

株式会社 日立システムズ

生成AI活用による 品質評価手法の改善に向けた実践的アプローチ

品質保証本部
中山茜

Date
2025/9/25

Contents

- 1 はじめに
- 2 課題
- 3 実施内容
- 4 実施結果
- 5 おわりに



日立システムズ提供サービス

幅広い分野・業種に対し、業種別の課題解決で培った専門サービスや
セキュリティ・クラウド・ネットワークの運用・監視を支援するマネージドサービスを提供します。
人とITの力を融合したサービスを通じて、社会課題の解決に貢献しています。

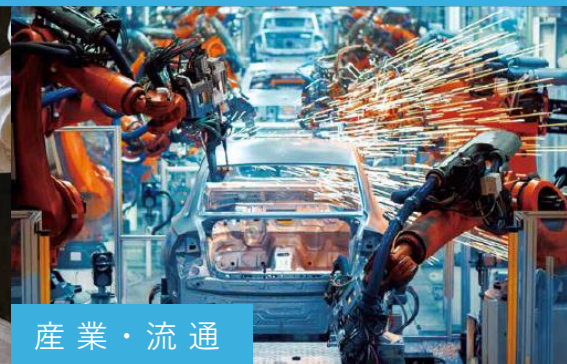
分野・業種別サービス



公共・社会インフラ



金融



産業・流通

マネージドサービス

セキュリティ



クラウド・ネットワーク&データセンター・IT運用



コンタクトセンター&BPO※



※Business Process Outsourcing（業務の外部委託）

サービス拠点(保守・工事)



1. はじめに > 1-2. 品質保証本部の活動

日立システムズ品質保証本部の活動

日立の品質DNAを継承し、常にお客さま第一・品質第一で自らの行動を判断、
お客さまに満足いただける高品質でなおかつ安心・安全な製品・サービスを提供します。

1. 重要事故の撲滅

- ・重要サービス基盤レジリエンス確保
- ・情報セキュリティ対策強化
- ・品質コンプライアンス事故防止



2. 事業拡大および、製品・サービスN倍化に対応した品質確保と既存領域の事故削減

- ・生成AIを活用した品質保証プロセスの確立と品証業務の効率・効果向上
- ・海外グループ会社の品質保証活動の継続的強化



3. 品質保証活動の継続的進化に向けた人財育成と組織活性化

- ・一人一人が活躍できる職場づくり(DEI)
- ・成長領域に向けた品証技術者育成(生成AI等)



品質保証における生成AIの取り組み

QA for AI

生成AIによる開発での品質確保
生成AI製品・サービスの品質確保

- ・ソースコード生成、テスト自動化に対応した品質確保プロセス、確認観点の整備
- ・生成AI案件捕捉によるリスク刈り取り、生成AI関連ガイドラインの適用推進

ナレッジ活用

蓄積した品質評価ノウハウの活用

- ・過去のトラブル事例の活用
- ・多数あるナレッジからのレコメンド

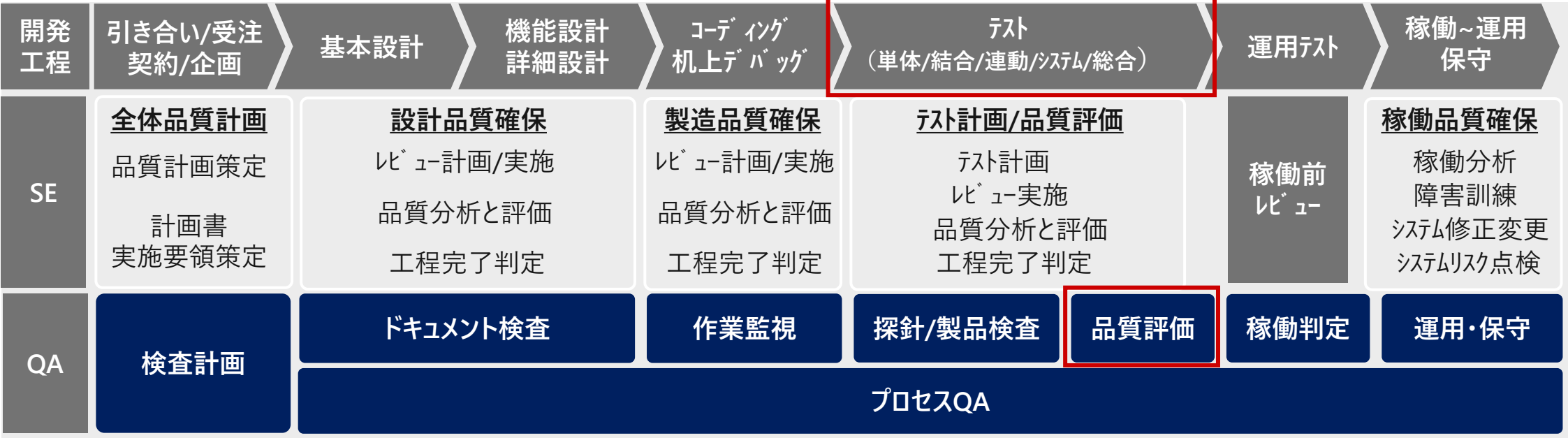
AI for QA

全品質保証業務への生成AI活用による
効率化と精度向上

- ・有効ユースケース適用拡大
- ・新規ユースケース効果検証

1. はじめに > 1-3. 開発工程における品質保証業務の流れ

開発工程における品質保証業務の流れ



品質を作り込み、品質を確保する責任がある

あらかじめ決めたプロセスを守って開発を進めているかどうか、自ら監視する
品質評価の段階では、まず開発者自身による評価・見解を出す

⇒ 自ら問題の対策を図り、品質を作り込む

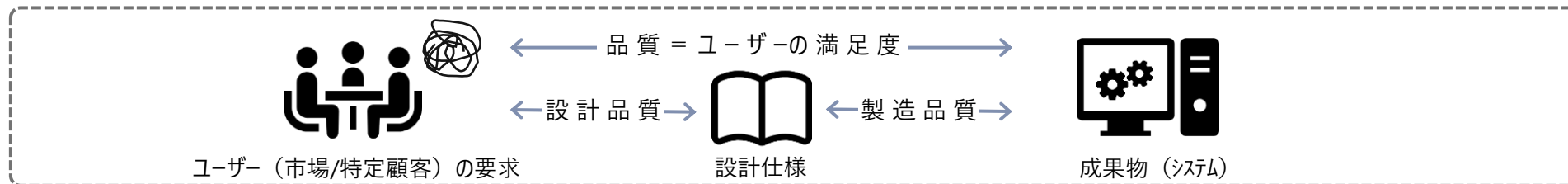


第三者の立場で品質を見極め、最終的な品質を保証する責任を持つ

「検査を実施した」「問題を指摘した」だけではなく、改善策の提案・フォローも行い、稼働できる品質が確保されているかどうかを見極める

テスト工程の品質評価

お客さま要件（業務機能要件/非機能要件）・設計仕様を満足するテストができたか評価することで、問題点のフィードバック（品質向上）に結びつける



品質評価

◆ 定量評価・・・データの数値に基づく評価

- 評価基準（品質指標、目標値）と実測データの比較による評価
- ・目標値に対する達成度評価：目標値に対し、実績値が異常値でないかを評価
 - ・相関評価：実績値を相関で見ることで、テストの十分性等を評価
 - ・時系列評価：テスト工程毎の不良密度を比較（半減の法則）

◆ 定性評価・・・データの内容に基づく評価

バグ票（不具合管理票）分類コードや記載内容を基にした不良の分析による評価
（分類コード例）

- ・現象
- ・原因
- ・要因
- ・作り込み工程
- ・摘出すべき工程 ...

分類コードをもとに不良の傾向を把握することで、製品の弱点を探り品質向上観点を探すことやテスト観点の偏りがなく、テスト内容や十分性を評価することを目的とする。

（コード分析例）

・要因コード別

不良要因「11:単純ミス」「12:思い込み」が摘出されている。

単純ミス：

担当者による偏りがあり、別の要因があるにもかかわらず要因分析不足により見逃している可能性がある。
セルフチェックの十分性や仕様理解度を確認の上、分類コードを再鑑すること。

思い込み：

初期化処理の処理位置の誤り、メッセージ変数の設定誤り。

要因コード「15:影響調査不足」で画面追加項目の再表示時における編集仕様の考慮漏れも摘出されており、設計書記載不足や設計時の考慮不足による不良残存が懸念されるため、根本原因を分析の上、設計書の追記や、設計書通り開発できているか再鑑する等の対策を検討すること。
不要の場合、不要理由を回答すること。

要因コード	11:単純ミス	12:思い込み	13:仕様理解不足 (複雑で想定できず)	15:影響調査不足	16:共通モジュール 理解不足	総計
担当者X	6	12	6			24
担当者Y	24			6		30
担当者Z	12				6	18
総計	42	12	6	6	6	72



品質評価における課題

前提



担当している産業分野は、
中小規模かつ短サイクルの開発案件が多く、
短期間で複数案件の品質評価が求められる

課題



過去案件の品質評価における
ノウハウ（懸念事項/品質向上観点等）が
十分に活用されていないことによる、
担当者毎の品質評価の**精度差**および**工数過多**

対策

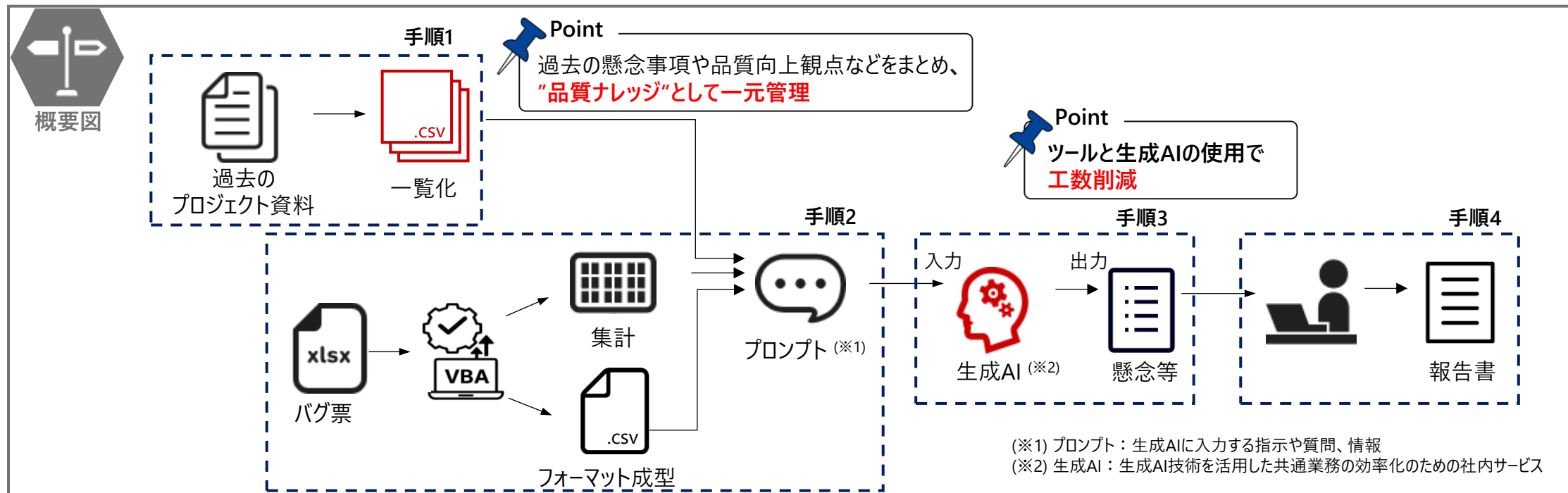


- ・過去の品質評価ノウハウを集約し、**一元管理**
- ・過去のデータや経験から多彩なパターンの問題に対する
解決策を**迅速**に見つけることができる**生成AI**を導入。
LLMの**文脈理解**と**推論力**により、曖昧な表現を含む
バグ票の分類や品質課題の分析を**効率的**に実施。

》》》 **生成AIを活用し、品質評価の精度確保、および工数削減を実現する**



実施概要



手順

- 1：過去プロジェクトの資料から懸念事項や品質向上観点などの品質評価ノウハウを収集し、**一覧形式**にまとめる
- 2：手順1で一覧化した資料とバグ票を利用し、バグ票のフォーマット成型とプロンプト作成を行う
- 3：手順2で作成したプロンプトを**生成AI**に入力し、評価対象の不良傾向に沿った懸念事項や品質向上観点を出力する
- 4：ハルシネーションの対策として、生成AIの出力事項を手手で精査する

手順1：過去プロジェクトの資料から懸念事項や品質向上観点などの品質評価ノウハウを収集し、一覧形式にまとめる

1-1. 過去PJの品質評価報告書やガイドラインから「懸念事項」や「品質向上観点」を収集

(品質評価報告書)

(2) B 票コード分析

...

c) 要因コード別

指摘 7：不良要因「思い込み」について、設計書記載不足による不良残存が懸念されるため、根本原因を分析し、必要に応じて対策を実施すること。

不良要因「11:単純ミス」「12:思い込み」が抽出されている。

単純ミス：設計書に記載されているがプログラム時の記載誤りや設定漏れであり、本工程抽出妥当。担当者による偏りもなく、要因コードの傾向として問題はない。

思い込み：初期化処理の処理位置の誤り、メッセージ変数の設定誤り。

「設計書を正確に見ずに作成した」、「設計書の記載粒度が粗く、認識に差が出た」等、不良残存が懸念されるため、

根本原因を分析の上、設計書の追記や、設計書通り開発できているか再鑑する等の対策が必要ないか検討すること。

要因コード：思い込み

「設計書を正確に見ずに作成した」、「設計書の記載粒度が粗く、認識に差が出た」等、不良残存が懸念されるため、根本原因を分析の上、設計書の追記や、設計書通り開発できているか再鑑する等の対策が必要ないか検討すること。

(品質評価ガイドライン)

8. バグコード分析				
現象コード	現象コード	現象コード	現象コード	現象コード
8.1	B票	現象コード	・ABENDが少ない ・異常終了系のバグの残存 ・プログラムの動作確認が不十分	・チェック観点見直し ・異常系テスト項目のCL追加
8.2		現象コード	・ABEND未収束 ・ABEND系以外のバグの残存 ・ABENDが収束していないと、テスト中であるが、異常終了している	・チェック観点見直し ・ABENDが収束していない ・通常CTやATではテスト終了になるにつれてABENDが収束していくため、
8.3		現象コード		
8.4		現象コード		

現象コード：ABEND（異常終了）が少ない

結合テストや連動テストではモジュール間IFなどの問題で、異常終了系の障害が発生しやすい。例えば結合テストでのABENDがゼロ、単体テストでも少ない場合は異常系テスト項目の追加や、テストデータ含めテスト内容の見極めが必要

1-2. 収集した情報を一覧形式にまとめ、「品質ナレッジ」を作成

工程,コード区分,コード名称,見解,対策

要因コード,13:仕様理解不足(複雑で想定できず),「設計時・開発時の確認プロセスが確立されていない」等の理由が考えられ、仕様理解不足による不良残存が懸念される,担当者間で差異はないか確認すること、担当者に対する業務仕様知識の周知や教育、設計書是正や追加テストの必要性はないかを確認すること。

要因コード,14:仕様理解不足(曖昧なまま確認せず),担当者の技術不足や設計書不備など処理の意図を理解仕切れないまま開発したことによる不良残存や開発体制の不備が懸念される,担当者間で差異はないか、設計書不備によるドキュメント修正の必要がないか確認すること。また、レビュー頻度や体制に問題がないか確認し、是正策を検討すること。

要因コード,15:影響調査不足,

要因コード,16:共通モジュール理解不足,担当者が部品となる共通モジュールの使用方法を理解しておらず、共通モジュールを使用する他の

Point

生成AIのInputとして表形式では文脈や意味のつながりが読み取りにくいいため
項目の区切りが明確で構造がシンプルなカンマ区切りの形式で一覧化



工程,コード区分,コード名称,見解,対策の順に記載

要因コード,

13:仕様理解不足(複雑で想定できず),

設計時・開発時の確認プロセスが確立されていない」等の理由が考えられ、仕様理解不足による不良残存が懸念される,

担当者間で差異はないか確認すること、担当者に対する業務仕様知識の周知や教育、設計書是正や追加テストの必要性はないかを確認すること。

手順2：手順1で一覧化した資料とバグ票を利用し、バグ票のフォーマット成型とプロンプト作成を行う

2-1.
ツール実行し、バグ票のCSVフォーマットへの変換と集計を実施する2-2.
ツール実行し、プロンプトを出力する

入力パラメータ

ファイルパス	C:\Users\%LDAP%\Documents\%サンプル%\サンプルB票.xlsx
シート名	記録
先頭行	7
現象コード列	F
原因コード列	O
要因コード列	P

変換・集計

Excel形式のバグ票をcsvに変換
→不良内容の要約に使用



バグ票を集計し、表形式で出力
→集計結果の可視化

現象コード	件数
22:一部停止 (不可避・再開不能)	1
31:画面編集・出力不正	18
...	...
総計	30



Point

生成AIでの集計は不正確であった
生成AIのInputとなる集計結果の正確性確保のため、
ツールでバグ票の分類コード別件数を集計

集計区分 現象コード

工程 UT (単体テスト)

出力件数 5

プロンプト作成



Point

**期待どおりの出力を得るため、役割/背景情報/
具体的な手順/出力形式を明示**

各種プロンプトの作成

- ・入力プロンプト
手順2-1で出力したB票集計結果を
文章化し、グルーピングする
- ・システムプロンプト
品質評価手順や制約事項、
“品質ナレッジ”やバグ票記載内容を含む

入力プロンプト

以下の情報を基に、品質評価を行ってください。

#集計

31:画面編集・出力不正：18件
37:メッセージ不正：7件
36:ログ参照・編集・出力不正：6件
34:ファイル編集・出力不正：5件
99:その他：4件

システムプロンプト

##役割

あなたはIT企業の品質保証エンジニアです。
今回、あなたはUT（単体テスト）の品質評価を行うことになりました。
以下に従い、品質評価を行ってください。

##制約事項

・全体プロセスはRP（要件定義）、B1（基本設計）、AD（詳細設計）、P（開発）、
UT（単体テスト）、CT（結合テスト）、ST（システムテスト）、AT（連動テスト）の順番
で実施される

・評価対象の工程はUT（単体テスト）である。

・評価対象に沿った見解を出力する必要がある。
例えば、評価対象がUT（単体テスト）の場合、「前工程でのテスト」は存在しないため、
該当する見解の出力は不要。

##評価手順

手順1.集計結果の要約

手順2.コード別分析

以下コード区分毎の評価内容に従い、コード別分析を行う。

評価内容は、コード区分、コード名称、見解の順に記載しています。ただし、以下見解が提供
されていない場合、コード名称や不良内容から懸念点および対策を考えること。

コード区分、コード名称、見解、対策 ...

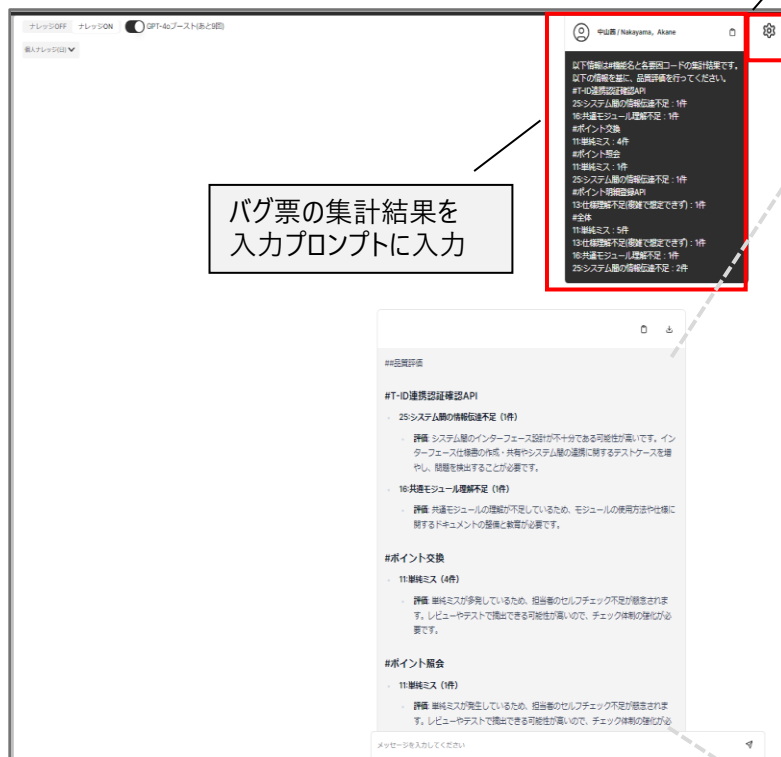
手順3：手順2で作成したプロンプトを生成AIに入力し、評価対象の不良傾向に沿った懸念事項や品質向上観点を出力する

評価手順や“品質ナレッジ”を
システムプロンプトに入力



Point

AzureOpenAIを活用して生成AIへの入力・出力を自動化
ツール操作のみで実行できるため、工数削減



生成AI 出力結果を一部抜粋

判定処理不良（原因コード: 04）

①B票コード分析

- ****懸念点**:** 機能の特性を確認すること。
例えば、複数のエラー処理を実装しており、エラーチェックの判定パターンに関するテストが多い場合は、判定処理不良が多い傾向は問題ない。ただし同様のエラー処理を複数実装している場合、類似不良が発生する懸念がある。

- ****対策**:** 設計書やCLを再確認しテストの十分性を確認、またエビデンスの再鑑により不良見逃しがないか確認すること。

懸念事項

②摘出不良の傾向分析

04:判定処理不良

- ****手順1.指摘のリストアップ**:**

- ○○国の入力チェック項目が△△国になっていた。
- ◇◇国の入力不可チェックの不具合。

- ****手順2.類似性の高い不良の要約**:**

- 入力チェック項目が誤って設定されているため、正しい判定が行われていない。

- ****手順3.問題点**:**

- 入力チェックの対象項目の設定ミスにより、誤った判定が行われている。

- ****手順4.対策**:**

- 入力チェック項目の設定を設計書と一致しているか確認する、正しい項目が設定されているか確認するテストケースを追加する。

品質向上観点

手順4：ハルシネーションの対策として、生成AIの出力事項を人手で精査する



生成AI出力結果

04:判定処理不良	<p>②抽出不良の傾向分析</p> <p>04:判定処理不良</p> <p>- **手順1: 指標のリストアップ**:</p> <ul style="list-style-type: none"> - ○○国の入力チェック項目が△△国になっていた。 - ◇◇国の入力不可チェックの不具合。 <p>- **手順2: 類似性の高い不良の要約**:</p> <ul style="list-style-type: none"> - 入力チェック項目が誤って設定されているため、正しい判定が行われていない。 <p>- **手順3: 問題点**:</p> <ul style="list-style-type: none"> - 入力チェックの設定ミスにより、誤った判定が行われている。 <p>- **手順4: 対策**:</p> <ul style="list-style-type: none"> - 入力チェック項目の設定を見直し、正しい項目が設定されているか確認するテストケースを追加する。
05:編集処理不良	<p>①B票コード分析</p> <p>05:編集処理不良</p> <p>- **見解**:</p> <p>- **懸念点**:</p> <p>- **対策**:</p> <p>編集処理に関する設計書やコードを再確認し、データの整合性やフォーマットに関する問題がないか確認すること。また、類似の編集処理が他の部分にも存在する場合、同様の不良がないか確認すること。</p>

出力事項が正しいか、必要な指摘事項が確認

- ①生成AI出力と、バグ票や集計表を比較し、**データの整合性や論理性を確認**
- ②現行どおりレビューを実施、**レビューが妥当性を確認**



手順2で作成した集計表を添付

現象コード	件数
22:一部停止 (不可避・再開不能)	6
31:画面編集・出力不正	6
...	...
総計	30

報告書

(2) B票コード分析

b) 原因コード別

指摘X：不良原因「判定処理不良」について担当者の偏りがみられ、類似不良の残存が懸念されるため、根本原因を分析し、必要に応じて対策を実施すること。

指摘Y：不良原因「編集処理不良」について、類似不良の残存が懸念されるため、影響範囲を確認し類似不良がないか再確認すること。

不良原因は「04:判定処理不良」「19:メッセージ不良」「05:編集処理不良」が抽出されている。

判定処理不良：主な不良は、入力チェック用パラメータに設定する項目の取り違えにより適切なエラー判定ができない。

同一の担当者で抽出されており、類似不良残存の懸念があるため、再テストの実施やエディタの再確認等対策を検討すること。

対策不要とする場合、不要理由を回答すること。

メッセージ不良：メッセージ内の変数の設定誤りであり、本工程抽出妥当。機能改修により追加したメッセージについては全件テスト済みであり問題ない。

編集処理不良：画面追加項目の再表示時における編集仕様の考慮漏れ、引数の設定誤りによる不良。

データの整合性やフォーマットに関する問題が他の機能にも影響を及ぼす懸念がある。

編集処理に関する設計書やコードを再確認し、データの整合性やフォーマットに関する問題がないか確認し、類似不良がないか再確認すること。

原因コード	01:初期化漏れ・初期設定不良	04:判定処理不良	05:編集処理不良	11:画面レイアウト・項目編集不良	19:メッセージ不良	総計
機能A		6	4		5	10
機能B				3		3
機能C			3			3
機能D	2		2		3	7
機能E	2					2
総計	4	6	9	3	8	30

c) 要因コード別

指摘Z：不良要因「思い込み」について、設計書記載不足による不良残存が懸念されるため、根本原因を分析し、必要に応じて対策を実施すること。

不良要因「11:単純ミス」「12:思い込み」が抽出されている。

単純ミス：設計書に記載されているがプログラム時の記載誤りや設定漏れであり、本工程抽出妥当。担当者による偏りもなく、要因コードの傾向として問題はない。

思い込み：初期化処理の処理位置の誤り、メッセージ変数の設定誤り。

「設計書を正確に見ずに作成した」、「設計書の記載粒度が粗く、認識に差が出た」等、不良残存が懸念されるため、根本原因を分析の上、設計書の追記や、設計書通り開発できているか再確認等の対策が必要ないか検討すること。

不要の場合、不要理由を回答すること。

要因コード	11:単純ミス	12:思い込み	13:仕様理解不足 (複雑で想定できず)	15:影響調査不足	16:共通モジュール 理解不足	総計
機能A	7			3		10
機能B	6					6
機能C	3					3
機能D	2	5				7
機能E			2			2
総計	18	5	2	3		28



適用効果

1

精度確保

生成AIに過去プロジェクトの懸念事項や品質向上観点を取り込むことで、
担当者が評価した結果と**同等の懸念事項が出力**されることに加え、
“品質ナレッジ”で**提供していない懸念事項**を生成AIにより出力が可能

(懸念事項の件数比較)

	A案件	B案件	C案件
現行	2件	4件	3件
生成AI利用	6件	6件	6件



出力例

“仕様理解不足”の不良が多く摘出されていることに対し、
「担当者の技術不足や設計書不備による不具合残存の懸念」に加え、
「レビュー頻度や担当者間の差異など開発の不備」を懸念事項として生成AIが出力。
また、仕様理解のための勉強会開催や、レビュー体制の強化、変更管理方法の周知等、
具体的な対策例の出力が可能。

均質化！

工数削減

2

生成AIとツールの使用により、
摘出不良の現象・原因・要因の傾向把握にかかる時間が**7 割減**

- ◆生成AI：過去のデータから、入力データに沿った情報を迅速に出力
- ◆ツール：資料の成型やプロンプト作成、生成AIへの入出力を一括で処理

(生成AI利用前後の工数比較)

	資料収集・集計	不良傾向確認	見解考案	総計
現行	0.5h	6.0h	1.5h	8.0h
生成AI利用	0.5h	1.0h	1.0h	2.5h

削減！



品質評価担当者による定性評価の精度や工数の偏りを低減することが可能
工数削減によって得られた時間をさらなる品質向上のための分析に充てることが可能

工夫点

プロンプト設計



以下情報を含むプロンプト設計を実施。

- ・「品質保証エンジニアとしての**役割**」を明示して品質に関連する知識や視点を出力に反映。
- ・「評価対象・制約事項などの**背景情報**」や「タスクを分けた**具体的な手順**」を含めることで、分析範囲や作業フローを明確にし、分析時間を短縮。

⇒ 期待に沿った出力の実現と
分析時間短縮

1

プロンプト共通化



生成AI活用における課題であった「プロンプト作成者による出力のばらつき」を解消するため、**共通のプロンプトをツールで自動生成**。共通プロンプト生成に加え、不良の分類コードに基づく作業結果とバグ票のフォーマット成形を一つのマクロツールで同時に実行。

⇒ 出力の安定と効率化

2

人手で確認する手順の追加



生成AI活用により多角的に懸念事項や品質向上観点を出力可能。ただし、**ハルシネーション**が起きる可能性があるため、「出力事項に誤りがないか」、「必要な観点か」を**人手**で確認する手順を追加。

⇒ 誤った情報の反映を防ぎ、
精度を担保

3



成果と今後の方針



成果

✓ 担当者による品質評価精度の偏りを低減

生成AIを利用し、バグ票コードの傾向に沿った懸念事項等を導出することで、品質評価担当者による定性評価の精度の偏りを低減することが可能

✓ コード分析/報告書作成の工数削減

生成AIとツールの使用により、摘出不良の現象・原因・要因の傾向把握にかかる工数の削減が可能



今後の方針

当社が管理するガイドラインやチェックリストなどの豊富なナレッジを生成AIに登録し、汎用性と精度の向上をめざす。ナレッジは膨大であり、現時点でRAGとして登録できる資料には制限があるが、生成AIの進化に合わせて運用方法（データ量・フォーマット等）を調整し、活用を図る。

✓ “品質ナレッジ”の蓄積継続と他業種への展開

他業種で管理されているデータ（懸念事項や品質向上観点の一覧）を収集、蓄積。当社が手掛ける全案件に対し、一元的に評価することを目指す。

✓ チェックリスト作成や不具合数の妥当性確認に生成AI使用

一定の指標値で定量的な評価手法を実施しているが、設計者やチェック観点を生成AIに取り込むことで、一定の指標値での評価だけでなく、妥当な件数を導出する。

Tips

表形式データでは文脈が伝わりにくい

生成AIは表形式のデータでは文脈を正しく理解できず、期待通りの出力が得られないことがある。
そのため、項目の区切りが明確なCSV形式に変換することで、意味のつながりを保ったまま生成AIに情報を渡すことができる。
こうした前処理により、社内に蓄積された業務データをAIが正しく理解し、分析の精度が安定する。

生成AIによる集計の不正確性

生成AIによるバグ票の件数集計は不正確な結果となることがある。
そこで、集計はツールで実施することで、生成AIのInputとなる集計結果の正確性を確保。
生成AIには分析と提案に特化させることで、精度と効率を両立。

プロンプト設計のばらつきによる出力の不安定性

プロンプトの設計者によって出力結果に差が出るという課題に対し、役割や背景、手順、出力形式を明示した共通プロンプトを自動生成。
属人性を排除しながら、誰が使っても安定した出力が得られる仕組みとした。

出力に誤情報が含まれる可能性（ハルシネーション）

生成AIは事実と異なる情報を出力することがある。
出力結果を手で精査し、バグ票や集計表と照合することで整合性・論理性を確認し、レビューを実施することで妥当性もチェック。
最終判断は人が行う運用にすることで、分析結果の精度を担保。

HITACHI