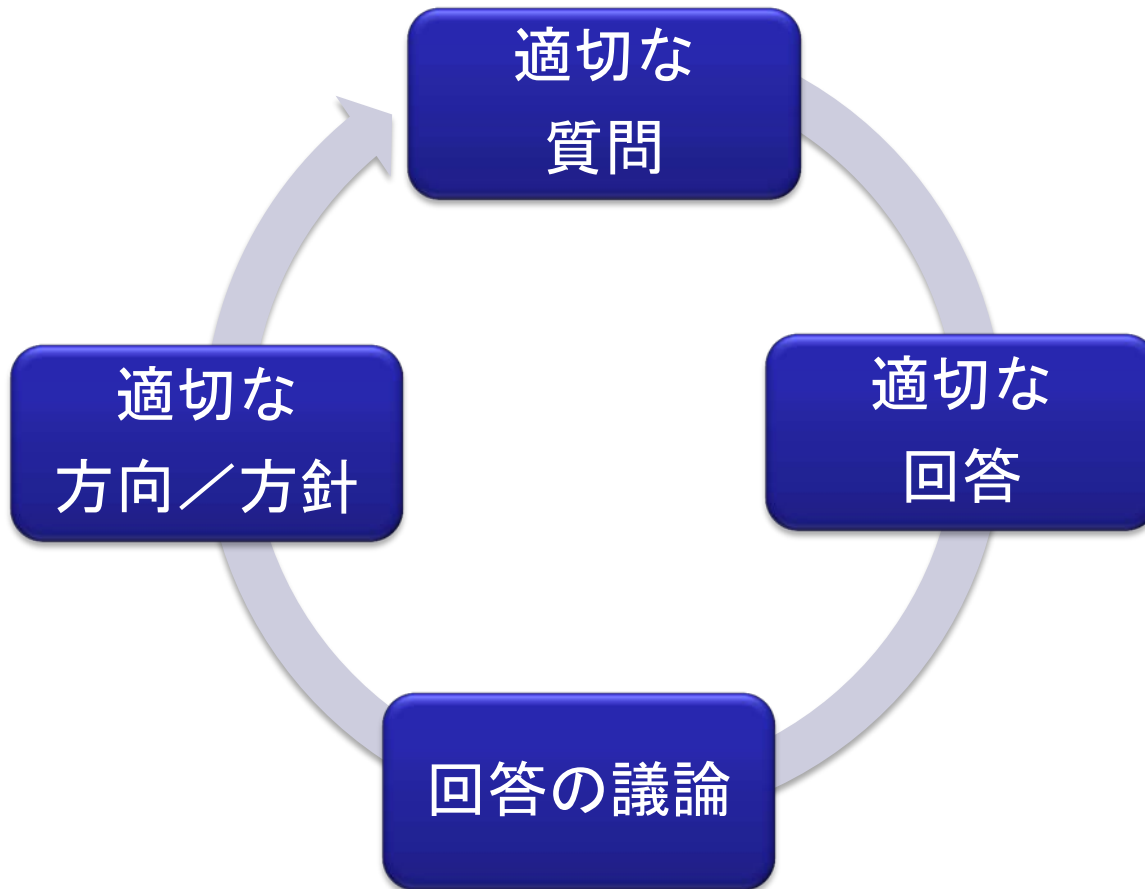
改善 2 : イテレーティブ RCA

- **負荷の軽減 タイムボックス**
 - RCAミーティング : 15分～30分／イテレーション
 - 準備資料は求めない
- **習慣づけとリズムづけ ➡ 負荷の軽減**
 - できるだけ毎日やる
 - 同じ時間にやる
 - ・ 朝会の後、昼食の15分前
- **ミーティングの後、次の質問を考える**
 - 質問策定に時間がかけられる. ➡ 適切な質問
- **タイムボックス内で議論**
 - 知見の伝達.

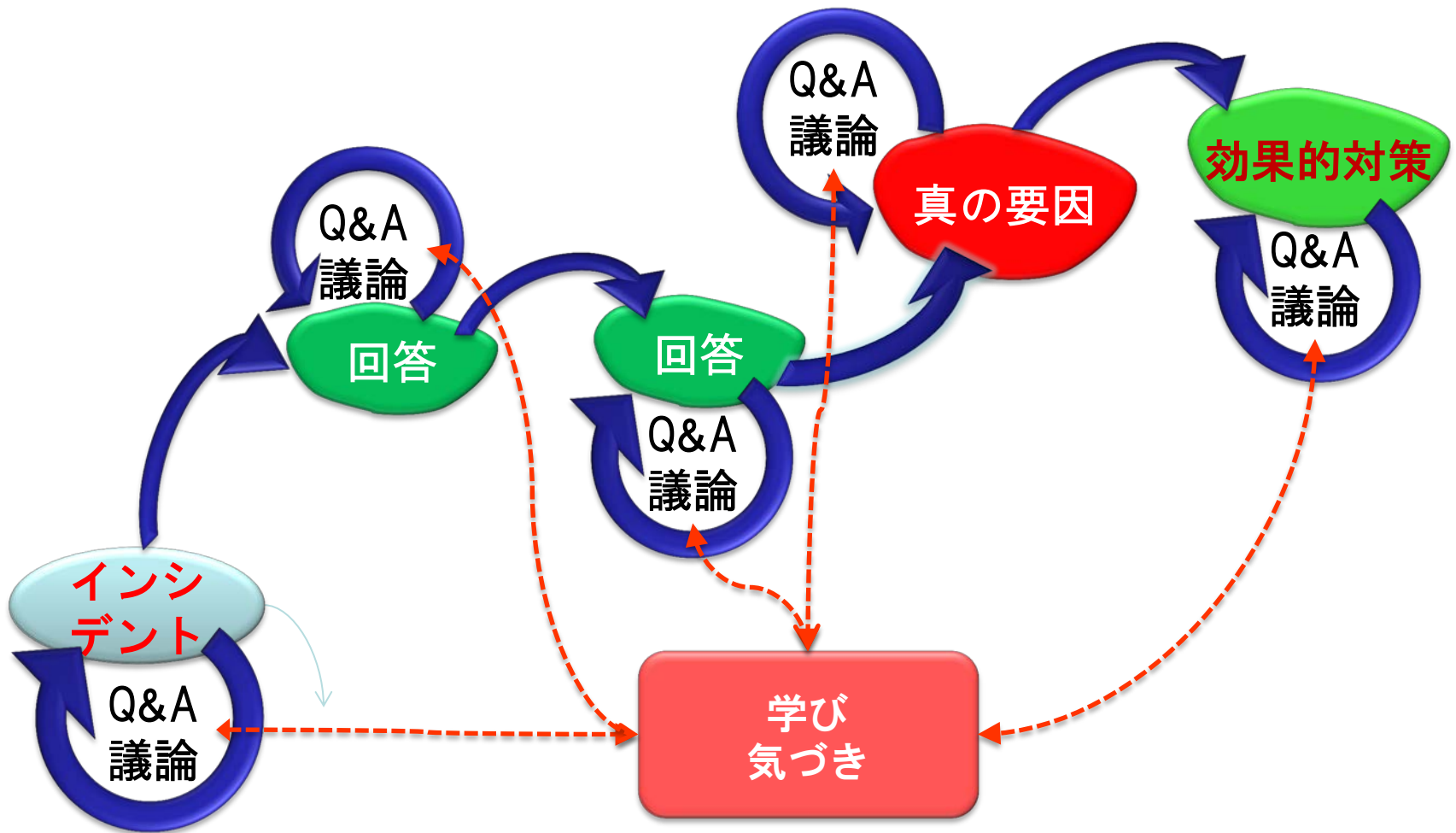
イテレーティブ RCA



イテレーティブRCAループ



イテレーションと適切な質問



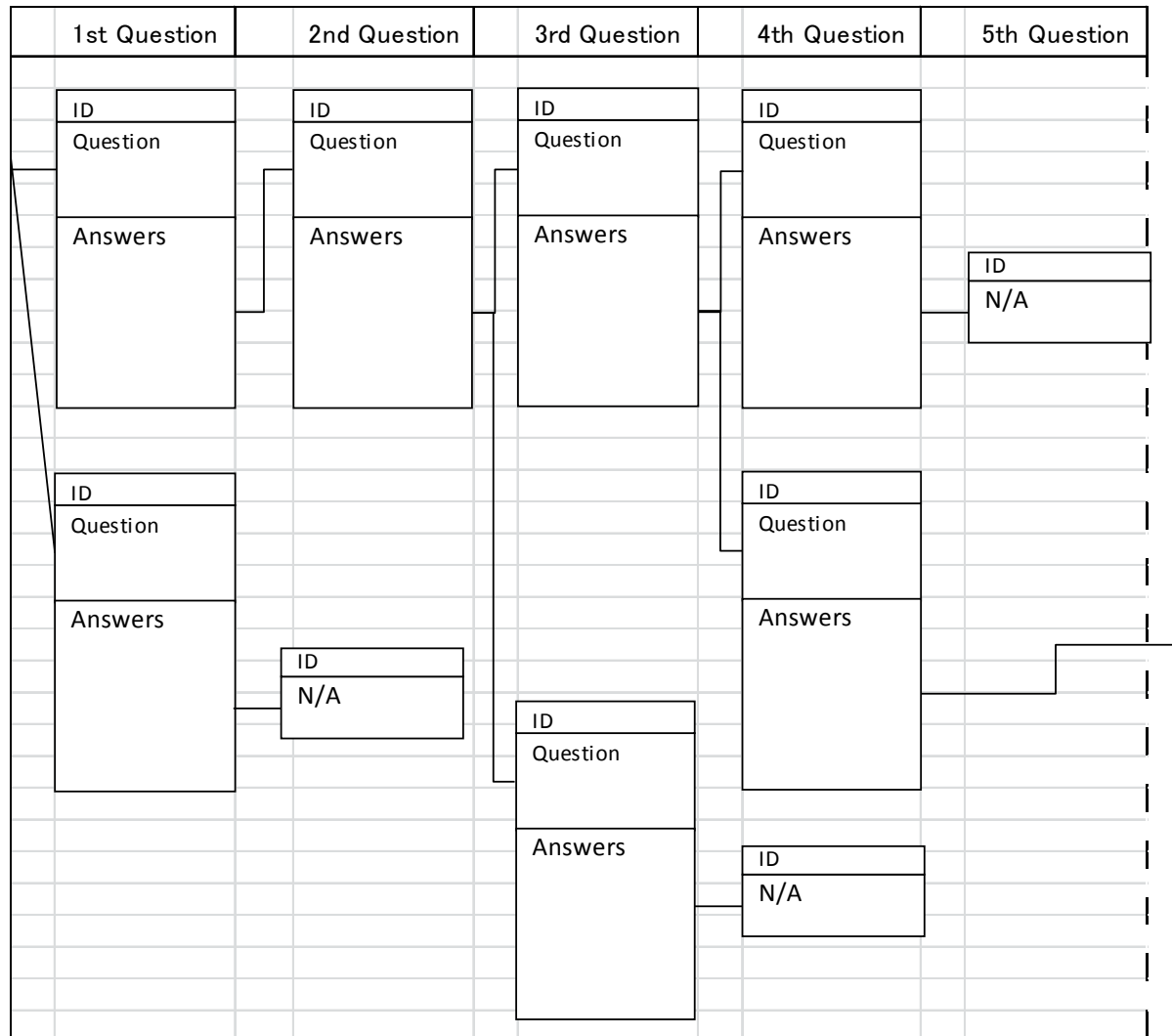
イテレーティブRCAの問題

- 時々次の質問がうまく出せない。
 - トラブルのメカニズムは複雑
 - 答えの内容：欠陥や要因以外の多くの因子がある
 - それぞれの因子はそれぞれ関係を持ち影響を与える。
 - ネットワークの関係
 - 因子の属性もあらわしていきたい。
- なぜなぜ テンプレート
 - 質問と答えしか書けない
 - ツリー構造



- ネットワーク型の欠陥モデルの必要性

ツリー構造のテンプレート



欠陥モデルの必要性：Project Fabre

Quality Engineering for the future

Project Fabre

12. 欠陥モデリング(定義)

誘発因子 (Induction Trigger)	成果物の中に含まれる、人間の思考の誤りを誘発する“トリガー”となる要素のこと。 誘発因子が存在すれば、開発者能力・経験・技術力と関係なく過失が引き起こされやすくなる。
過失因子 (Negligence Factor)	人間の思考や判断の誤りそのもののこと。 欠陥は過失因子の集合(=連続)として生み出される。
増幅因子 (Amplifier)	過失の連鎖を助長し、欠陥の混入確率を増幅させる要素。 多くは定量的に測定可能である。外乱・環境特性ともいう。
欠陥 (Defect)	成果物に含まれた、人間の思考の過ちが具現・表出化したもの。不具合・障害等の「現象」を発生させる。
表出現象 (Incident)	欠陥によって引き起こされる不具合・障害。 多くは定量的に測定／加算可能。

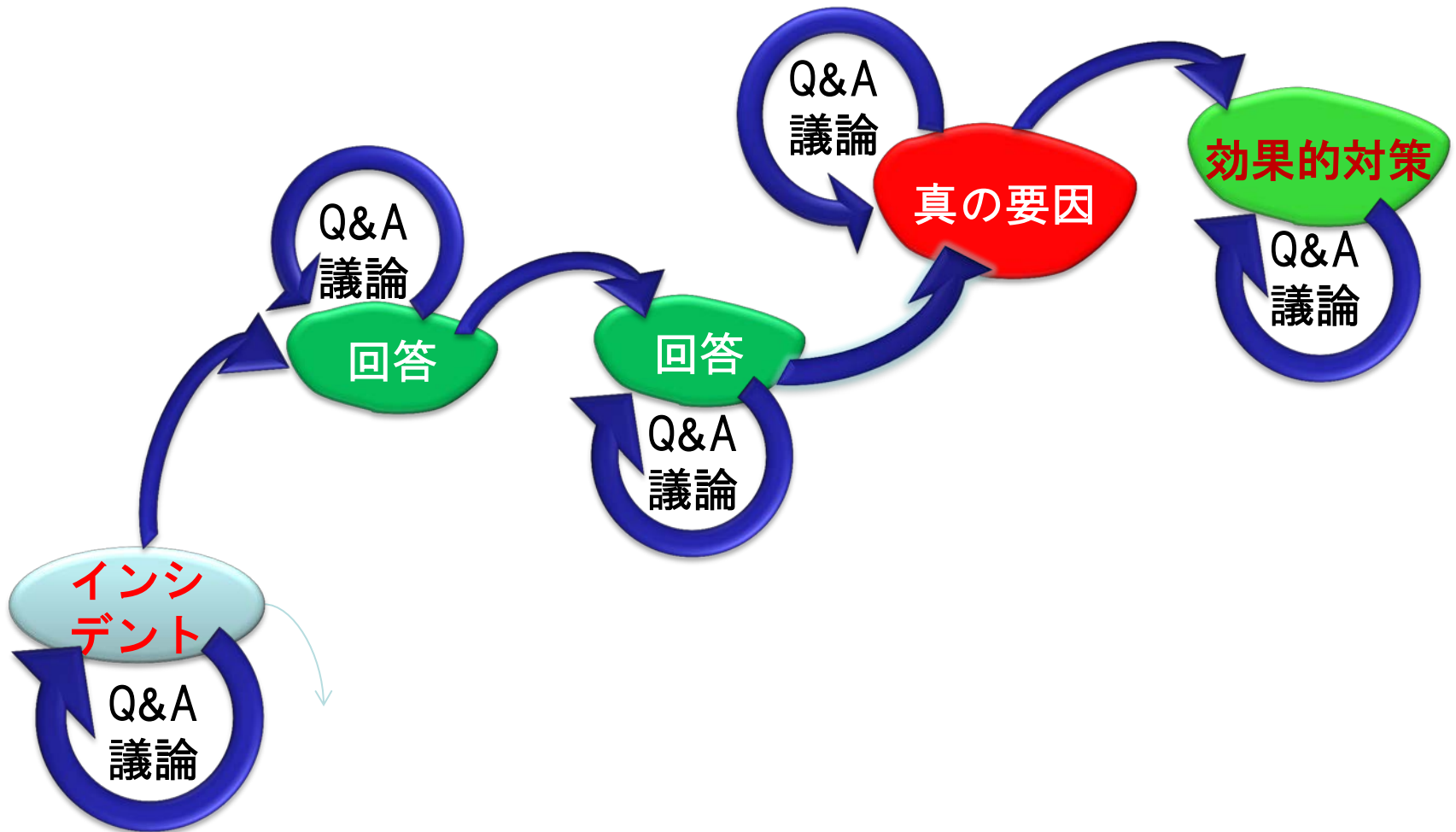
Copyright © 2013 Project Fabre All Rights Reserved. 19

Nobuhiro Hosokawa, Yasuharu Nishi, Aya Ureshino, Makoto Nonaka, Yukiko Hara, JaSST
2013 Tokyo – Project Fabre, 2013

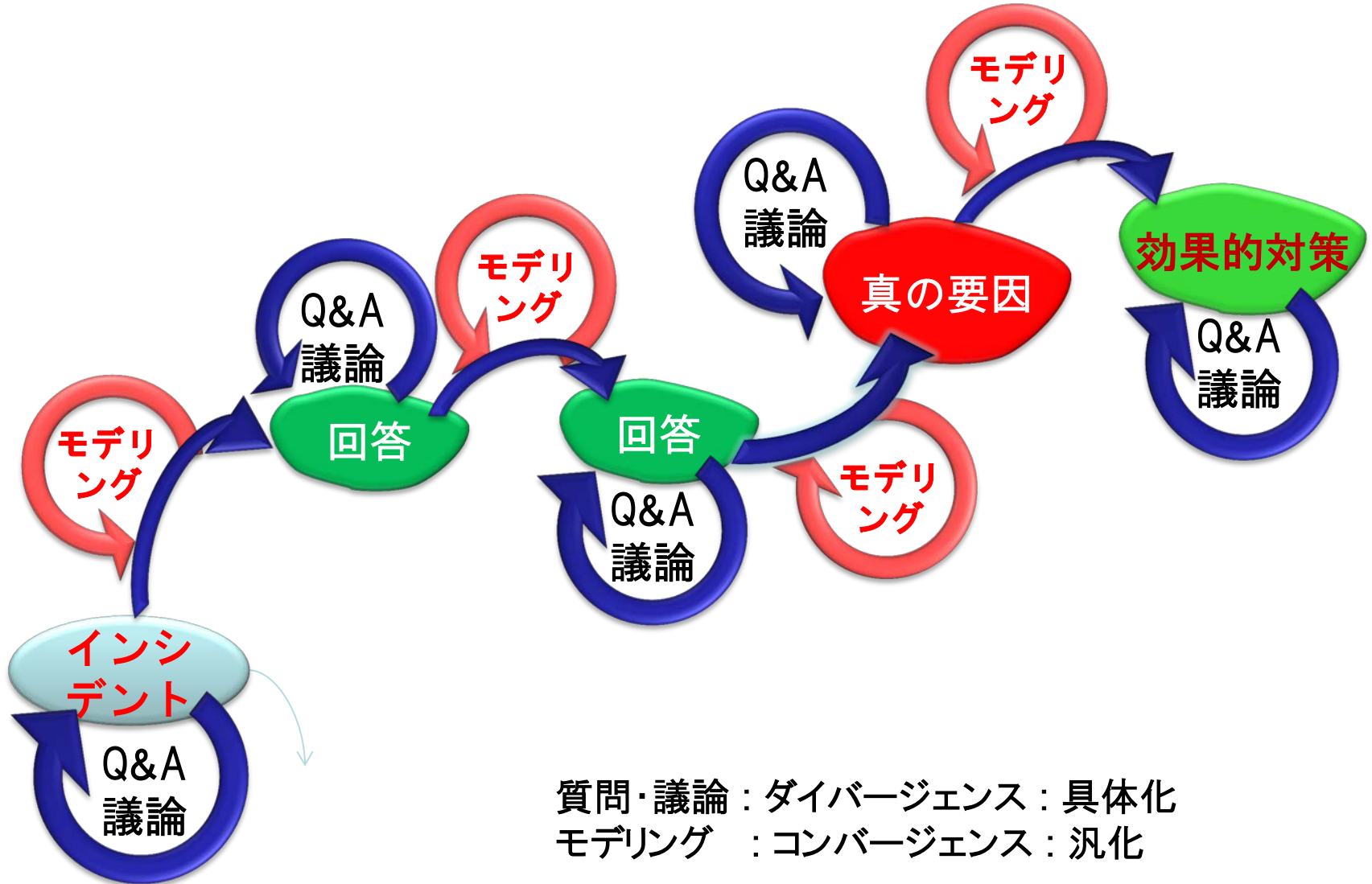
欠陥モデルの例



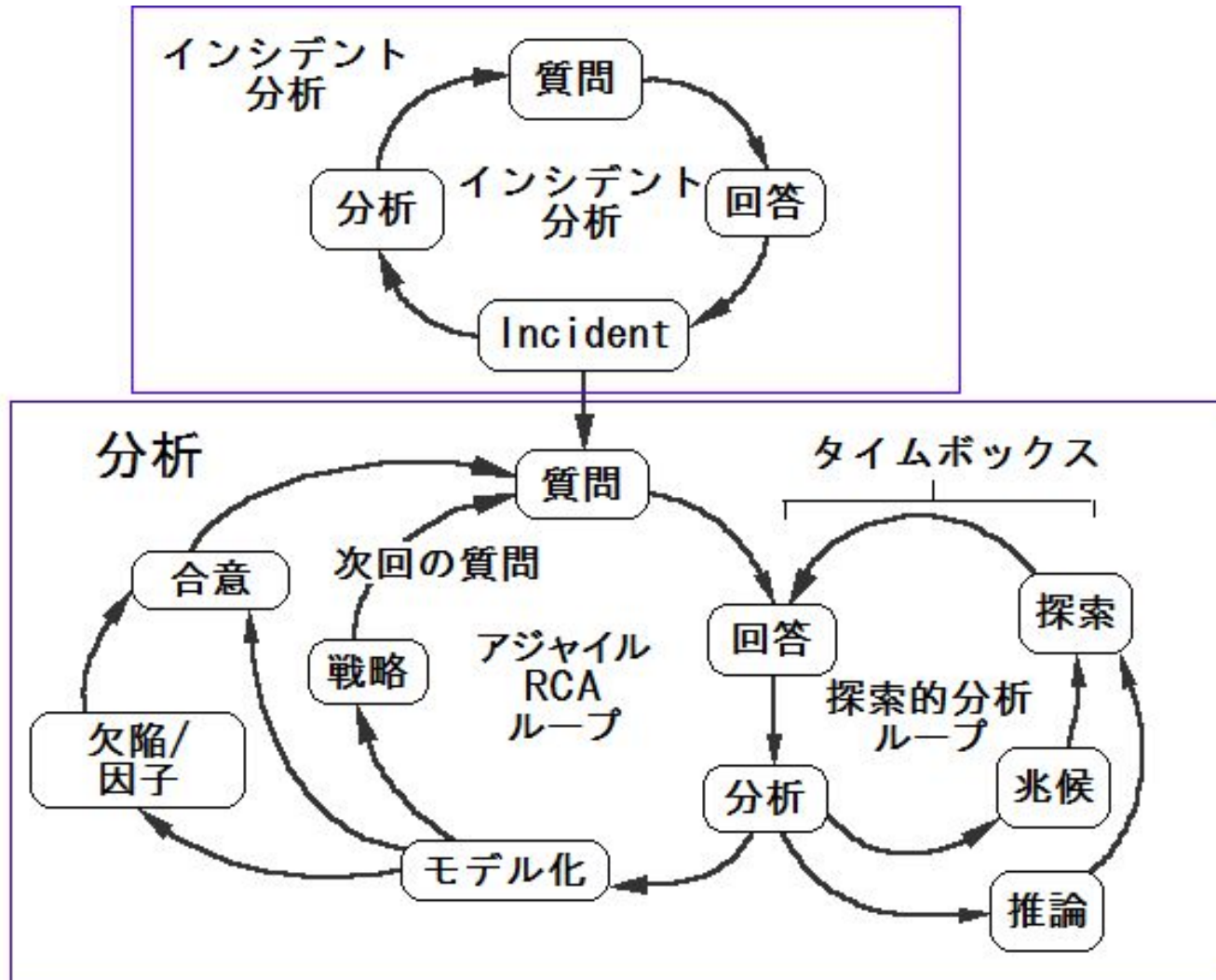
イテレーションがフィードバックループを生む



イテレーションがフィードバックループを生む



アジャイルRCAのプロセス



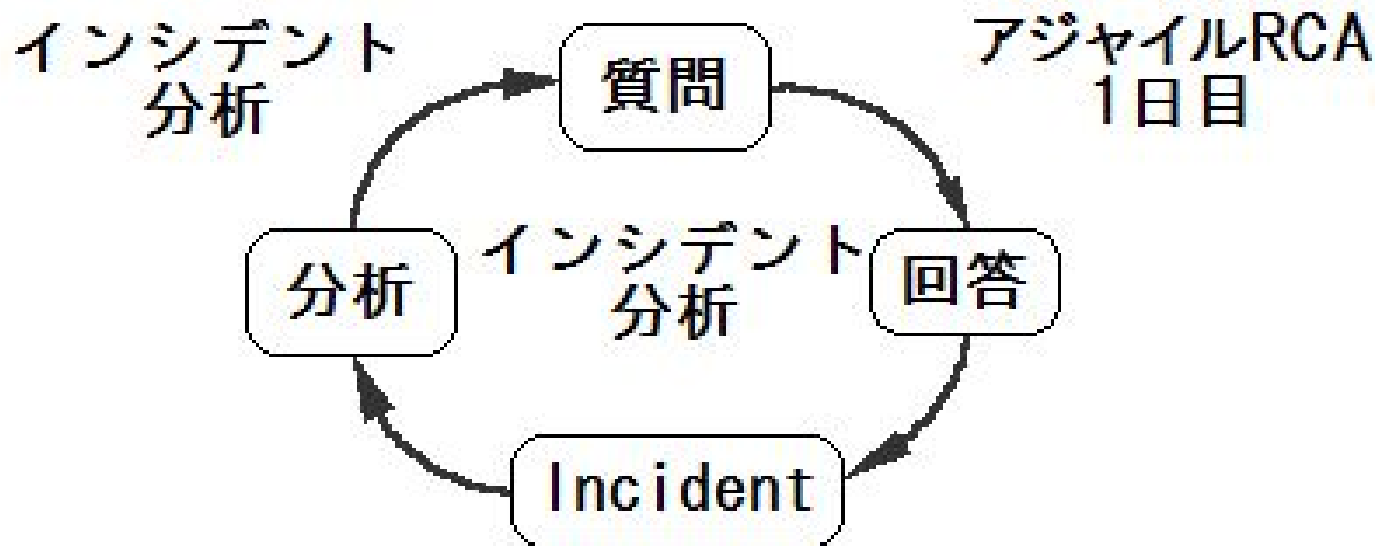
3つのループ

- **インシデント分析ループ**
 - 障害で何が起こったかを分析し共有する
- **探索的分析ループ**
 - 時間まで被分析者の回答を基に探索ループで質問を繰り返し、回答の理解のための確認や、背景や裏付ける事実を引き出していく。.
 - タイムボックス時間厳守
- **アジャイルRCAループ**
 - タイムボックスが終わったあと、探索ループで得られた回答を分析して欠陥をモデル化し、戦略をもって次の質問を策定していく。

分析チーム

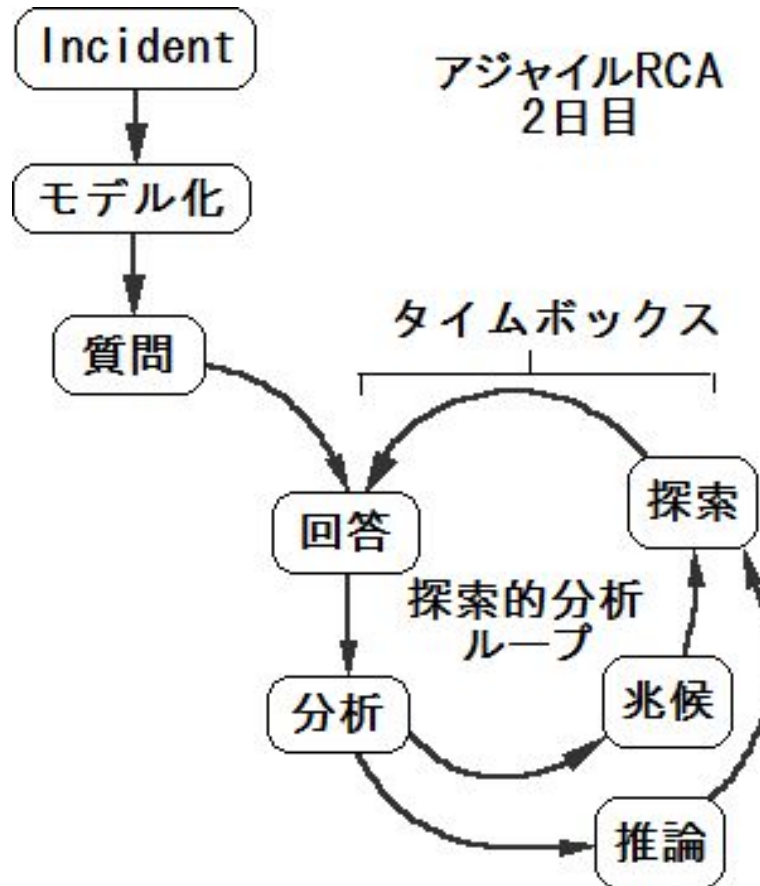
- **モデレータ**
 - **アジャイルRCAの目的、原理を理解して、プロセスをドライブする人**
 - **欠陥モデルの作成をする**
 - **質問の作成、戦略建てを行う**
- **担当者**
- **レビューア**
 - **モデレータと協調して、戦略に基づいて質問の作成、ファシリテーションをサポートする。**
 - **欠陥モデルのレビューを行い、戦略建てのサポートをする**

イテレーション1日目: インシデント分析



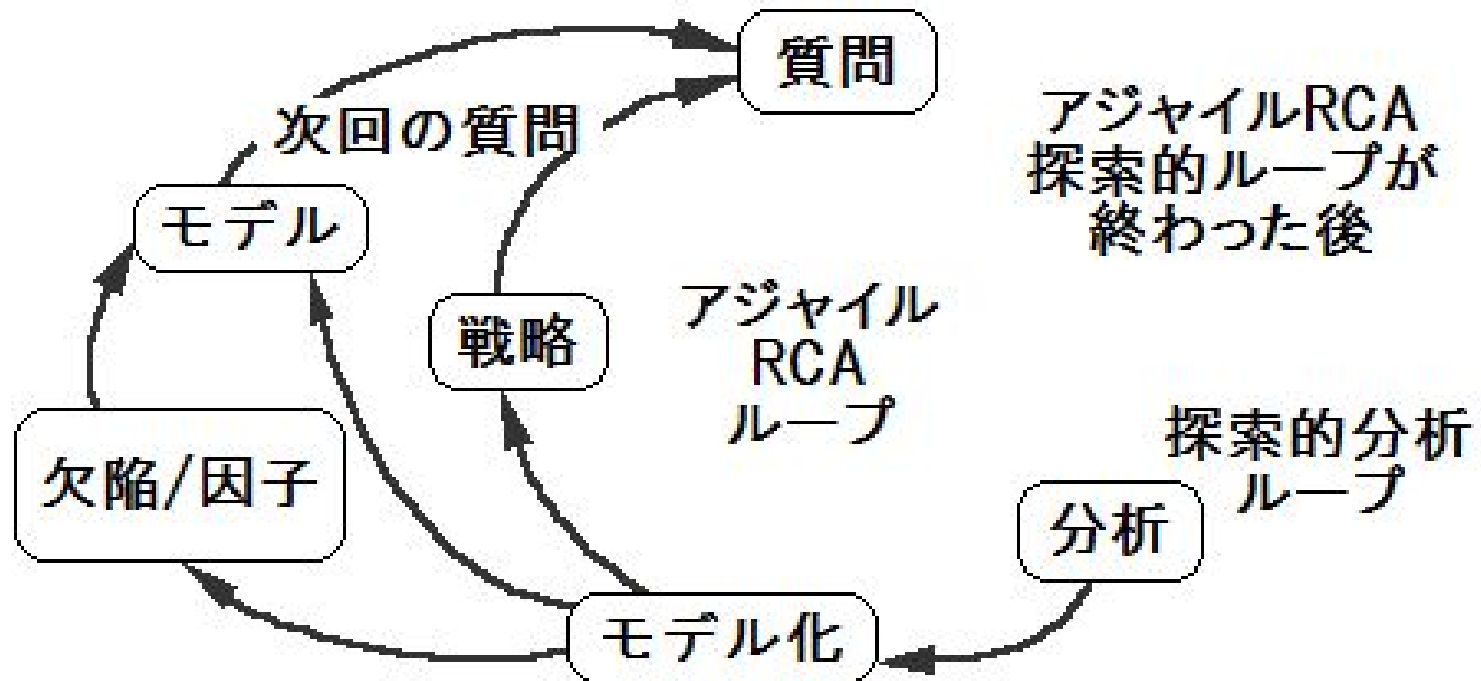
- 事実、背景、状況、起きたことだけを聞く
- 分析は我慢する
- 傾聴、パラフレーズが有効

イテレーション2日目: 探索的分析



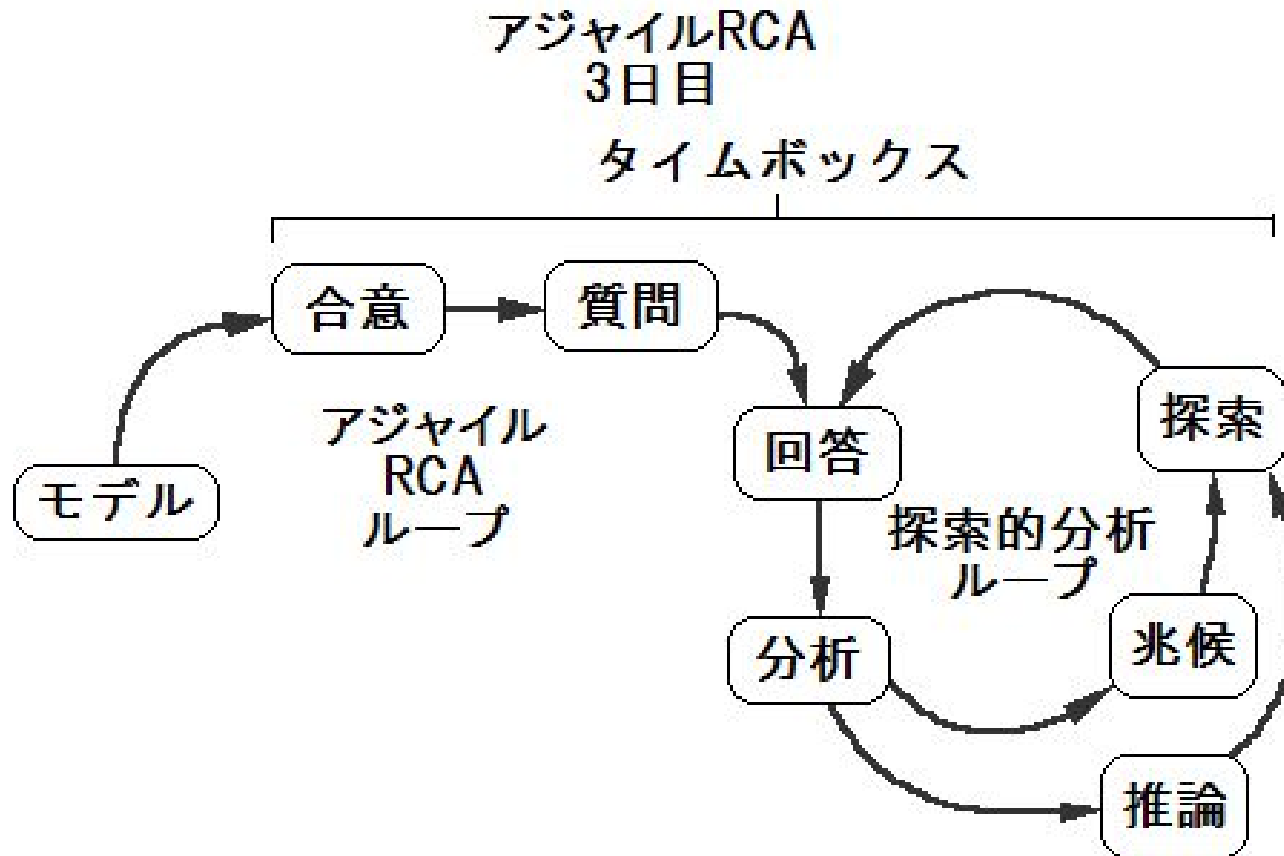
インシデント分析ループで作られた質問をベースに
探索的に質問と回答を繰り返すループ

探索的ループの後：モデリング



欠陥モデルの作成
それをもとに次のイテレーションの質問を策定

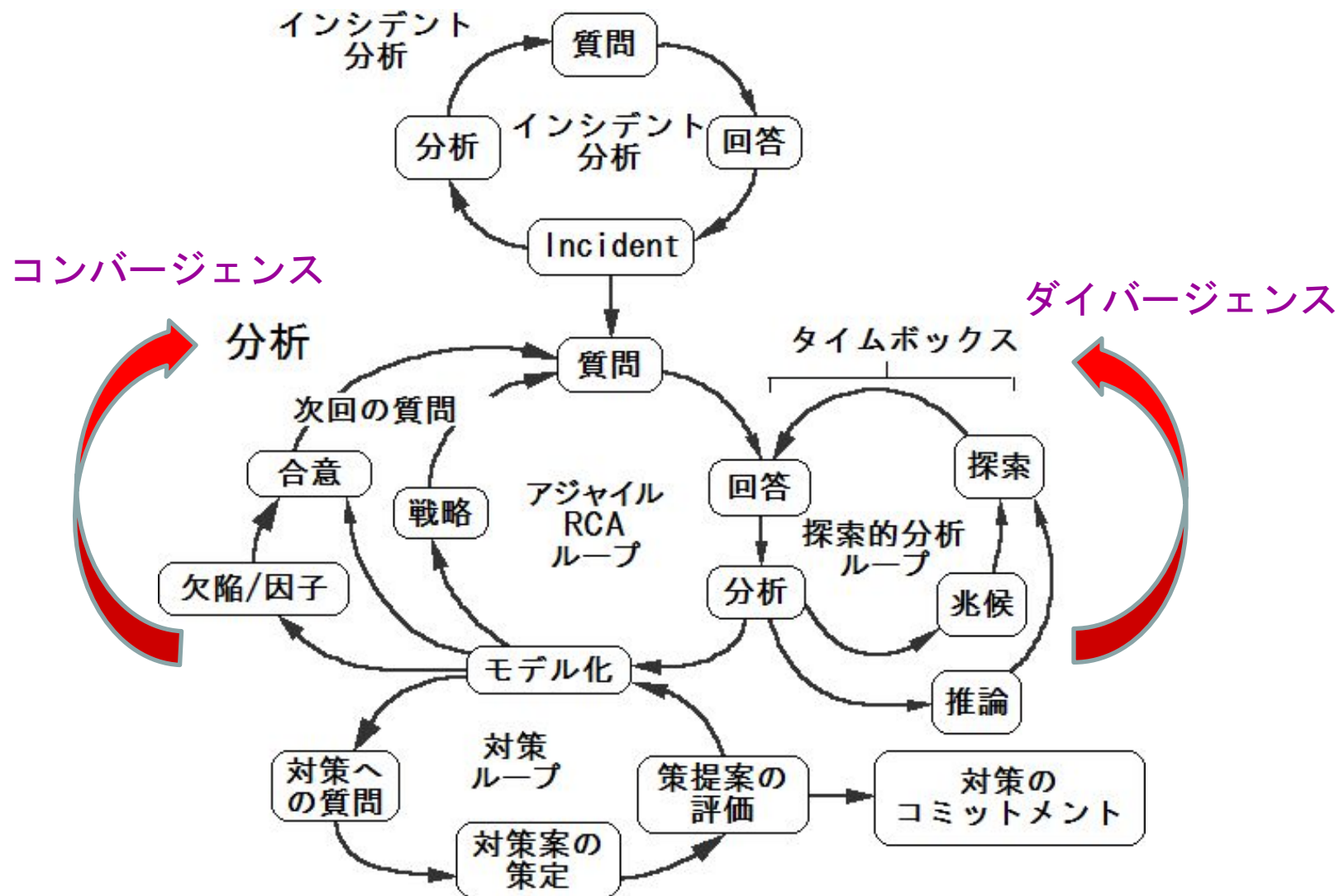
イテレーション1日目: 欠陥モデルで質問する



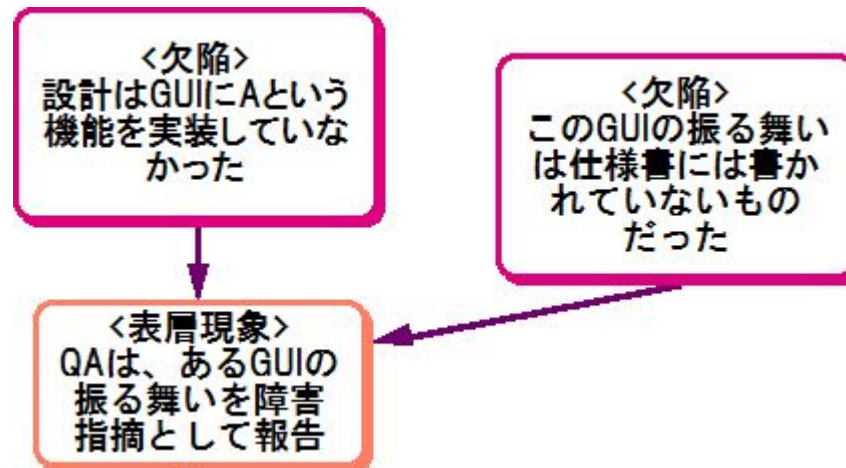
再び探索的ループに入る

1. アジャイルRCAループで作った欠陥モデルをチームで合意する
2. アジャイルRCAループで策定した質問をベースに探索的分析ループを回す

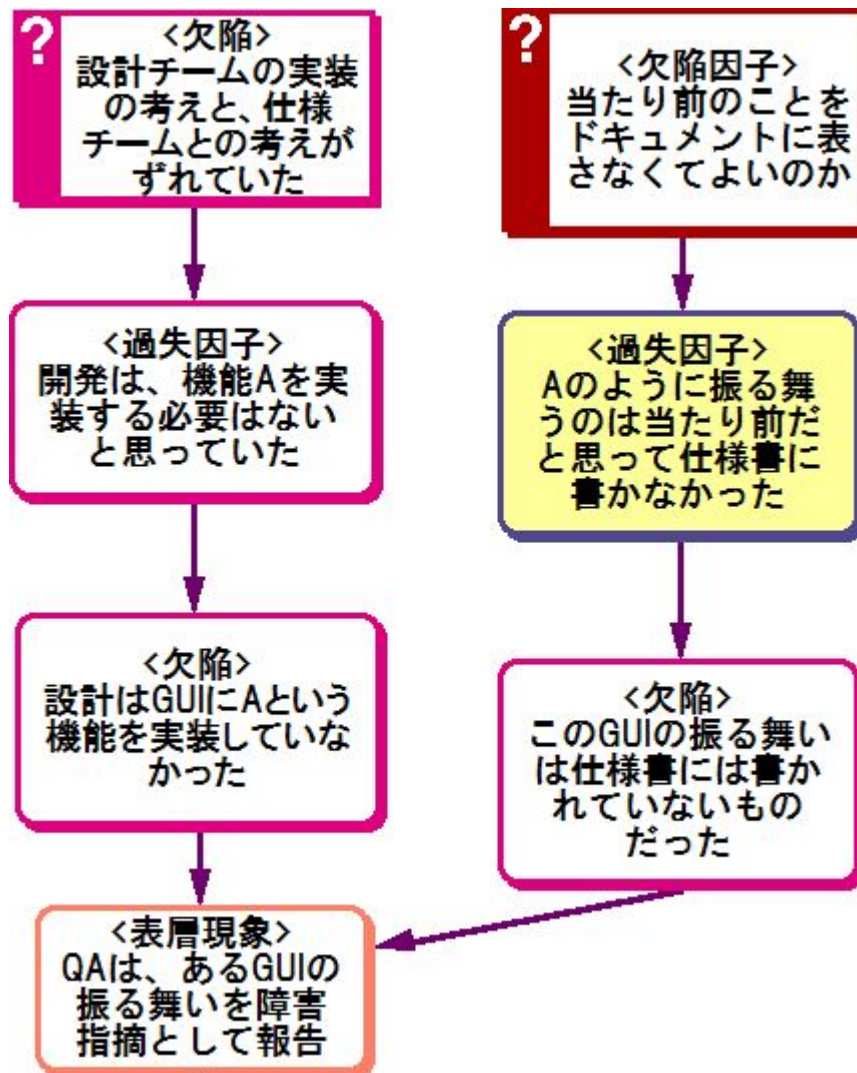
対策まで含めたアジャイルRCA



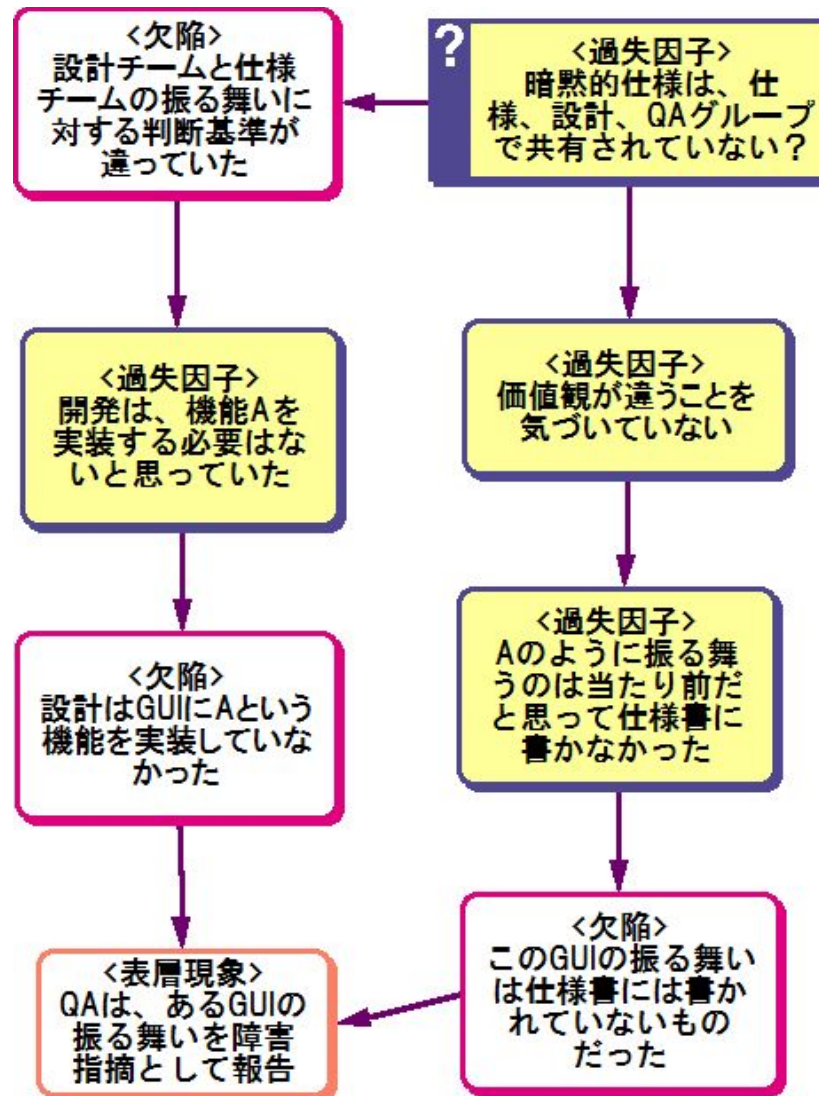
事例：1日目



事例：2日目



事例：3日目



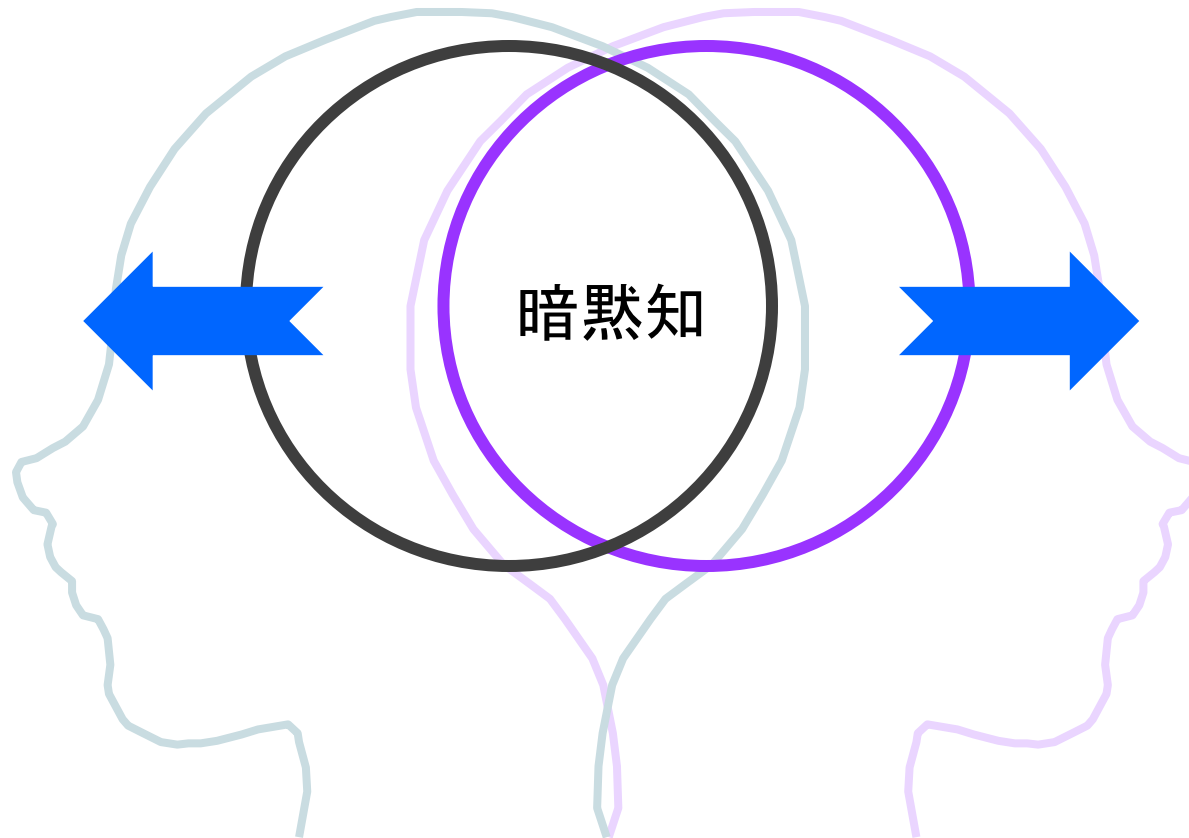
暗黙知



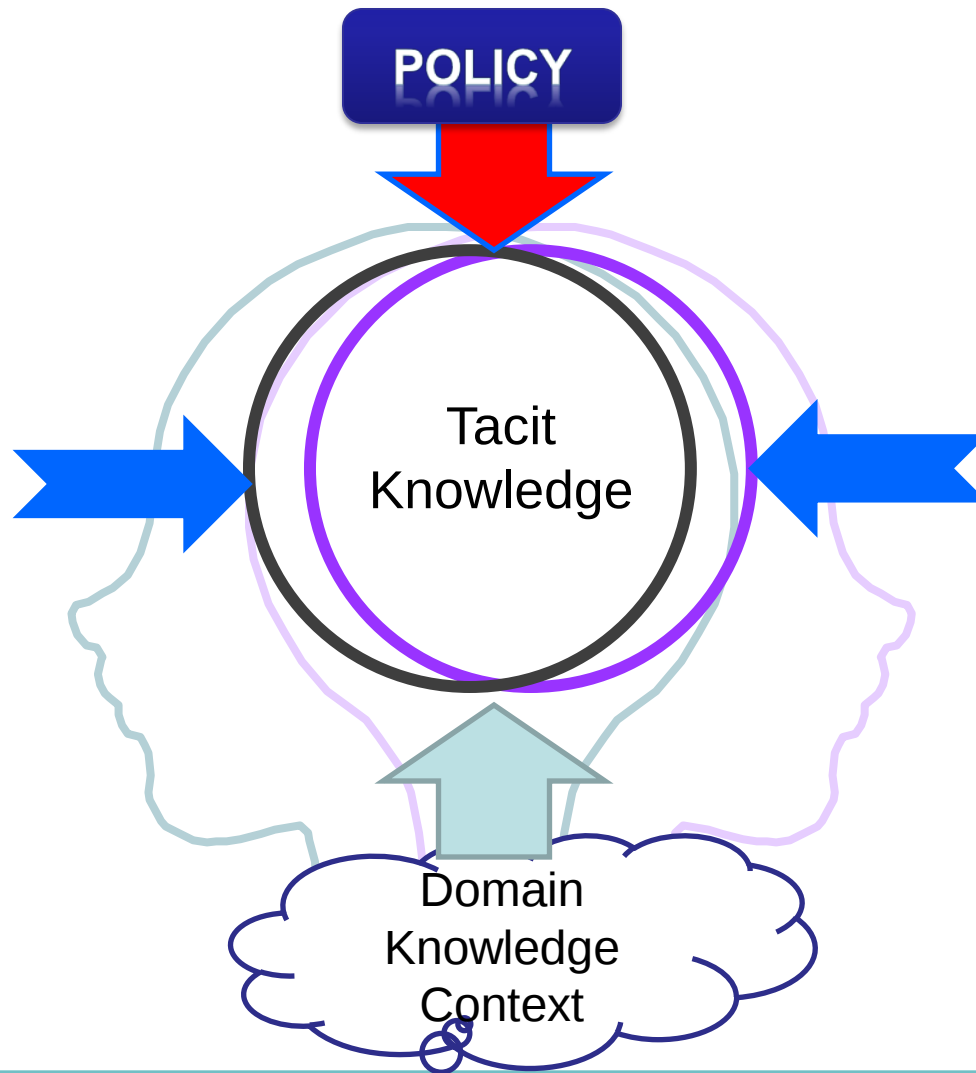
暗黙知によるコミュニケーション



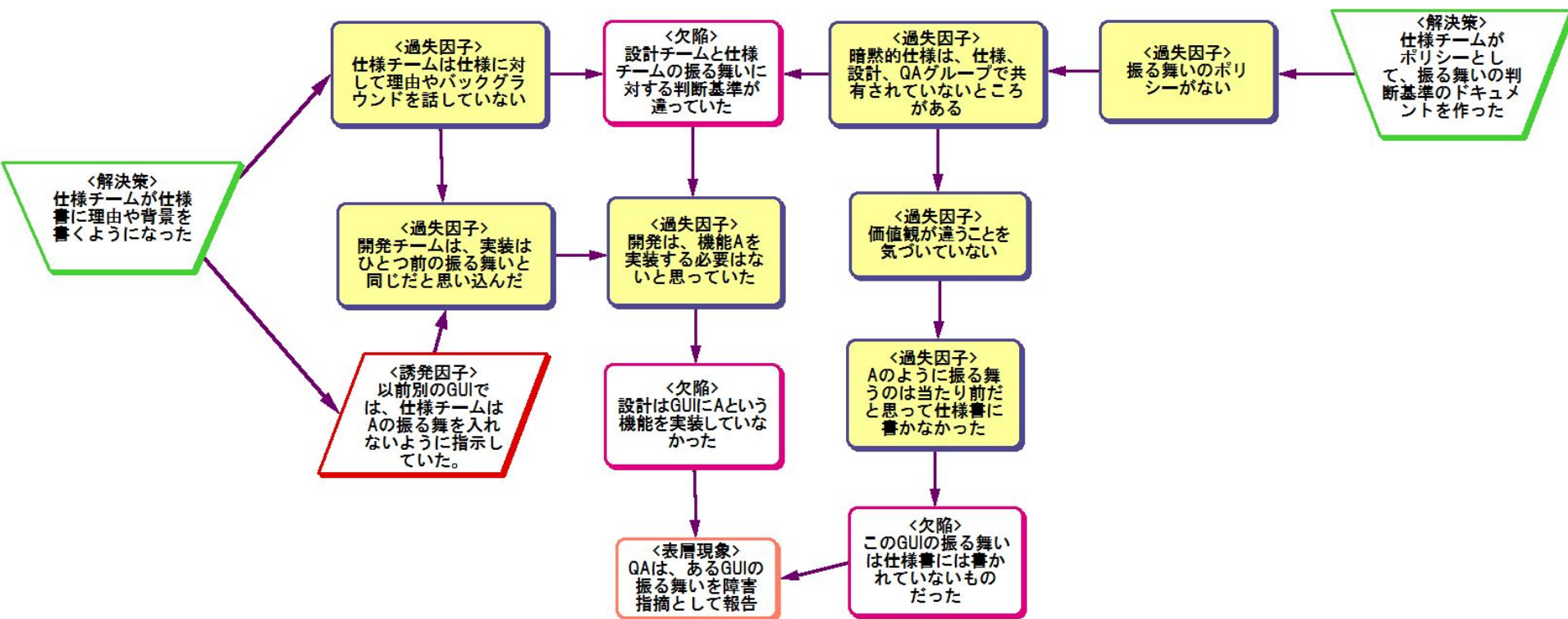
暗黙知の齟齬 = 想定外



ポリシーで暗黙知をサポートする



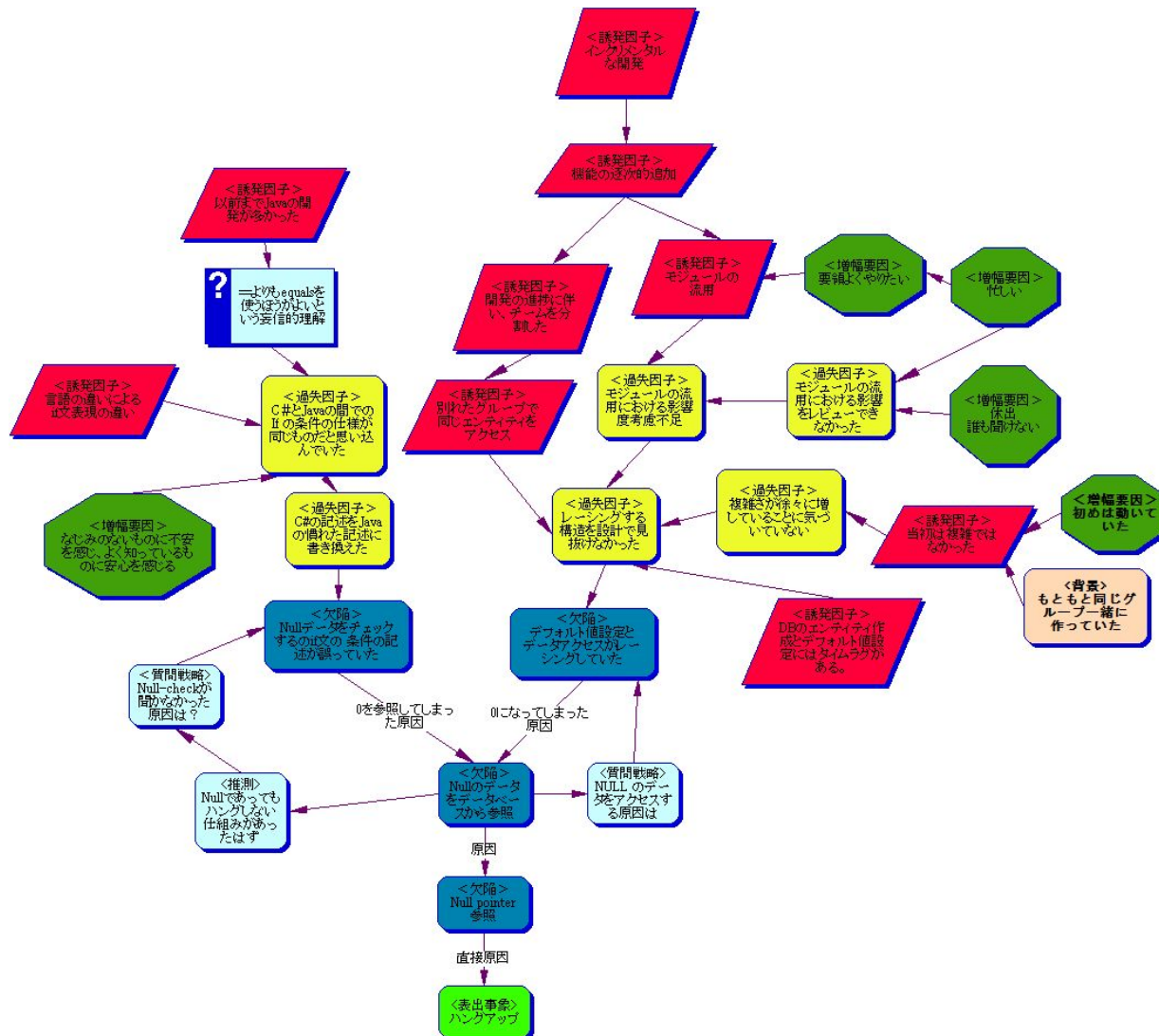
事例：最終日



ファシリテーションのコツ

- 欠陥が見つかったとき、すぐにその欠陥に突っ込まない
- その因子を探る
 - 状況、条件、背景
 - どのように作られたか
 - 与えられた条件
 - 言い訳
 - 愚痴
- 言ったことをその場でモデルに書き込んでいく

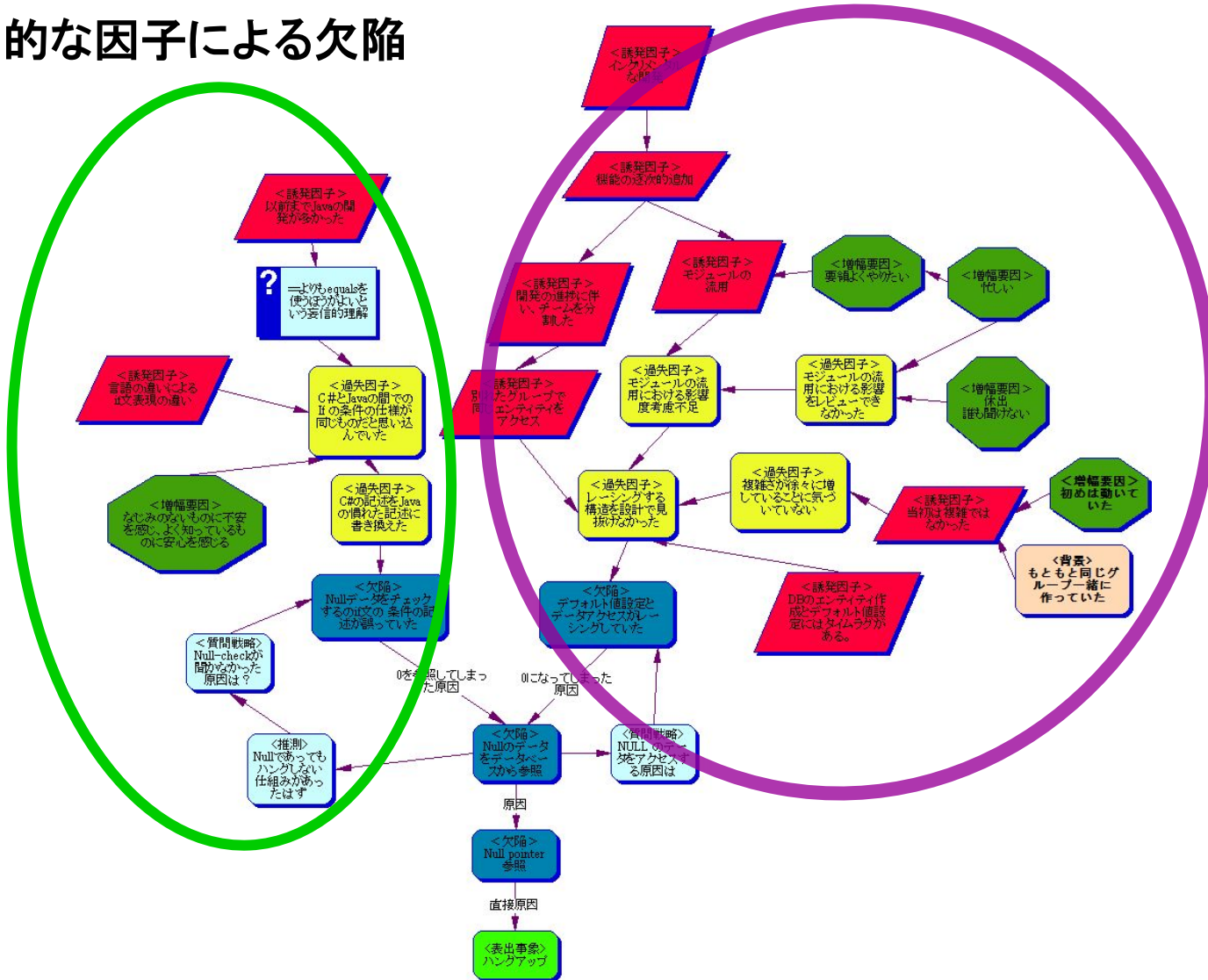
事例その2



事例その2

プロセス因子による欠陥

個人的な因子による欠陥



アジャイルRCAの効果 1

- **継続性**
 - **従来**
 - ・ 時間がかかる重たい作業で、継続しなかった.
 - **アジャイルRCA**
 - ・ 二つのチームで6か月以上継続.
 - ・ 習慣的に行うモチベーションが出てきた.

アジャイルRCAの効果 2

- **分析スキルの改善**

- **分析スピード**

- 4倍以上の改善

- 従来

- トータル8時間

- アジャイルRCA

- トータル2時間 (min 1.5時間)

- **真の原因への到達**

- 従来

- 到達しない場合もしばしばあった

- アジャイルRCA

- およそ1.5時間で到達

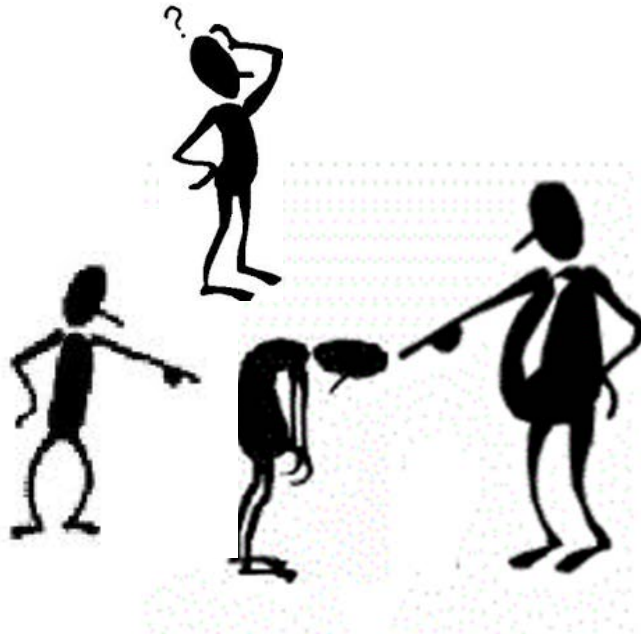
- **ファシリテーションの向上**

アジャイルRCAの効果 3

- チームとの信頼関係が生まれる
 - 担当者をリスペクトする
 - ・ 貴重な情報を提供してくれた人
- 人を責めない分析
 - 分析の対象は人ではなく欠陥モデル
 - 初めから正しくやることの難しさがわかる
 - もしあなたも、その分析と同じ状況、条件、環境、タイミングにおかれたら、同じ間違いを犯すことを痛烈に認識するはず
- 現場からの品質改善のモチベーションが高まる

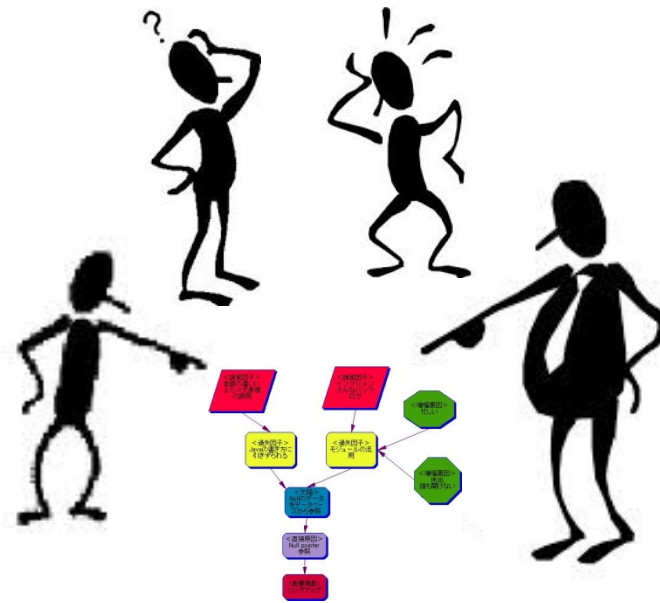
客観的、冷静な分析

使用前



欠陥：物理的な問題
対象：現場のモノ

使用後



欠陥：バグ
対象：欠陥モデル

アジャイルRCAの結果

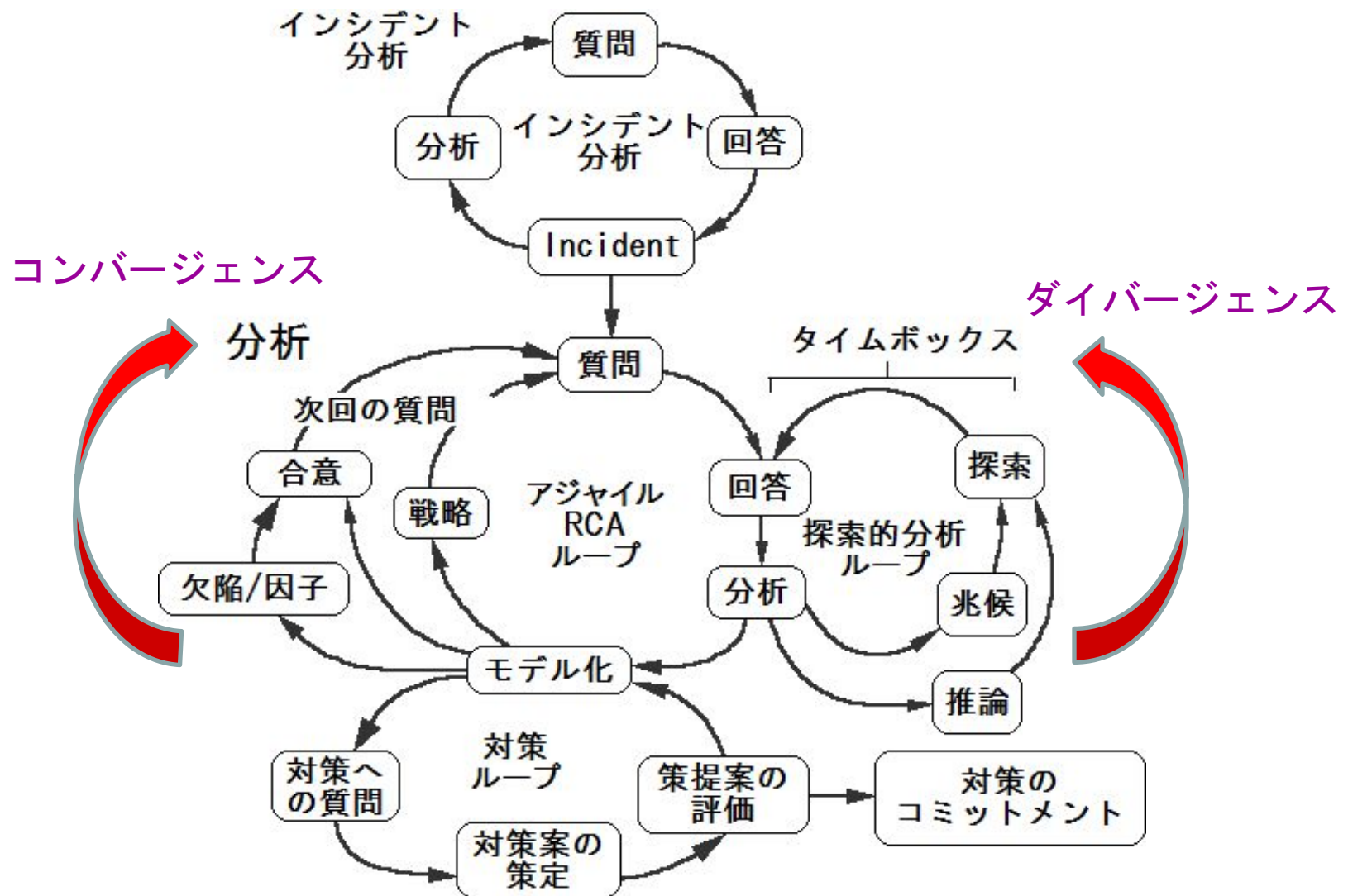
	課題数	イテレーション	欠陥数	因子数	トータル時間 (分)
チームA	6	4	2	15	103
チームB	7	4	4	21	133

課題数 : 扱った課題の数
イテレーション : 一つの課題に対するイテレーション回数の平均
欠陥数 : 一つの課題に対する欠陥の数の平均
因子数 : 一つの課題に対する因子の数の平均
トータル時間 : アジャイルRCAのトータルの時間の平均.

アジャイルという理由

- 複数の改善のループ
 - 欠陥モデルがイテレーションごとに進化する
 - 欠陥モデルを毎回評価し合意している
 - 欠陥モデルの上で考え議論をしている
 - 次のイテレーションに対して欠陥モデルを剪定している
- .
- アジャイルRCAは、欠陥モデルを用いて複数のイテレーティブなループにより継続的に要因分析の改善をしている

対策まで含めたアジャイルRCA



アジャイルRCAの課題

- 欠陥モデルの改善
 - 因子の整備改善
 - モデリングのスピードアップ
 - パターン化／メタモデリング
- 質問生成とその戦略の改善
- モデリングの利用
 - 未然防止活動：プロセス、レビュー、テストニング
 - モデリングのデータベース化
- アジャイルRCAプロセスのスピードアップ
 - リアルタイムプロセス

Agile RCA

ご清聴ありがとうございました

Reference

- **Agile Specification Quality Control: Shifting emphasis from cleanup to sampling defects, Tom Gilb, 2005**
- **Nobuhiro Hosokawa, Yasuharu Nishi, Aya Ureshino, Makoto Nonaka, Yukiko Hara, JaSST 2013 Tokyo – Project Fabre, 2013**
- **THE ART OF POWERFUL QUESTIONS, Eric E.Vogt, Juanita Brown, and David Isaacs, 2003**
- **Software Inspection, Tom Gilb, Dorothy Graham, 1993**
- **The Practical Guide to Defect Prevention, Marc McDonald, Robert Musson, Ross Smith, 2007, Microsoft Press**
- **ODC – a 10x for Root Cause Analysis, Ram Chillarege, 2006**
- **Naze Naze Bunseki, Hitoshi Ogura, Nikkei BP, 2010**