ソフトウェアレビューにおける生成 AI 活用の研究

～ChatGPT が欠陥検出と指摘伝達をアシスト～

研 究 員：北里　竜　（ブライシス株式会社）
片桐　汐駿（アズビル株式会社）
馬場　大輔（株式会社オージス総研）
星野　智彦（株式会社アイシン）
主　査　：中谷　一樹（TIS 株式会社）
副 主 査：上田　裕之（株式会社 DTS インサイト）
アドバイザー：安達　賢二（株式会社 HBA）

1.研究概要

本研究では，生成 AI を活用することで,レビューアのスキルに依存せずにレビューの質を向上させる手法を提案する.SQiP 研究会の過去研究成果の中から,生成 AI と親和性の高い重大欠陥検出および指摘事項伝達に関する 2 つの手法を選定し,それぞれの手法に対して生成 AI を活用した手法を考案した.実験によって,レビューの質が向上したことを示す一定の成果を得ることができた.

2.はじめに

ソフトウェア開発の現場では,手戻り防止や品質向上のためにレビューが重要なプロセスであると認識されているにも関わらず,本来検出すべき欠陥をレビューアが見逃してしまうことや,たとえ検出できても欠陥指摘の仕方が悪いため,作成者が素直に受け入れず指摘が反映されないことも少なくない.これは,レビューが属人性の高いプロセスであり,「レビューのスキル依存」の問題が常に存在するためである.

そこで我々は,最近注目を浴びている生成 AI を活用することにより,この問題を解決する可能性を探ることにした.具体的には,生成 AI の一種である,ChatGPT を代表とする大規模言語モデル(LLM)の強みを活かし,レビューアをアシストする.それによって,スキル依存性を下げ,適用手法で期待されている効果を,誰でも出せるようになることを目指した.

本論文の構成を次に示す.3 章では本研究で扱う課題および課題解決のため我々が提案する手法,4 章では手法を実適用した実験内容と結果および考察,5 章以降ではまとめおよび今後の展望について述べる.

3.課題

3.1.課題設定

レビューのスキル依存はどの現場でも発生し得る課題であり,これまでも様々な工夫や研究が行われてきた.SQiP 研究会においても欠陥検出や指摘伝達の質を高めるための手法が考案されており,開発現場へ導入すれば高い効果が期待できるが,これらの手法を適用する場面においても,例えば要約力や言葉の表現力などといった別の新たなスキルが要求される.これらのスキルの根本にあるものは人の自然言語能力であり,それを習得するまでに時間がかかる上,実践するのが簡単ではないという課題がある.

今回研究するにあたり,我々がレビューで目指すゴールは以下の 2 点とした.

- 重大な/指摘すべき(リスクの許容範囲を超える)欠陥の検出
- 欠陥の確実な修正

SQiP 研究会における過去研究の内,設定されたゴールとの関連性が高く,手法の適用時

に必要なスキルが生成 AI によって補えそうであることを条件に,以下の 2 つを選択した.

- WUT 法[1]
- RCS 法[2]

3.2.先行研究

3.1 で抽出した先行研究の概要は以下のとおりである.

表 1　先行研究の概要および期待する効果

適用タイミング（改善対象）	手法名	適用における課題	期待する効果
欠陥検出	● WUT 法 レビューア向け思考能力（仮説力･要約力）トレーニング法	● 要約力・仮説力を求められる ● トレーニングが必要	● 要約による対象の迅速な理解 ● 細かい部分にとらわれないことによる全体の俯瞰 ● 仮説を立てることによる起こり得るリスクの予見
指摘伝達	● RCS 法 指摘を前向きに受け止めてもらうためのレビュー手法	● 想像力・言語能力を求められる	● 相手のタイプや性格に応じた指摘伝達(言葉遣い) ● 指摘の内容や意図について,レビューアと作成者間でのズレの解消 ● 作成者が指摘の意図について納得

以上の手法は,一定の効果が期待できるものの，要約力、仮説力、言語化力など体得・実践に時間を要する自然言語能力（スキル）に依存するため，該当するスキルを持たないレビューアのいる現場では,導入して一定の効果を得られるまで時間がかかる可能性がある.

3.3.仮説

生成 AI の強みとして主に,言語能力,応答速度,知識の幅広さの 3 点が挙げられる.我々はその 3 点に着目し,先行研究の課題として挙げられたスキル依存性を解決する一途となるのではないかという仮説を立てた.

3.4.提案

生成 AI の強みによって先行研究における課題が解決されるという仮説をもとに,先行研究における作業の一部を生成 AI に代行させる手法を以下に記載する.

3.4.1.欠陥検出のアシストに ChatGPT を用いた手法

先行研究である WUT 法では,以下の手順で欠陥を検出する.[1]

1. レビュー対象の特徴を掴む（課題：要約力が必要）
2. レビュー対象が実現すべき仕様について仮説を立てる（課題：仮説力が必要）
3. 実際にレビューを実施して指摘する

この手順のうち,1.と 2.を生成 AI にアシストさせる方法を提案する.

(1) 特徴を掴む手順を生成 AI にアシストさせる方法

レビュー対象成果物を生成 AI への入力として,文章を要約させることで,要約力のスキルが不足している人間でもシステム全体を俯瞰で見て,特徴を掴みやすくする.この提案は，ChatGPT の強みである,言語能力と応答の速さに着目したものである.文書の要約を ChatGPT にアシストさせることで,言語能力を補うとともに,要約にかかる時間を短縮することができる.

(2) 仮説を立てる手順を生成 AI にアシストさせる方法

レビュー対象成果物を作成するためのインプット文書を生成 AI に読み込ませて,レ

ビュー対象成果物に記載すべき事項を挙げさせる.それを見ることで,レビューアが記載すべき事項の漏れ等について仮説を立てやすくなる.

この提案は,ChatGPT の強みである,知識の幅広さに着目したものである.一般的に必要とされる要素を幅広く洗い出すことにより,重大な欠陥を見逃しにくくなる.

3.4.2.指摘伝達のアシストに ChatGPT を用いた手法

先行研究である RCS 法では,以下の手順で指摘を伝達する.[2]

1. 作成者のコミュニケーションスタイルを判別する
2. 欠陥を検出し,指摘事項を記載する
3. 作成者のコミュニケーションスタイルに適した表現で指摘を伝える
   （課題：想像力・言語能力が必要）

この手順のうち,3.を生成 AI にアシストさせる方法を提案する.

(1) 指摘を伝える手順を生成 AI にアシストさせる方法

指摘と,作成者のコミュニケーションスタイルを生成 AI への入力として,作成者が受け入れやすい表現に言い換えてから伝えることで,作成者が指摘を納得しやすくなる.

この提案は,RCS 法のベースとなっている,CS 法で言及されている 4 つのコミュニケーションスタイルについて,ChatGPT が学習済みであることに着目したものである.作成者のコミュニケーションスタイルに適した表現に言い換えさせることで,自分と異なるコミュニケーションスタイルの人の考え方を想像するのが苦手なレビューアでも言い換えを行うことができる.これにより,作成者が納得しやすい指摘ができ,意図しない修正や修正漏れを防ぐことができる.

4.実験内容

ChatGPT のバージョンについて,本研究における実験ではすべて ChatGPT3.5 を使用し,レビュー対象としては,要求仕様書を対象とする.

4.1.欠陥検出のアシスト

4.1.1.欠陥検出のアシストに ChatGPT を用いた手法の有効性確認

提案手法によって,スキルの差に関係なく重大な欠陥を検出できることを検証するため,以下の実験を行った

(1) 事前準備

〈実験参加者の選定〉

研究員の所属する会社から 6 名選出し,3 名ずつ 2 つのグループ（A/B）に分けた.

〈レビュー対象の準備〉

架空の要求仕様書を 2 つ準備した.（ともに 5 ページ,2,000 字程度）

〈ChatGPT による要約資料・仮説資料の準備〉

ChatGPT によって,以下の資料を準備した.

・レビュー対象の全体を要約させた資料

・レビュー対象の背景をもとに,対象のシステムにどういった画面が必要で,画面の仕様として何が必要かについて記載された資料(仮説資料)

要約資料・仮説資料を作成するにあたって ChatGPT に入力したプロンプトについては表 2 に記載する.

表 2　資料生成に用いたプロンプト

生成処理	プロンプト
要約資料の生成	下記の要求仕様書について,初めて見る人が内容を把握できるように,背景も含め,箇条書きで要約してください. [要求仕様書の全体]

仮説資料の生成	あなたは優秀なシステムエンジニアである.今回,システムの開発を行うことになり,システム化の背景を下記にまとめた. このシステムに必要な画面と,それぞれの画面の仕様を挙げてほしい. [要求仕様書の背景部分のみ]

〈実験について振り返りを行うためのアンケートの準備〉

実験参加者の感想をもとに実験の振り返りを行うため,アンケートを準備した.

アンケートの設問および回答の一部を図 1,全体を付録⑤に記載する.

(2) 実験手順

実験におけるレビュー方法を以下に定義する.

- 手法なし：各レビューアそれぞれのやり方で要求仕様書をレビュー
- 手法あり：まず,要約資料・仮説資料をもとにレビューした後,
  要求仕様書も対象に追加してレビュー

グループ A/B の全メンバーが手法を用いた場合とそうでない場合の双方でレビューできるよう,以下の割り振りでレビューを実施する.

表 3　欠陥検出アシスト検証実験の割り振り

	レビュー１回目 レビュー対象：要求仕様書 1	レビュー２回目 レビュー対象：要求仕様書 2
グループ A	手法なし	手法あり
グループ B	手法あり	手法なし

(3) 実験結果

2 つの実験完了後,実験によって得られた指摘事項の一覧を以下の 3 つに分類した.

「分類 1：誤字・脱字」「分類 2：記述が曖昧・不足」「分類 3：仕様検討漏れ・誤り」

本実験では,「分類 3：仕様検討漏れ・誤り」を重大欠陥として取り扱っている.

分類された指摘事項から,手法の有無を比較した結果について実験結果に記載する.

グループ毎の平均欠陥検出数を表 4-1,レビューア毎の欠陥検出数を表 4-2 に示す.

表 4-1　欠陥検出アシスト検証実験の結果(グループ毎)

	手法なし 平均欠陥検出数(件/人)			手法あり 平均欠陥検出数(件/人)			前後比 (手法あり÷手法なし)		
	分類 1	分類 2	分類 3	分類 1	分類 2	分類 3	分類 1	分類 2	分類 3
グループ A	5.3	6.7	6.3	2.0	2.0	10.0	38%	30%	158%
グループ B	2.0	4.7	5.0	2.3	3.3	5.0	117%	71%	100%
平均	3.65	5.7	5.65	2.15	2.65	7.5	59%	47%	**132%**

表 4-2　欠陥検出アシスト検証実験の結果(レビューア毎)

	手法なし 欠陥検出数(件)			手法あり 欠陥検出数(件)		
	分類 1	分類 2	分類 3	分類 1	分類 2	分類 3
レビューア A_1	5	4	2	2	2	2
レビューア A_2	8	10	1	2	1	5
レビューア A_3	3	6	16	2	3	23
レビューア B_1	2	2	8	6	0	7
レビューア B_2	2	6	1	0	4	2
レビューア B_3	2	6	6	1	6	6

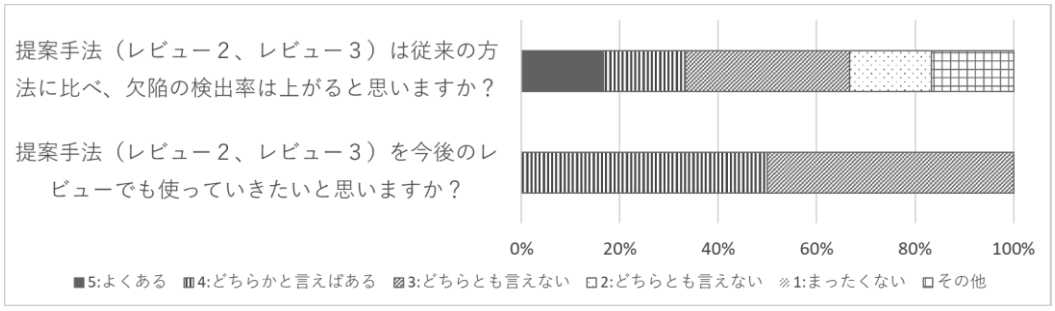

図 1　使い勝手と今後についてのアンケート結果

実験の結果,手法ありによる重大欠陥の検出数は,レビューア独自の手法と比較して前後比 132%となっている.先行研究における前後比が 135%であるため,重大欠陥の検出数については先行研究と同程度の効果を得ることができている結果となった.レビューア毎の重大欠陥検出数については,提案手法によって増加したのが 3 名,増減なしが 2 名,減少したのが 1 名であり,個人差はあるが,効果を得ることができた人数の方が多い結果となった.分類 1,分類 2 についての検出数は減っているが,より重大な欠陥を先に検出したいという提案手法の目的に沿った結果であると考えられる.

4.1.2.欠陥検出のアシストに ChatGPT を用いた手法の考察

実験により,提案手法の有効性を確認することができた.

先行研究の手法において,レビューアの要約力や仮説力が求められるため,導入のハードルが高いという課題があったが,生成 AI を活用することにより,導入における課題をクリアした上で,スキルの高低に関係なく一定の効果を出せるという結果が過半数以上のレビューアで得られた.効果が出なかったレビューアについて,原因はわかっていないため,今後も引き続き検証していきたい.また,図 1 より,今後のレビューでも使いたいという回答が多かったことが,提案手法の導入しやすさを表している.一方,アンケート回答の中には「要約・仮説資料はテキストのみでわかりにくい」「要約によって全体の把握はやりやすくなったが欠陥検知に繋がるかはレビューアのスキルに依る」「仮説資料が逆に混乱を招く要因となった」という旨の回答もあり,今後,改善の余地があることがわかった.

提案手法の懸念点として,生成 AI が出力した要約や仮説の結果を鵜呑みにしてレビューアが思考停止した状態でそのまま使われてしまう可能性があることや,まったく使えないものであると判断して活用しようとしないことなどが考えられる.生成 AI の出力内容はレビューアをアシストするためのものであって,あくまでも見るべきはレビュー対象であるということを留意する必要を感じた.

4.2.指摘伝達のアシスト

4.2.1.指摘伝達のアシストに ChatGPT を用いた手法の有効性確認

(1) 実験内容

提案手法を用いて指摘を言い換えて伝えることで,作成者が指摘を受け止めやすくなることを検証するために,以下の実験を行った.

指摘を受け止めやすくなる状態は,具体的には以下の観点で判断することとする.

・指摘の言い方に好感が持てるか

・指摘の内容が納得できるか

(2) 実験手順

研究員 4 名で,提案手法の有効性を検証する実験を行った.

実験は,下記の手順で行った.

① 研究員を作成者 1 名とレビューア 3 名の役割に分ける

② 先行研究における RCS 判定質問事項[2]に基づいて,作成者の RCS（レビューコミュニケーションスタイル)の種類を判別する
③ 作成者は,レビュー対象をレビューア 3 名に提出する
④ レビューアは,各自でレビュー対象を読み込み,指摘事項を列挙する
⑤ 指摘事項を ChatGPT に読み込ませ,作成者の RCS に適した表現に言い換える
⑥ 言い換え前と言い換え後の指摘事項の両方を作成者に提出する
※指摘内容で言い換え後のものとわかってしまう可能性もあるが,どちらが言い換え後かわからないようにする
⑦ 作成者が,提出された指摘事項に対して,「言い方の好感度」「内容の納得度」の 2 つの観点を 5 段階で評価する.
⑧ 言い換え後の評価に 1 がついたものについては,言い換え後の内容を確認し,明らかに日本語として破綻している場合は評価対象から外す
⑨ ①～⑧の手順を,役割を変えながら 3 回実施する

②の判別に用いる RCS について表 5 に示す.

表 5　RCS の種類と特徴[2]

	RCS の種類	特徴	
a	デシジョン RCS	決断が早く論理派	成果が大事、自分で決めたい
b	クリエイティブ RCS	決断が早く感情派	楽しいが大事、創造的、皆で決めたい
c	ロジカル RCS	決断が遅く論理派	正確が大事、考える時間、自分のペース
d	エモーション RCS	決断が遅く感情派	善いが大事、役に立ちたい、調和

研究員 4 名のうち、クリエイティブ RCS に該当する研究員がいなかったため,今回はデシジョン RCS,ロジカル RCS,エモーション RCS の 3 タイプの RCS で実験を行う.

実験に用いたプロンプトの一部を表 6 に示す.

表 6　言い換えに用いたプロンプト

生成処理	プロンプト
言い換え	人のコミュニケーションスタイルにはいくつかのパターンがあります.まず,決断が速いか遅いかの 2 パターンに分けられます.さらに,理論派か感情派かの 2 パターンに分けられます.つまり,全 4 パターンに分けられます. 次の[指摘内容]を,[RCS の種類]の人に伝わりやすい言い方で言い換えてください. [指摘内容,RCS の種類]

(3) 実験結果

実験結果を表 7,表 8,言い換えの具体例を表 9 に示す.

表 7：言い換え前後の評価(言い方の好感度)

	RCS の種類	言い方の好感度(平均)	
		言い換え前	言い換え後
1 回目	ロジカル RCS	3.23	**4↗**
2 回目	デシジョン RCS	3.55	2.91↘
3 回目	エモーション RCS	2.95	**3.05↗**

「言い方の好感度」に関しては,1 回目では,期待通りに言い換えによって向上する結果となったが,2 回目では,低下する結果,3 回目では,わずかに向上する結果となった.

表 8：言い換え前後の評価(内容の納得度)

	RCS の種類	内容の納得度(平均)	
		言い換え前	言い換え後
1 回目	ロジカル RCS	3.69	3.54↘
2 回目	デシジョン RCS	3.36	3.09↘
3 回目	エモーション RCS	2.79	2.68↘

「内容の納得度」に関しては,言い換えによって内容が変化しない前提であったため,言い換えによる変化は発生しない結果を想定していたが,1 回目～3 回目の全てでわずかに低下する結果となった.

表 9：言い換えの具体例

言い換え前	2. ユーザーのニーズと期待値 「炒飯が人気のある料理として注目されているユーザー」というのは、「炒飯が好きな人」なのか「炒飯が看板商品である店」なのかわからない。
言い換え後	「炒飯が人気のある料理として注目されているユーザー」というのは、「炒飯が好きな人」なのか「炒飯が看板商品である店」なのか、私にはよくわかりません。どちらかを教えていただけると、より具体的に理解できるので助かります。

詳細な実験結果については,付録⑥,⑦,⑧に示す.

4.2.2.指摘伝達のアシストに ChatGPT を用いた手法の考察

(1) 言い方の好感度についての考察

実験結果にて,言い方の好感度が向上する結果が出たことから,ChatGPT によって作成者の RCS に適した言い換えができていることがわかった.

- 言い方の好感度が向上したタイプ：エモーション RCS,ロジカル RCS
  - エモーション RCS に対して：へりくだった表現に言い換えられている,個人の感想ではなくユーザ目線での指摘になっている
  - ロジカル RCS に対して：意味のある短い文章を,接続詞で繋いで,論理的に展開された表現に言い換えられている

一方,言い換えによって言い方の好感度が低下する結果となったものもあった.

- 言い方の好感度が低下したタイプ：デシジョン RCS
  - デシジョン RCS に対して：言い方が回りくどい

(2) 内容の納得度についての考察

実験結果にて,内容の納得度が言い換えによって内容が変化しない前提であったが,言い換えによって,表現だけでなく,内容までも変更されてしまい,悪い影響を及ぼしたものと考えられる.

- 言い換えによって,内容が変更されてしまった,具体的な失敗例
  - 本来伝えるべき具体例や引用が省略されてしまい,何が言いたいかわからない
  - 肯定的な意味が否定的な意味に変わってしまった

内容の納得度を向上させるためには,指摘内容の変化を避けるためのプロンプトの入力を検討することも考えられるが,最終的には必ず人の目を通して言い換え後の内容を確認することが必要となってくると考えている.

5.まとめ

5.1.結論

本研究では，ソフトウェアレビューにおける生成 AI 活用をテーマに，欠陥検出と指摘伝達に関する 2 つのレビュー手法（SQiP 研究会の既存提案手法）に，生成 AI の活用を組み入れた．実験により，提案手法で必要とされるスキルの有無に依存することなく，提案手法を実践できること，期待効果と同等の成果を出せることが概ね確認できた．一方，指摘伝達における内容の納得度など，提案手法による効果が得られなかったものもあった．現状，生成 AI の出力を完全に信頼することはできず[3]，生成 AI を活用する上での課題の 1 つと言える．しかし，欠陥検出そのものなど，作業を完全に生成 AI に任せるのではなく，生成 AI が得意とする部分だけを切り出して人のスキルを補完するアプローチで活用すれば効果が期待できる．生成 AI の進化は激しく今後どこまで任せられるか，どんな活用の仕方が考えられるかは未知の世界であり無限の可能性を秘めているが，それは今後の期待としておいて，今回の研究では部分的な作業をアシストしてもらう活用の仕方に留めておく．

5.2.今後の展望

本研究における今後の課題は，以下の通りである．

(1)　ソフトウェア開発現場への導入と改善

現状，我々の所属組織において生成 AI に機密情報を投入できる環境は整備段階であるが，今後，提案手法を実際の業務で適用し，効果を測定すると共に，課題の洗い出しと改善を図りつつ，提案手法を継続的に適用することで，レビューア自体の能力の向上に繋がる副次的な効果も狙っていきたい．

(2)　ソフトウェア開発プロセスにおける生成 AI の活用への対応

生成 AI の活用は，他のソフトウェア開発プロセスにおいても活用が進んでいるため，プログラミングやテストの工程などにも着目し，生成 AI から出力された成果物に対するレビューの在り方についても検討していく必要がある．

またプロンプトエンジニアリング[4]という言葉が広まりつつあるとおり，今後は，プロンプトを対象としたレビューも検討していく必要があると我々は考え，試しに要求事項をインプットとして要件定義書を生成 AI に作成させる際のプロンプトをレビューする実験を行ってみたところ，作成者が考えつかなかった発想を獲得し，プロンプトが改善され生成 AI からの出力の品質も向上する効果が確認できた．作成するにあたって ChatGPT に入力したプロンプトおよび出力結果については付録⑨に記載する．加えて，副次的な効果として，有識者が持つプロンプトエンジニアリングについてのスキルを全体で共有できることや，成果物作成に必要な情報や方針を早期に合意形成できるという効果も見られた．

アジャイル開発方式やリーン思考を取り入れる組織が増えていく中，今後は，成果物が完成してからレビューを実施するのではなく，生成 AI との共創で成果物を作成しながら，レビューも実施するという方式が主流になる可能性もある．その点も踏まえながら，今後の生成 AI の進化に対してアンテナの感度を高く持ち，ソフトウェア開発プロセス全体の品質・生産性向上のために何ができるか継続的に考えていきたい．

6.参考文献

[1]荒井 良幸，久松 利光，延原 敦「レビューア向け思考能力トレーニング法の提案- 仮説力と要約力で，重大欠陥の検出効率向上」2016

[2] 弦間 健，辻村 隆二，伊藤 修司「指摘を前向きに受け止めてもらうためのレビュー手法提案～RCS 法（レビューコミュニケーションスタイル手法）の提案～」2018

[3] 徳本 晋 「AI によって変わる品質の考え方」2023

[4] Prompt Engineering Guide https://www.promptingguide.ai/jp