

# 生成AIを活用した ステークホルダー起点のレビュー観点導出手法

～上級レビューア-との“境”をゼロへ! 『SAKAI』法の提案～

## 研究コース2 ソフトウェアレビュー SAKAIチーム

伊井野 理恵子(株式会社 AGEST)

鈴木 弘章 (キヤノンイメージングシステムズ株式会社)

降旗 夏子 (MHI エアロスペースシステムズ株式会社)



SAKAI法

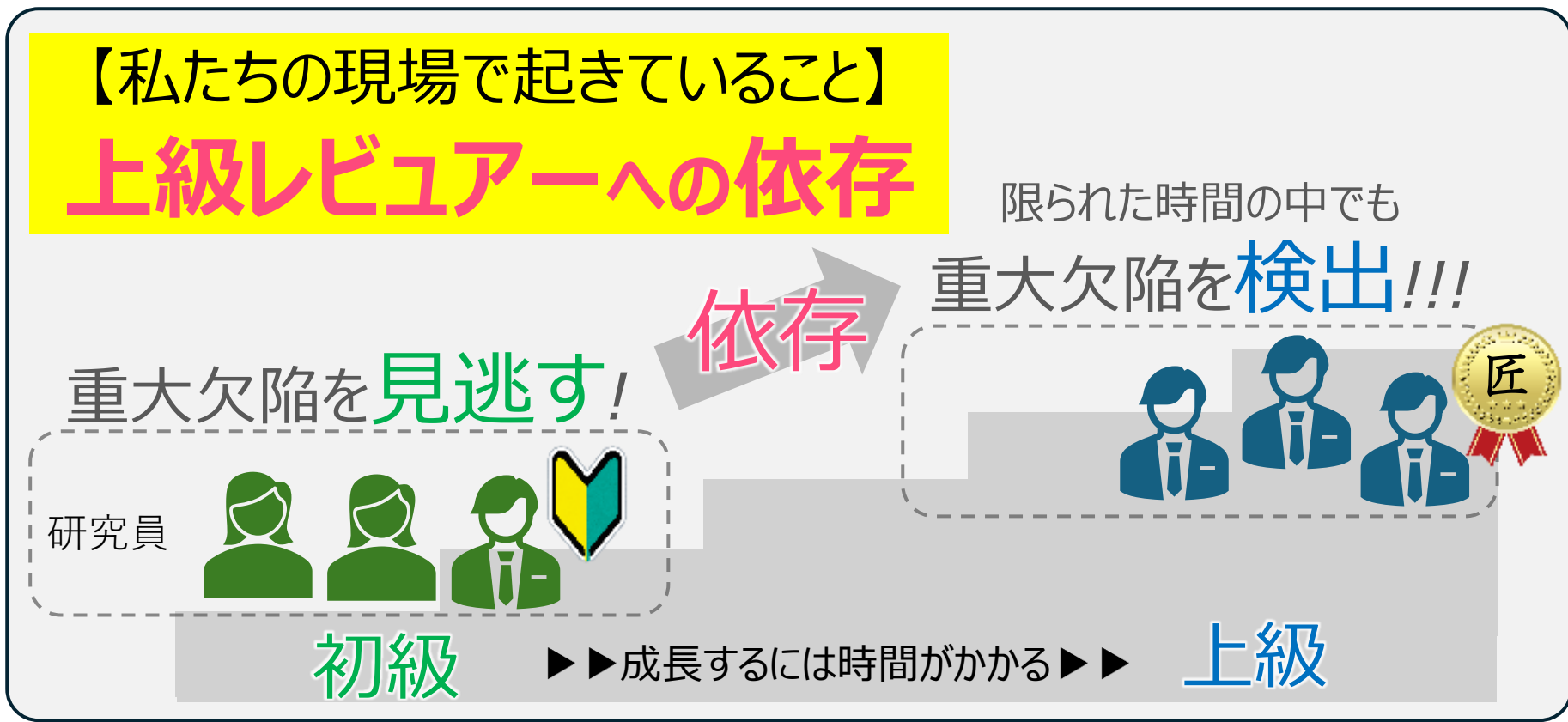


# 1 背景

# 1. 背景

1

課題



上級レビュアーのように、限られた時間の中で、重大欠陥を指摘できるようにする  
これを**今すぐ**実現するためには、どうすればよいのか！？

上級との  
違いは？

# 1. 背景

2

着目点

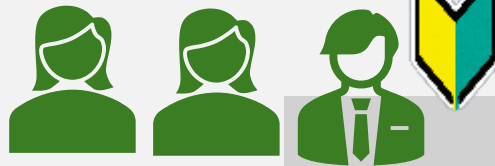
上級レビューア-とは**何が違うの？**

レビュー対象物を見た瞬間に

**どういう視点で何を見るか**を判断

読んで気になった所に反応

研究員



初級



重大欠陥につながる可能性の高い**「観点」**

上級

この「観点」を**短時間で効率よく**  
**誰でも活用できる**形にできないか？



**SAKAI法**の出発点

# 2

## 提案 手法

生成AIを活用した  
ステークホルダー起点のレビュー観点導出手法

# SAKAI法

(Stakeholder Action Kanshingo extraction supported by AI) Method



# 2. 提案手法

## 『SAKAI』法の狙い

重大欠陥の検出に**有効な観点**を、**短時間**で導出し、レビューに活用できるようにすること

1

特徴

### 2つの特徴

特徴

1

レビュー対象の特性に応じた  
**効果**的なやり方

#### 観点導出の方法

レビュー対象の特性に応じた観点を導出できる既存研究(SAKE手法)を参考に再設計

特徴

2

現場で運用可能なレベルの  
**効率**的なやり方

#### 生成AIの活用

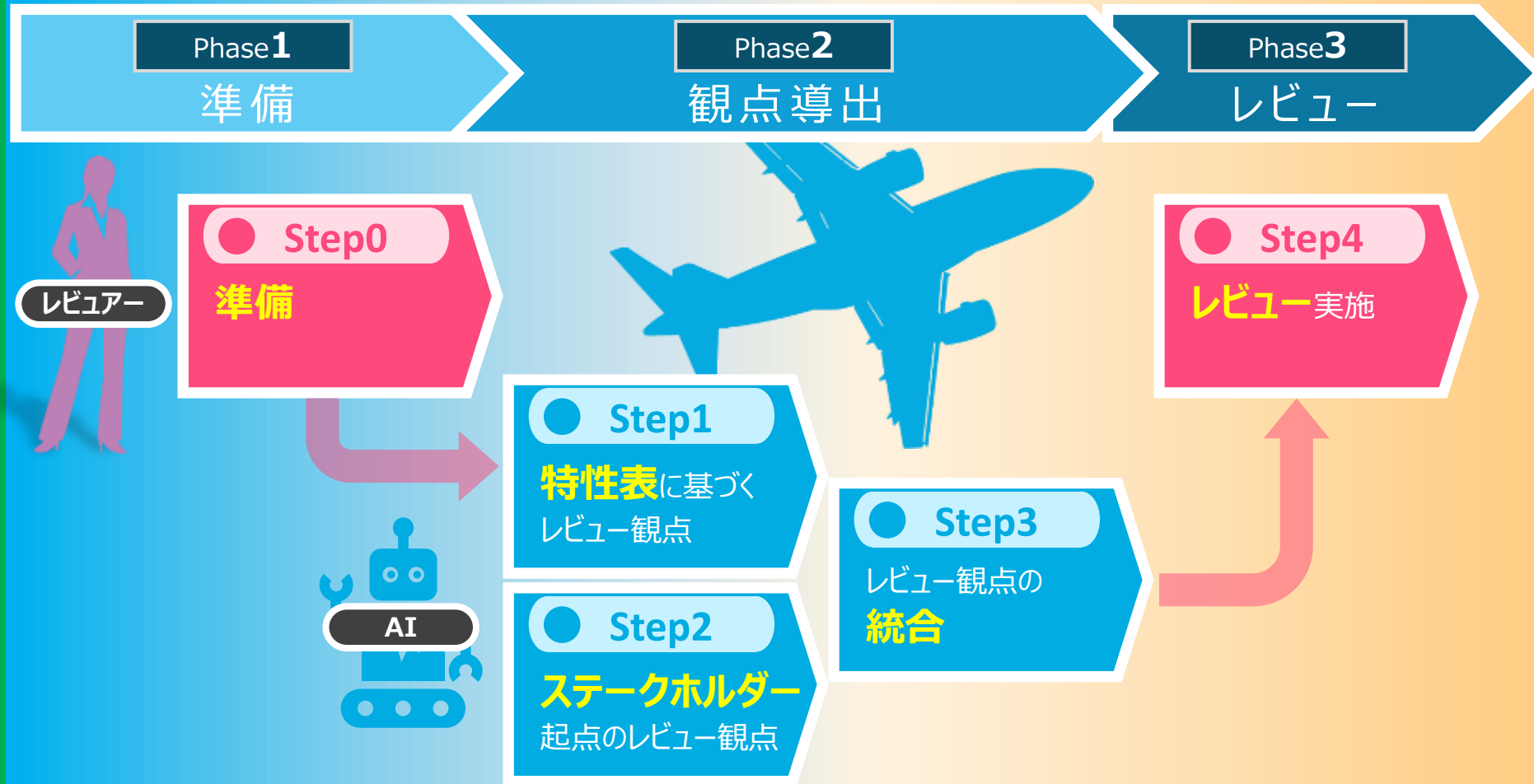
再設計した観点導出のプロセスを生成AIを活用することでより現場で回せる形にする

# 2. 提案 手法

2

手順

## SAKAI法の全体プロセス



# 2. 提案 手法

2

手順

## SAKAI法の全体プロセス



# Phase1 準備

Phase1

準備

Phase2

観占導出

Phase3

レビュー

● Step0

準備

レビュアー

僅かな  
準備で

AI

## Step0 準備

INPUT

レビュー対象

プロジェクトに  
関する情報

プロジェクトの品質方針、システムの特徴、  
過去に発生した障害やトラブルの情報など

漏れてほしくない  
要素の抽出

レビュアー

「プロジェクト固有の重要な要素」  
を抽出

OUTPUT

生成AIに渡す追加情報

「クレマー」...

重要な  
ステークホルダー

見逃したくない  
重大欠陥

「hoge hoge」...

プロジェクト固有の重要な要素を抽出する

スタートホルダー  
起点のレビュー観点

# Phase2 観点導出

Phase1

準備

Phase2

観点導出

Phase3

レビュー

● Step0

準備

レビュアー

僅かな  
準備で

AI

● Step1

**特性表**に基づく  
レビュー観点

● Step2

**ステークホルダー**  
起点のレビュー観点

たった  
数分で

● Step3

レビュー観点の  
**統合**

● Step4

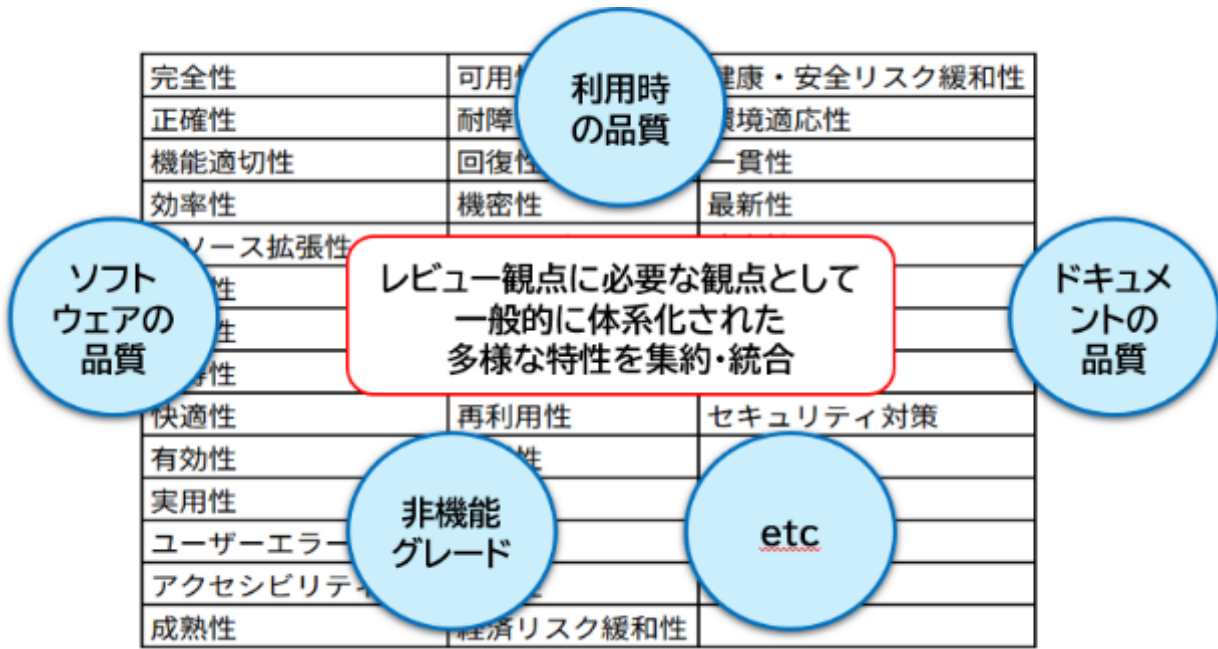
レビュー実施

有効な  
観点が

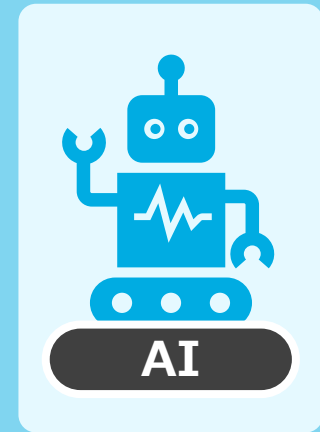
# 特性表に基づくレビュー観点

## ※ 1 特性表

既存研究(SAKE手法)にて体系的に整理された  
抽象度の高いレビュー観点集



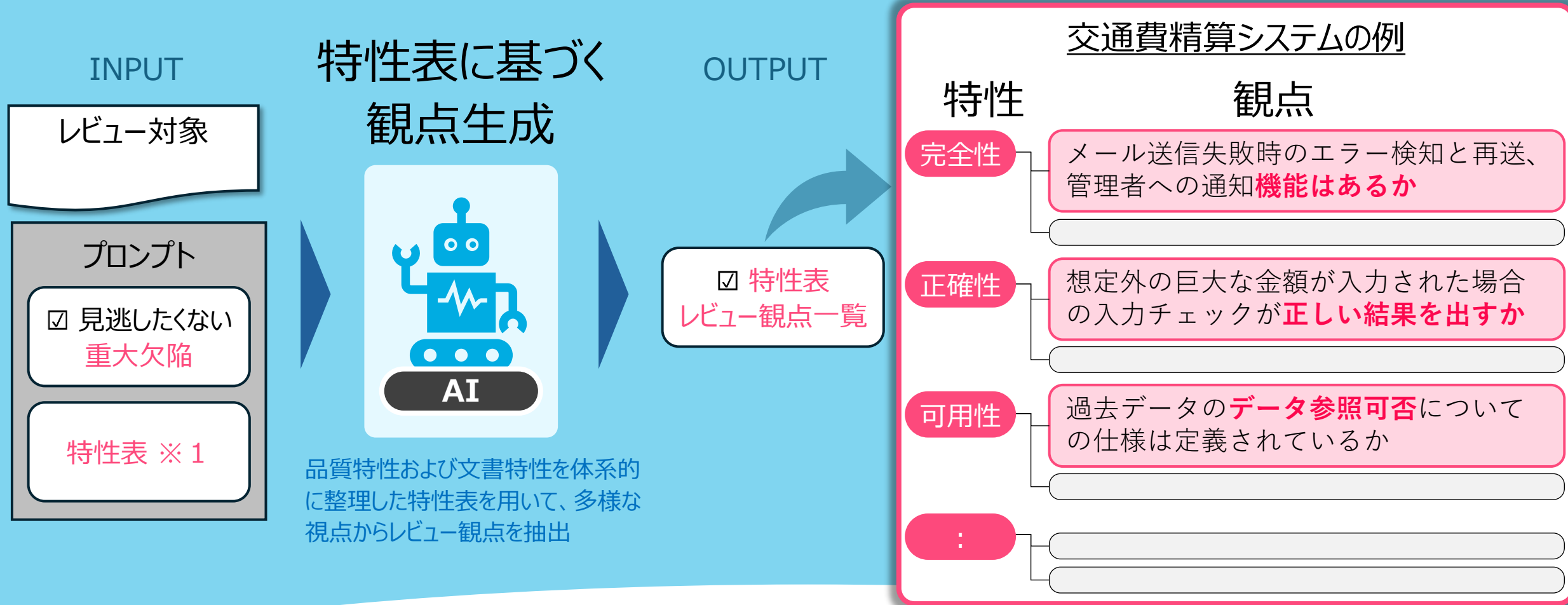
## 特性表に基づく 観点生成



品質特性および文書特性を体系的に整理した特性表を用いて、多様な視点からレビュー観点を抽出

漏れなく幅広い視点から欠陥を見つけるための観点

# 特性表に基づくレビュー観点

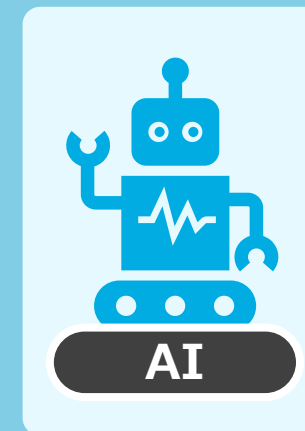


漏れなく幅広い視点から欠陥を見つけるための観点

# ステークホルダー起点のレビュー観点



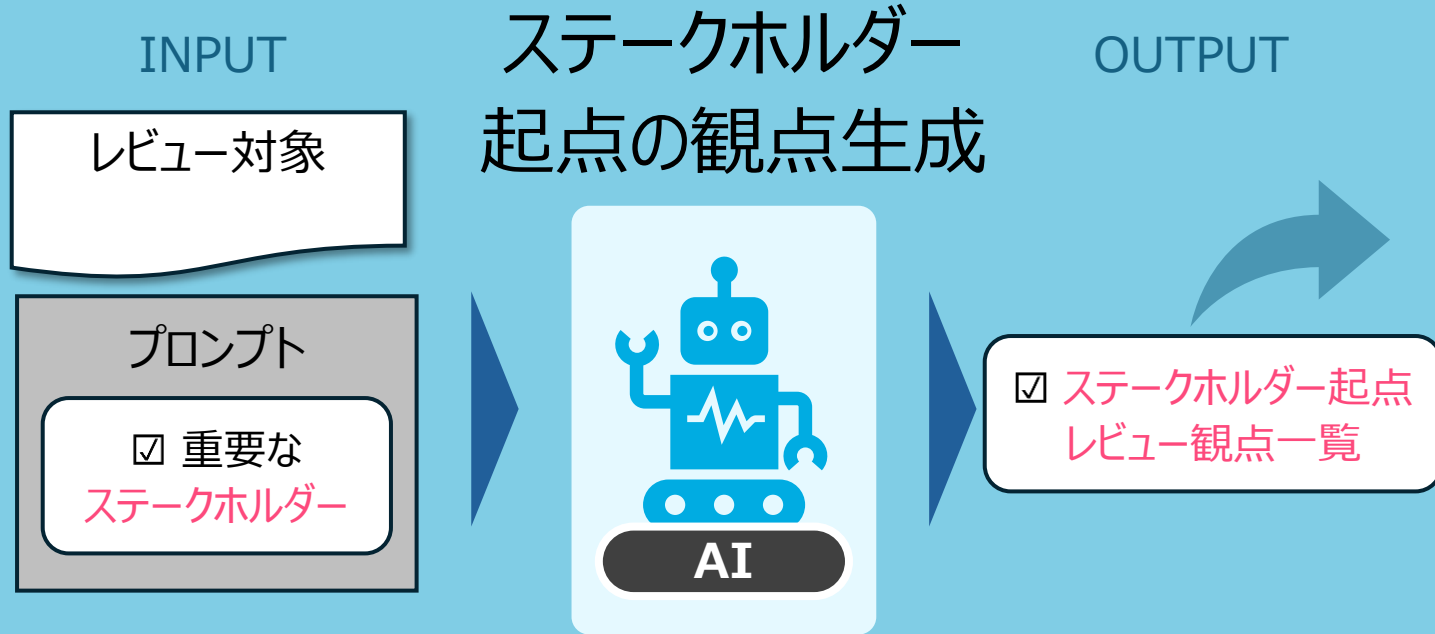
ステークホルダー  
起点の観点生成



各ステークホルダーのアクションおよび  
関心事を整理し、重大欠陥の検出  
につながるレビュー観点を抽出

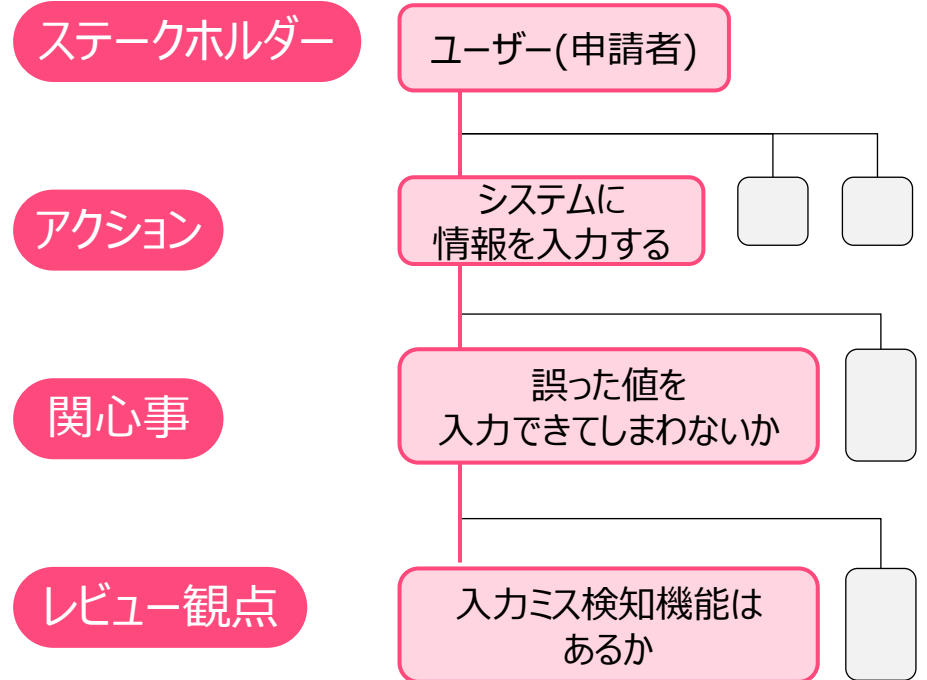
レビュー対象の特性を踏まえた欠陥を見つけるための観点

# ステークホルダー起点のレビュー観点



各ステークホルダーのアクションおよび関心事を整理し、重大欠陥の検出につながるレビュー観点を抽出

## 交通費精算システムの例



レビュー対象の特性を踏まえた欠陥を見つけるための観点

# レビュー観点の統合

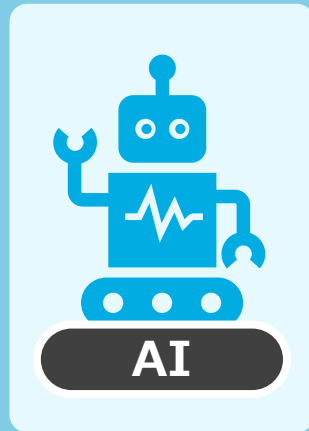
INPUT

観点の統合  
& 優先度付け

OUTPUT

プロンプト

- 特性表  
レビュー観点一覧
- ステークホルダー起点  
レビュー観点一覧



SAKAI  
レビュー観点一覧

発生頻度、影響度、修正コストの  
3つの軸で評価し、総合スコアの  
点数に基づき、優先度を決定

## 交通費精算システムの例

観点	発生頻度	影響度	修正工数	総合スコア	優先度
ログインIDによる自データのみの表示制限（フィルタリング）が設計にあるか。	3	3	3	9	高
データの保持期間及び過去データの検索/表示パフォーマンス要件は定義されているか	2	2	3	7	高 7以上が高
未登録・権限外ユーザへのエラー画面やハンドリング仕様は定義されているか	1	3	2	6	中
...					

重大欠陥につながる可能性の高い観点の優先度を高める

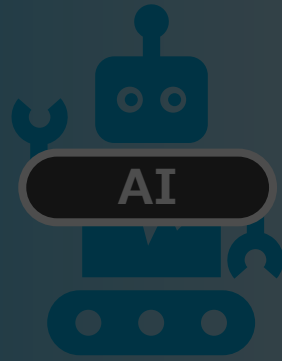
# Phase3 レビュー



レビュアー

● Step0  
**準備**

僅かな準備で



AI

● Step1  
**特性表**に基づく  
レビュー観点

● Step2  
**ステークホルダー**  
起点のレビュー観点

たった数分で

● Step3  
レビュー観点の  
**統合**

● Step4  
**レビュー実施**

有効な観点が

# Step4 レビュー実施

INPUT

レビュー対象

SAKAI  
レビュー観点一覧

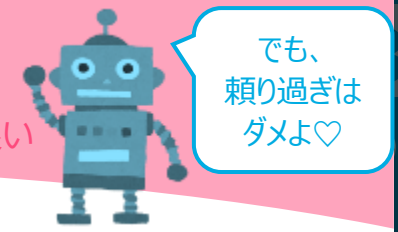
人がレビュー実施



OUTPUT

レビュー指摘

優先度の高いレビュー観点から見ていく  
※どこをどう確認すれば良いのか、生成AIに質問しても良い



優先度の高い観点でレビュー実施

**Step4**  
レビュー実施

有効な  
観点が

スタートホルダー  
起点のレビュー観点

Phase1

準備

Phase2

観点導出

Phase3

レビュー

● Step0

準備

レビュアー

● Step1

特性表に基づく  
レビュー観点

● Step2

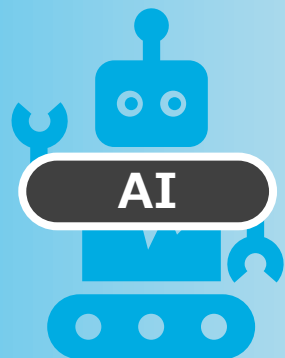
ステークホルダー  
起点のレビュー観点

● Step3

レビュー観点の  
統合

● Step4

レビュー実施



AI



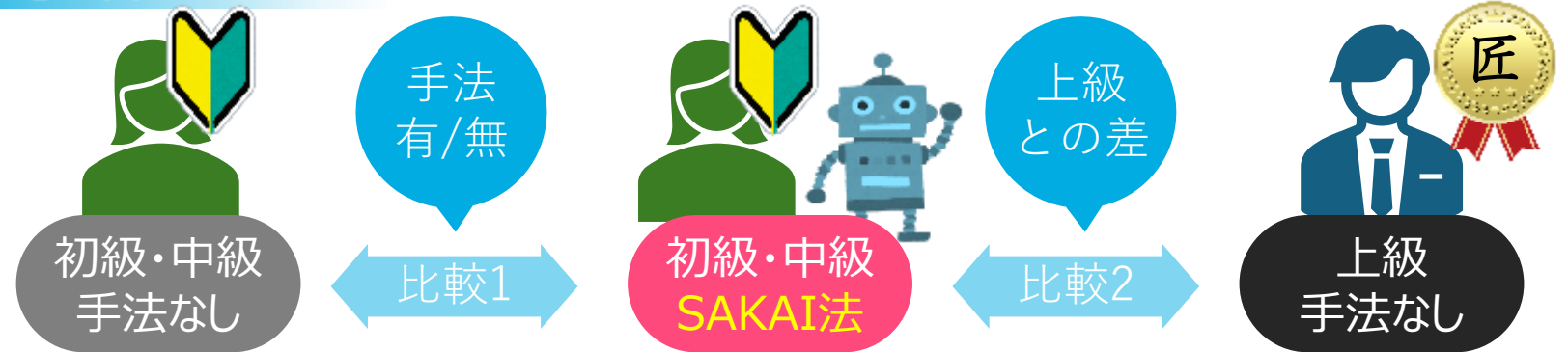
# 3 検証



# 3. 検証

## 2 検証方法

### 検証手順



初級・中級15名 <1回目>		初級・中級15名 <2回目>		上級3名	
グループA	仕様書X	グループA	仕様書Y	<1回目>	仕様書Y
グループB	仕様書Y	グループB	仕様書X	<2回目>	仕様書X
準備	なし	準備	観点導出	準備	なし
レビュー	30分	レビュー	30分	レビュー	30分

必要時間

### 評価方法



**重大欠陥の検出数**が

- ✓ どれほど増えるか？ (比較1)
- ✓ 上級に近づけるか？ (比較2)



**SAKAI法の観点導出**は

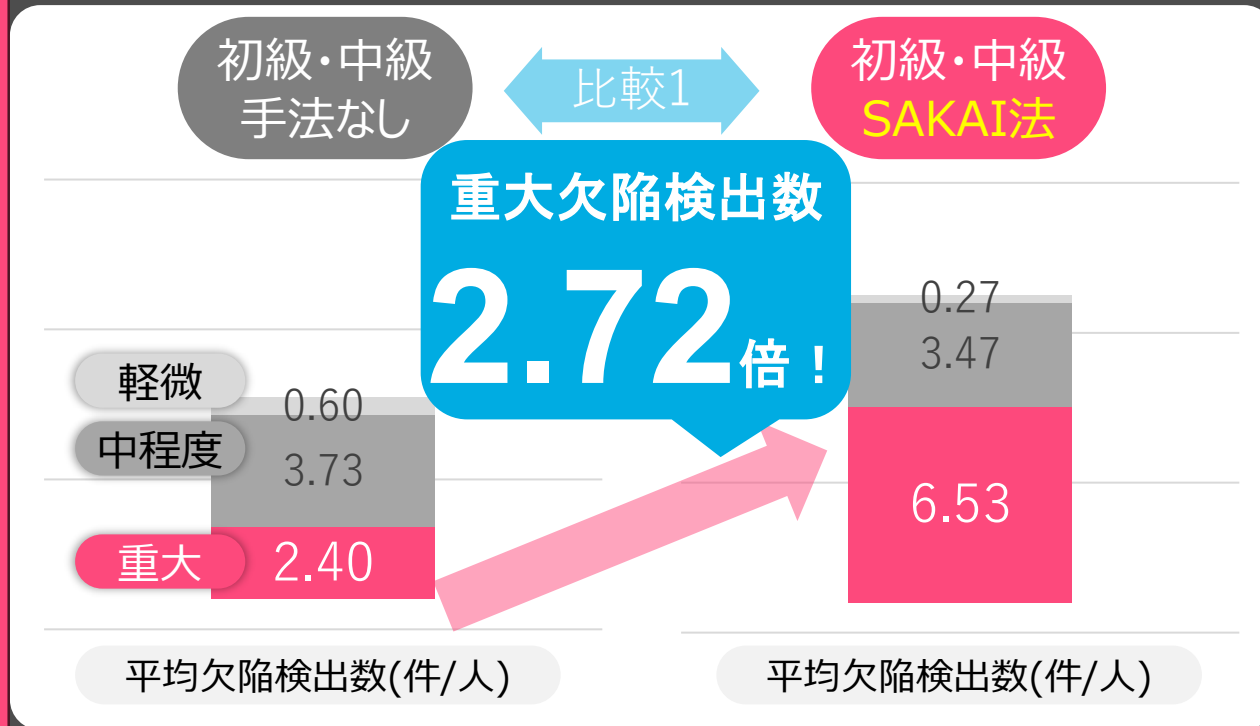
- ✓ 時間効率がどれほど良いか？
- ✓ 現場で運用可能な時間か？

# 3. 検証

## 3 検証結果

### (1) 効果

#### (1) 効果について <比較1>



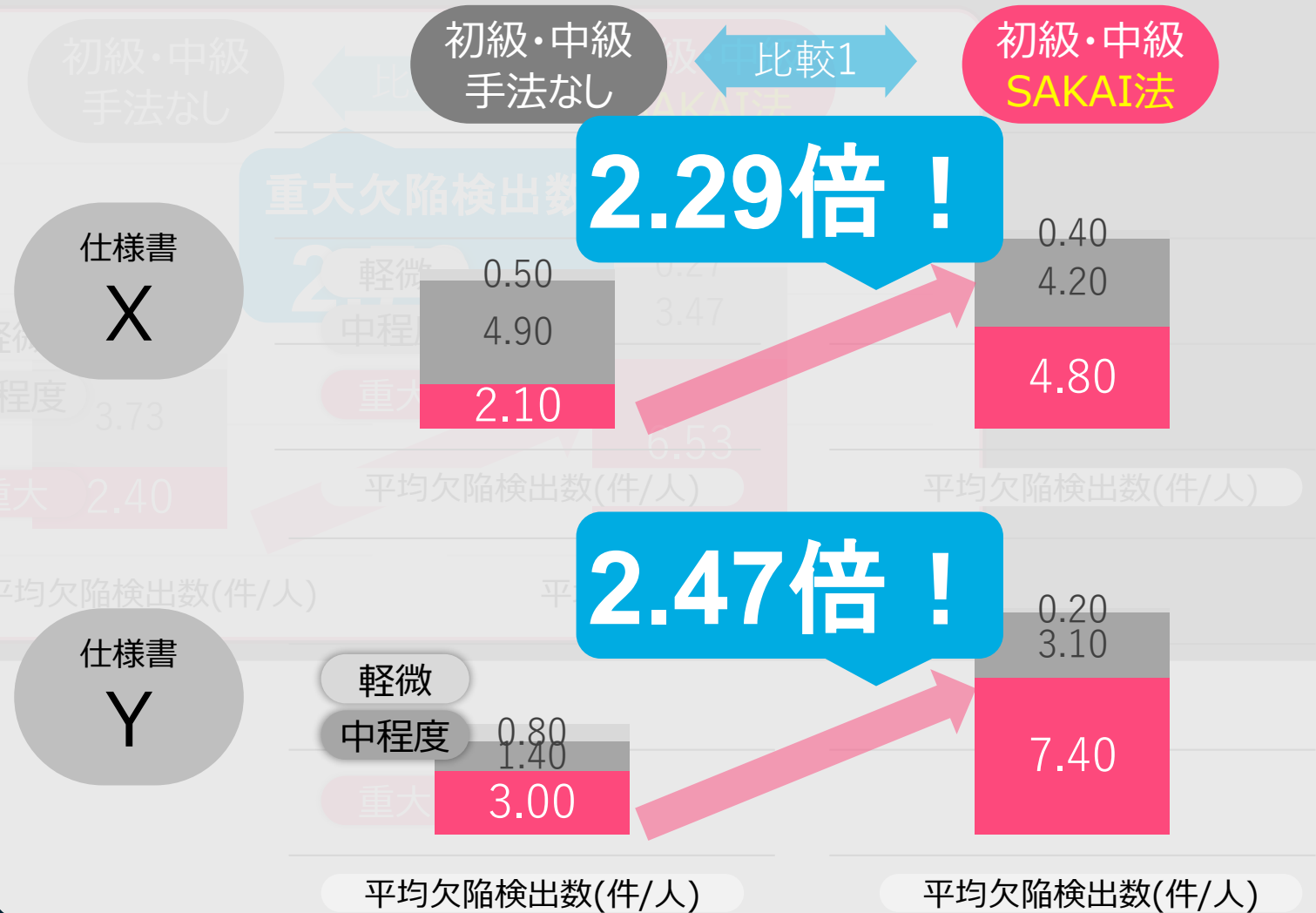
重大欠陥検出数が大幅向上!

# 3. 検証

## 3 検証結果

(1) 効果

(1) 効果について **仕様書別にも、共に増加！**

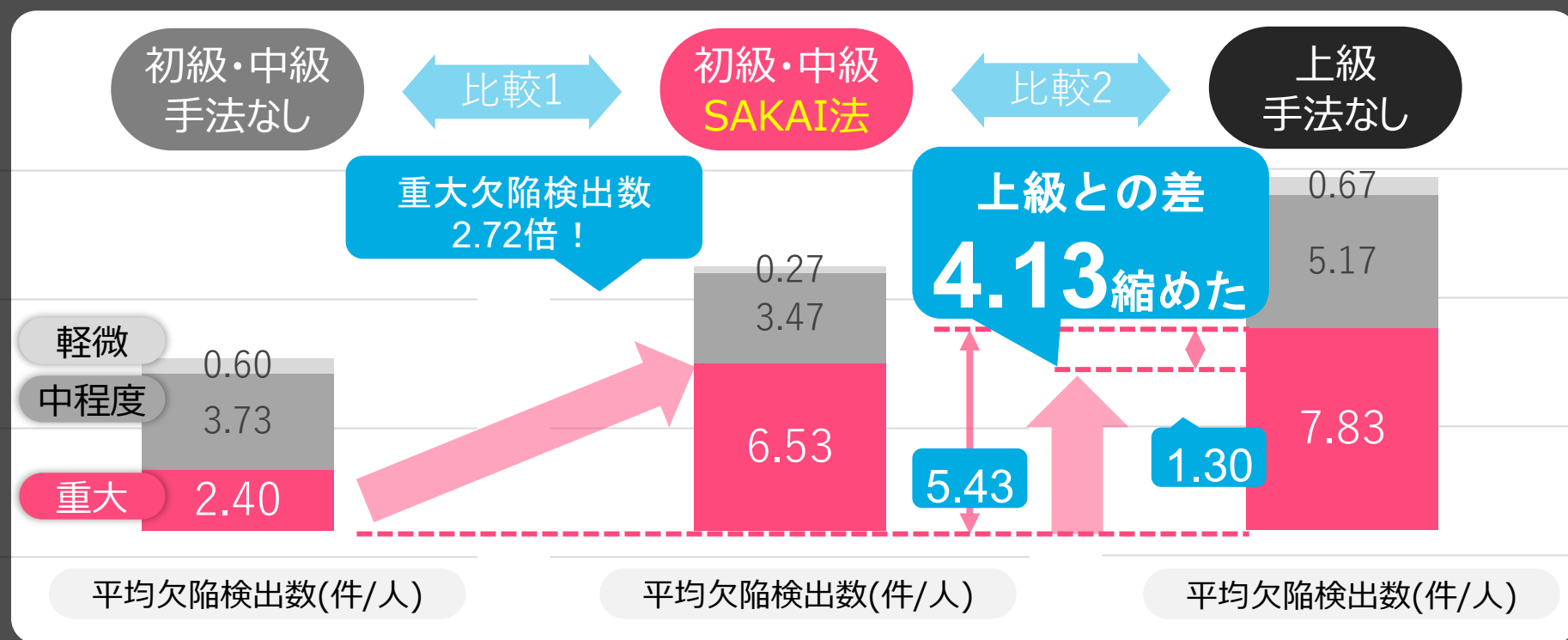


# 3. 検証

## 3 検証結果

### (1) 効果

#### (1) 効果について <比較2>



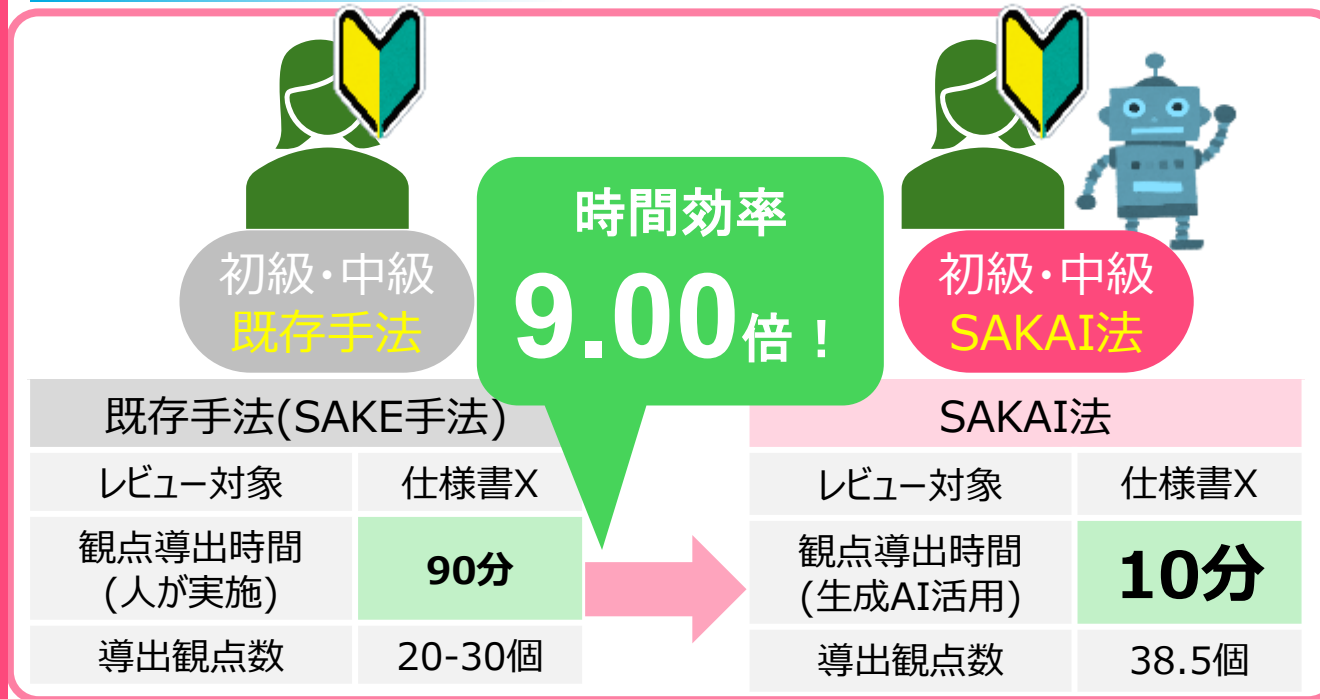
上級との差はあと僅かに！

# 3. 検証

## 3 検証結果

### (2) 効率

(2) 効率について <①既存手法との比較>



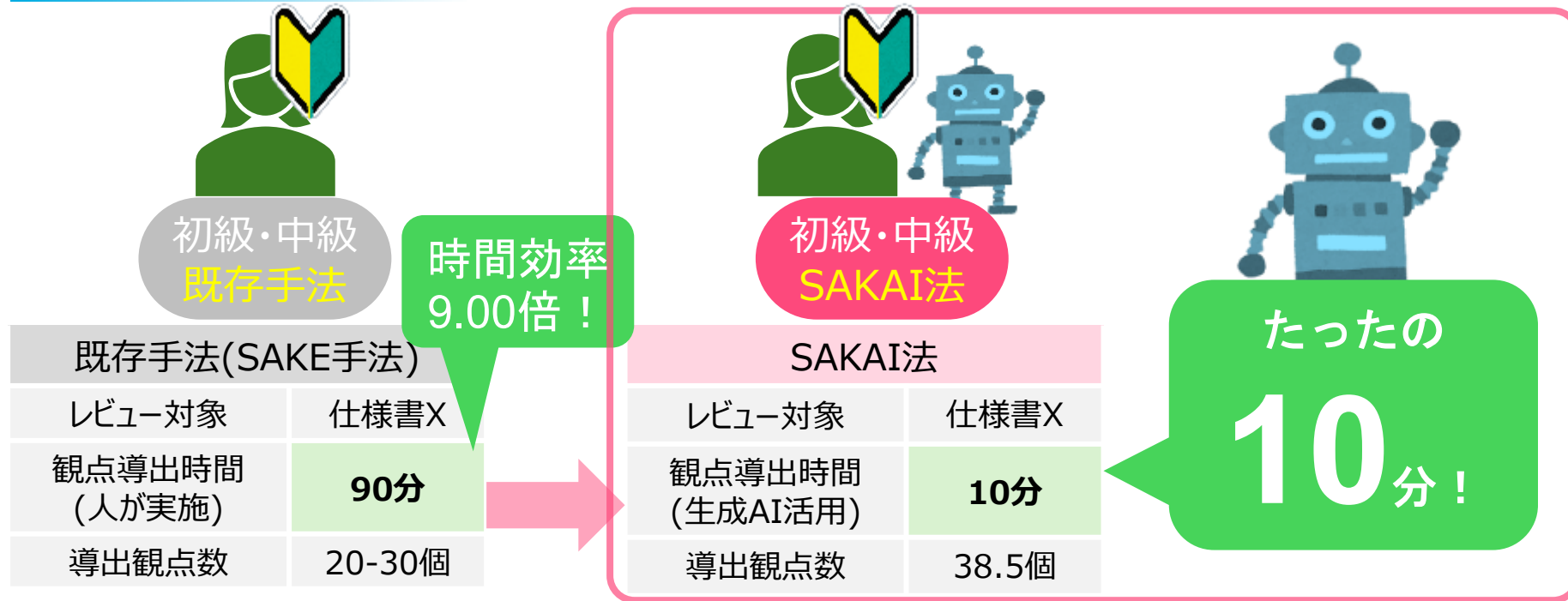
少なくとも見ても時間効率9倍以上！

# 3. 検証

## 3 検証結果

### (2) 効率

#### (2) 効率について <②運用面での評価>



10分なら現場でも運用可能！

# 3. 検証

## 3 検証結果

(3) まとめ

(3) まとめ



**重大欠陥の検出数**が

- ✓ どれほど増えるか？
- ✓ 上級に近づけるか？

**2.72倍！**  
差は**1.3件！**



**SAKAI法の観点導出**は

- ✓ 時間効率がどれほど良いか？
- ✓ 現場で運用可能な時間か？

**9倍以上！**  
たった**10分！**

# 3. 検証

4

気づき

その1)

検証を通して気づいたこと (その1)



## 観点の粒度にばらつきがあった

抽象的な観点例

飛行時間の算出ロジック(天候や準備時間を考慮した余裕率)の妥当



余裕率…?

具体的な観点例

到着時刻・飛行時間の計算精度：距離から飛行時間を自動計算する際、離着陸の準備時間や風向き等を考慮しているか  
単純な「距離÷速度」では到着時刻がずれる。

実運用のロスも考慮された計算式になっているかを確認するのね！



### 現状

観点の抽象度が高すぎて指摘を想起できず、欠陥検出できない人も

### 今後の展望

具体的な例も確実に出力されるようにプロンプトを改善するなど、工夫の余地あり

# 3. 検証

4

気づき

その2)

検証を通して気づいたこと（その2）



**レビュー対象物に即した観点が重要**



機能適合性の例

仕様書Y：交通費精算システム

システム入力の3日後に支払いが行われる仕様において、**銀行休業日**や振込処理の締め時間を考慮した支払日計算ロジックになっているか

仕様書X：ヘリコプター予約システム

燃料費計算において「出発場所～行先の距離」のみを計算対象としているが、待機場所から出発場所への移動（回送）にかかる**燃料費**が考慮されているか

レビュー対象が異なれば、観点も違う  
重要なのはレビュー対象物に即した観点を設定すること

# 4 まとめ



# 4.まとめ

上級と初級・中級には「**観点**」の差があった！  
この「差」をゼロに近づけたい！  
この観点を**短時間**で**誰でも**活用できる形にするために



**SAKAI法** (Stakeholder Action Kanshingo extraction supported by AI) Method  
生成AIを活用したステークホルダー起点のレビュー観点導出手法



レビュー対象の特性に応じた  
**効果的**なやり方

観点導出の方法

レビュー対象の特性に応じた観点を導出  
できる既存研究(SAKE手法)を参考に  
再設計



現場で運用可能なレベルの  
**効率的**なやり方

生成AIの活用

再設計した観点導出のプロセスを生成AI  
を活用することでより現場で回せる形にする



レビュー前の10分

# 最後に、皆さんへ。

## 10分だけください。

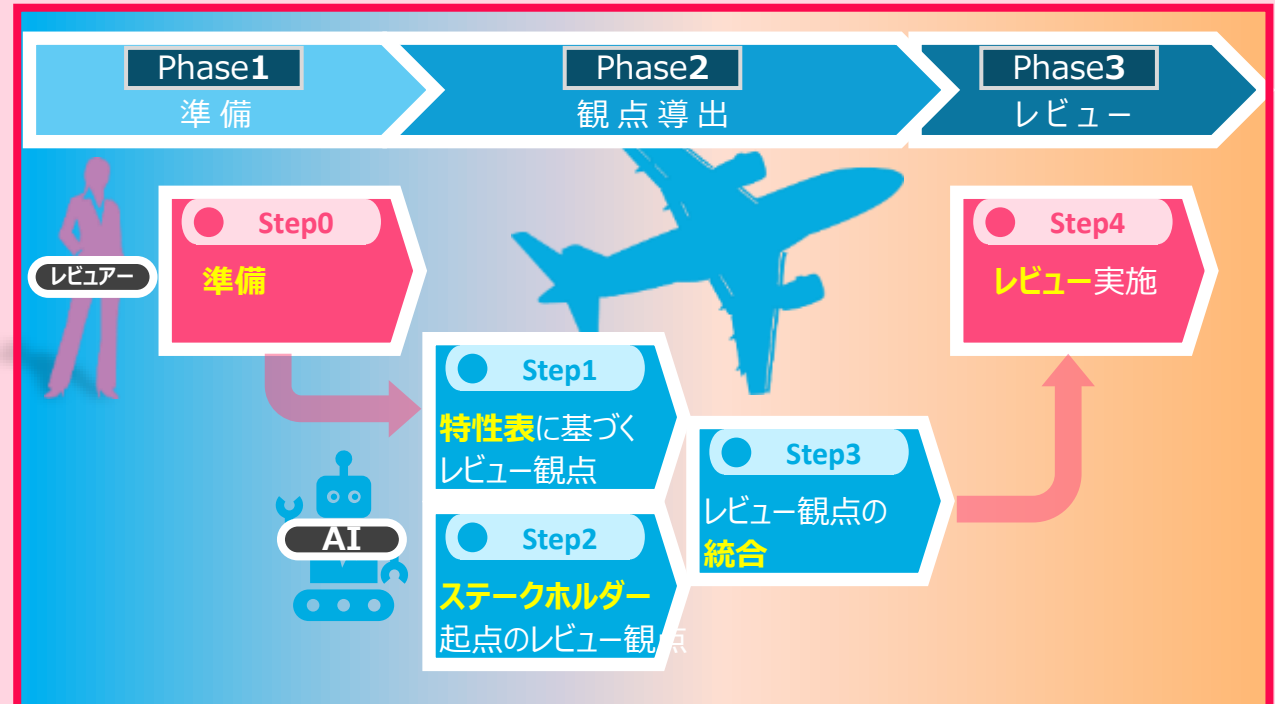
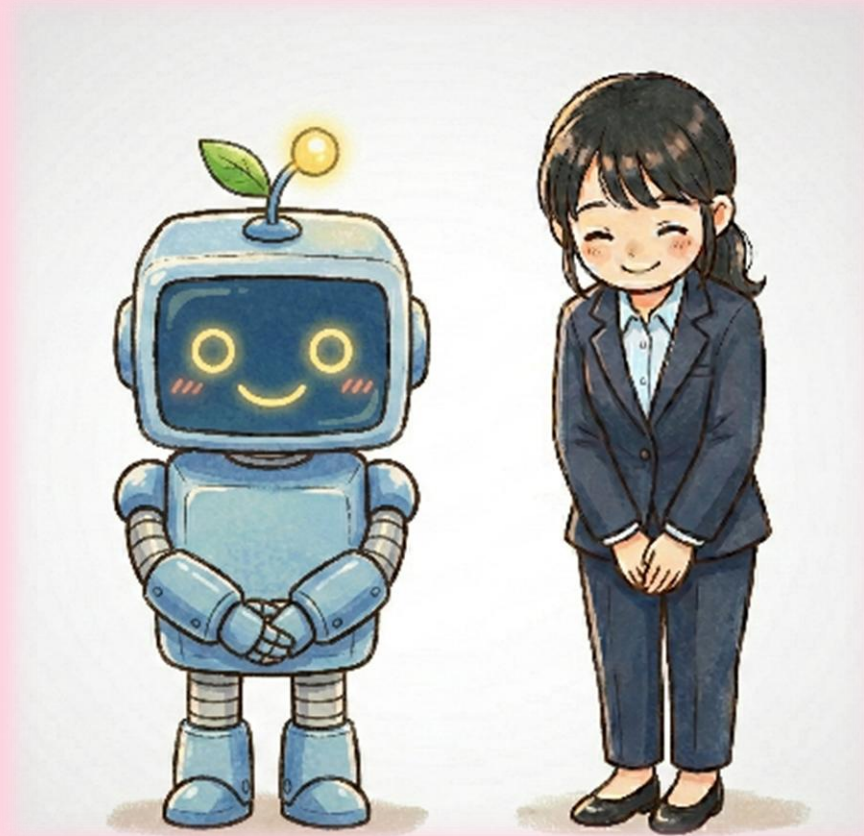
SAKAI法は、上級との間にある「境」を限りなくゼロへ近づけます。  
レビュー前の10分、生成AIにプロンプトを投げる。それだけです。  
重大欠陥の検出数は伸び、新人でもベテランの視点に近づけます。

使わない選択肢はない！ぜひ試してください。



SAKAIをアップデート

ご清聴ありがとうございました



本手法は生成AIに情報をインプットします。  
**機密情報入力が許可された環境**  
で、ご利用ください。