

生成 AI を活用した不具合帳票記述支援 エージェントの検討

A Study of Generative AI-Based Support Agents for Defect Report Documentation

2024年度、生成AIの業務活用が急速に進展する中、日立ソリューションズ東日本(以下HSE)では、ソフトウェア開発工程における不具合帳票(以下B票)の記述支援を目的としたAIエージェントの検討を行った。本稿では、大規模言語モデル(以下LLM)を活用した記述支援エージェント「B票羅針盤*1」の構成と実装方法を紹介し、社内開発案件および請負案件における実証結果を報告する。検証の結果、記述品質のばらつきやレビュー工数の課題に対して一定の効果が確認され、品質保証部(以下QA)によるレビュー支援にも有用であることが示された。一方で、LLM特有のハルシネーションやポチヨムキン理解のリスクも存在するため、活用方法には留意が必要である。今後は、社内展開および他業種への応用可能性についても検討を進める。

松本和芳 Kazuyoshi
Matsumoto

1. はじめに

2023年11月 米国Microsoft社は Microsoft 365 Copilotをエンタープライズ向けに一般公開した。OpenAI社が開発したLLMであるGPTシリーズを使用して、Microsoft 365アプリケーション(Word, Excel, Outlook, Teams等)と連携するAIシステムとして注目を集めることとなった。日立グループも2024年度から全社員が利用できるようになり、試行錯誤を繰り返しながら、生成AIを業務に活用し、効率化を目指した。2024年度は生成AI活用元年になった。

HSEの生産技術部はこのLLMをソフトウェア開発工程に導入し、生産性向上と作業品質向上を両立するユースケースを検討してきた。本稿は、その中でも、テスト工程への活用事例を示し、その有用性について考察をする。

2. B票記述とその問題点

(1) テスト工程とB票

システム開発では、仕様書/設計書に基づき、「作成したシステムが、このような操作をすれば、このように動作するはず」をまとめた帳票に基づき、実際開発したシステムを動作させることで、期待通り動作するかを確認する作業

を行う。これをテスト工程と呼ぶ。対象とする範囲や、テストの粒度により、「単体テスト(以下UT)」、「組み合わせテスト(以下CT)」と呼び、段階的にテストを実施する。このとき、期待通り動作しなかった場合、それを不良(または、不具合)と呼ぶ。

多くのプロジェクトにおいて、発生した不良に対応して、その不良がどのような不良なのかを示す「帳票」を作成する。この帳票をB票と呼ぶ。

(2) B票と不良分析

プロジェクトによって、差異はあるものの、「B票」には「現象」「原因」「対策」が記述されており、それぞれ100字~500字程度の自然言語で記述するのが一般的である。これらのテキストに基づいて、それぞれの不良に対して、いくつかの分類コードを割り当てる。コード分類することで、対象プロジェクトがどのような種類の不良がどのモジュールに多く含んでいるか、分析し(以下、不良分析)、次工程に進んでよいかの判定に利用する。このように、不良の管理と分析は高品質な開発工程に不可欠である。

(3) B票記述の問題点

B票の記述は、テスト担当者にゆだねられており、B票の記述品質にばらつきがある。B票記述に不備や不足があった場合、以下のようなリスクがある。

*1 B票羅針盤は社内呼称。B票記述を正しい方向に導くという意味を込めている。

- 現象を再現できずに、修正が不十分になる。
- 原因の本質を見失い、同じ間違いを再発させる。
- コードが適切ではなく、不良分析ができない。

特に、そのプロジェクトに初めて参加したメンバがいた場合、この課題は顕著に現れる。HSEではこれまで、B票の記述レビューに多くの人員と時間をかけて対応してきた。しかし、限られた開発日程の中で、十分な時間が取れず、後工程でその問題が発覚し、手戻りが発生することも多々あった。

3. 不具合帳票記述支援エージェント

(1) 評価システムの開発

前章で記載したコードが不適切になる課題は、自然言語処理分野で研究が進められており、コードの自動分類システムも数多く提案されている。LLMの発展と普及に伴い、より高精度な分類が可能になった。その一方、元になる文章記述品質の課題が残っている。そこで、筆者はLLMを活用した文書校正に着目した。LLMを使った文章校正は幅広く利用されている。この技術で、文章記述品質の課題を解決できると考え、方式の検討を開始した。

(2) 想定されるユースケース

想定されるユースケースは以下のとおり(図 1)。

- エージェント提供者はあらかじめ、「現象」「原因」「対策」からなるB票の構造を含んだ「システムプロンプト*2」と「記述ルール」を設定する。
- プロジェクトリーダー(以下PL)が必要に応じてそれらをカスタマイズする。
- 担当者は、「B票」と「AIに何をしてほしいのか」を示す「ユーザプロンプト*3」を入力する。
- AIは、「システムプロンプト」に基づき「ユーザプロンプト」から指示を抽出し、それに基づき「B票」を「記述ルール」に照らし合わせ、「指摘」と「B票の修正案」を作成する。
- 担当者は、プロンプトの返答として、「指摘」と「B票の修正案」を得る。

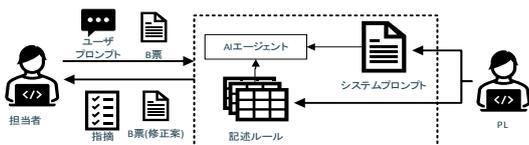


図 1 評価システムのユースケース

*2 AI に与える指示や役割、ルールなどをまとめた指示文。AI がどのように動作すべきかを定義する。

(3) 評価システムの実装

当初は、単純にLLMに対するプロンプトの工夫により、B票の記述に対する改善提案を取得できないかと試行した(図 2)。

```
# Role
あなたはソフトウェア開発における原因分析の専門家です。今回のあなたの役割は、これらのテキストが適切な形式で書かれているかを確認することです。

# 指示
テキストがソフトウェア開発における不具合記録帳票として適切か確認してください。

#不具合帳票

## 現象
...(略)...
```

図 2 当初試行したプロンプト(抜粋)

試行では、一般的な日本語の誤字脱字の指摘にとどまり、出力結果が、「現象」「原因」「対策」の構造を取らなくなるという動作を示した。

そこで、プロンプトの工夫を行い、B票が「現象、原因、対策」が記述されているドキュメント構造があり、それぞれが特定のルールに従う必要がある旨をプロンプトに追記した。その結果、指摘精度が上がり、十分実用に耐えるものが得られた。

最終的に、システムプロンプトをより精度よく読みとるようにし、SharePoint上のファイルを参照しながら動作するエージェントとして、Microsoft 365 Copilot Studio Lite上に実装した。これを「B票羅針盤」と呼ぶ。システムプロンプトと、記述ルールの一部を示す(図 3, 図 4)。

```
# Role
あなたはソフトウェア開発における原因分析の専門家です。ソフトウェア開発において、現象、原因、対策を体系化することで、...(略)...。分析を行うためには、現象、原因、対策が適切に記述されていることが重要です。今回のあなたの役割は、これらのテキストが適切な形式で書かれているかルールを参照しながら確認することです。

# 共通の指示
- ナレッジにあるファイルを参照しながら「現象、原因、対策」のテキストを確認してください。
...(略)...

# 機能 1: 「現象」に書かれたテキストを確認します
...(略)...ナレッジにある「現象記述ルール」に従って、ソフトウェアの不具合に対する「現象」に書かれたテキストを確認します。
...(略)...
```

図 3 システムプロンプト(抜粋)

- 再現手順を明確に記述できていること
- 発生環境が記述されていること
- 期待される動作と実際の動作を分けて記述されていること
- 再現頻度が記述されていること
- ...

図 4 現象記述ルール(抜粋)

*3 ユーザが AI に対して具体的に何をしてほしいかを入力する指示文。

このように、B 票の構造がどのようになっており、各項目について、どのような表記ルールがあるかを細かく記述した。その結果、より精度の高い指摘を出力できるようになった(図 5)。



図 5 Microsoft 365 Copilot Chat Agent による実装

更に、追加プロンプトで、修正案を示してもらうことも可能である。

4. 実務における検証

4.1 請負開発案件での検証

(1) 案件概要

B 票羅針盤を、ソフトウェア請負開発案件の単体テスト (UT) と組み合わせテスト (CT) に適用した。主要メンバは対象システムの開発を継続受注している。本プロジェクトはそのシステムに対する改修案件である。

(2) 本プロジェクトにおける不良管理手順

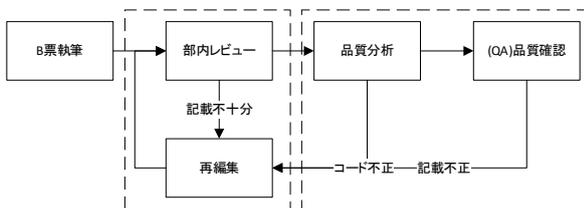


図 6 不良管理手順

このプロジェクトでは、テスト工程で抽出した問題点はすべて、テストを担当した担当者が B 票を記述する。この B 票は、PL による部内レビューを実施する。その後、PL は、B 票に各種分類コードを付与し、不良分析を行う。また、B 票一覧と品質見解書は、QA のレビューにより、次工程への推進可否を決定する。

(3) B 票羅針盤の適用評価

本プロジェクトでは、UT 工程、CT 工程で、それぞれ 3 件、4 件の不良を抽出した。規模も小さく、発生不良数も少なかったため、部内レビューの手順で、PL が B 票羅針盤を使い、得られた修正案を元にプロジェクトに合わせて B 票を修正した。

前回プロジェクトでは、B 票の表現が不十分で、不適切な記述が散見されたため、品質分析時に「B 票全体を見直す」という手戻りが発生していた。しかし、B 票羅針盤の適用により、部内レビューにおける「記載不十分」な箇所がなくなり、品質分析時に抽出される「コード不正」も解消された。試算ではあるが、部内レビューおよび品質分析における手戻りを含んだ工数削減の効果は、本プロジェクトでも、3~5 人日程度となった。規模が大きい案件では、さらなる削減効果が見込めると筆者は考える。

(4) B 票羅針盤を使った B 票修正事例

以下は、実案件で発生した不良に対する B 票の「原因」のテキストである。B 票羅針盤を使って修正した事例を示す(図 7)。

修正前 (入力)	Pandas.read_excel 関数で出力テンプレートファイル (Excel) を読み込んでいる。当該処理にて、「テンプレート」シートの存在チェックを行わずに、「テンプレート」シートへのアクセスを行っていたため、ValueError が発生し、API 処理が異常終了していた。また、前述の通りシートの存在チェックを行っていないため、ログにエラーメッセージが出力されなかった。
修正例 (出力)	この不具合はアプリケーションの疎通確認時に、開発担当者が発見した。開発担当者が原因特定のため、ログファイルを解析した。ログには「[#7] output file is ready [xxx]」(※xxx はリクエスト ID) というメッセージを最後に、以降の処理が記録されていなかった。 その後、開発担当者がプログラムを調査した。pandas.read_excel 関数で出力テンプレートファイル (Excel) を読み込んでいる。 当該処理にて、「テンプレート」シートの存在チェックを行わずに、「テンプレート」シートへのアクセスを行っていることが判明した。 この実装により、シートが存在しない場合に ValueError が発生し、API 処理が異常終了していた。さらに、シートの存在チェックが行われていなかったため、ログにエラーメッセージが出力されず、原因の特定が困難になっていた。 本現象の直接的な要因は、開発者がエラー処理の設計を十分に検討していなかったことである。

図 7 B 票羅針盤を使った修正事例

当初の文章にはなかった、原因特定に至るまでの経緯や、「開発者がエラー処理の設計を十分に検討していなか

った」のような動機的原因までの確に書けている。この品質の記述に至るまで、およそ 2~3 往復のチャットでたどり着いた。時間にすると 5~10 分程度であった。

4.2 自社開発案件での検証

B 票羅針盤を、QA が参加した P 票検討会に適用した。P 票とは、B 票を元に、どのようにソースコードを修正するかを記述したドキュメントである。B 票と同様「現象、原因、対策」の主要な構造は同じである。P 票検討会は開発を担当した事業部門と、QA 等関係部門でレビューする会議である。

弊社の製品開発プロジェクトの P 票検討会に参加した QA が、当日入手した B 票をレビュー中に手元で B 票羅針盤に投入することで、観点漏れがないか確認する目的で活用した。

会議では、限られた時間内に、的確な指摘を行う必要があるため、観点漏れや指摘漏れがあったが、B 票羅針盤を適用することで、B 票を俯瞰的に見ることができ、レビューの視点を確認するために非常に効果があった。

5. 評価

以上の検証から、不具合帳票記述支援エージェントはテスト工程において有用であることがわかった。

その一方、B 票羅針盤に修正案を提示させると、LLM を基底に使っている以上、ハルシネーション*4やポチョムキン理解(Potemkin Understanding)*5の影響が大きく出る。実際修正案の提示では、ありもしない画面名や機能名、存在しないエラーメッセージも回答に含まれていた。ただし、B 票羅針盤から提示されたものが、明らかに「例」であるとユーザが理解しているユースケースにおいては問題にならない。

以上により、最終成果物(納品物)の作成や、夜間バッチでの自動訂正のような使い方ではなく、以下のようなユースケースが妥当と考える。

1. 開発者が B 票を記載する際に手元に置いて、AI のサポートを得る。
2. PL や QA がレビュー時に品質確認ツールとして手元で確認時に利用する。

6. おわりに

以上により、B 票の記述不正を支援する AI エージェントは、システム開発の品質向上に貢献し、品質マインドの

醸成につながることを示した。今後、このエージェントは、社内はもちろん日立グループで活用できるよう展開するつもりである。特に、日立におけるソフトウェア開発経験が浅いパートナ企業メンバや若手社員への教育効果が高いと筆者は考える。

不具合に関する帳票はソフトウェア以外にも、製造業にも適用の可能性があるかもしれない。グループ会社を中心に、ニーズ調査とユースケース分析を実施するつもりである。

7. 謝辞

最後になりましたが、執筆にあたり、社内関係部署には多大なご支援をいただきました。開発業務での実適用とその検証を快く引き受けてくださった、HSE の社会基盤ソリューション第二本部のメンバ、特に斎藤さん、佐々木さん、川原田さん、金子さん、品質保証本部の吉川さん、西山さんからいただいたご協力に心より感謝申し上げます。

参考文献

- 1) AUTOMATED CLASSIFICATION OF DEFECTIVE CODE FROM BUG TRACKING TOOL DATA, 米国特許 US2022/0374333 A1(2022)
- 2) 生成 AI の実力は？ 文書校正業務への活用検証レポート - NTTTPC (2025/11/13 閲覧), https://www.nttpc.co.jp/technology/ai_proofread.html
- 3) AI for Enterprise Productivity | Microsoft 365 Copilot (2025/11/13 閲覧), <https://www.microsoft.com/en-us/microsoft-365-copilot/enterprise>
- 4) Potemkin Understanding in Large Language Models (2025/11/13 閲覧), <https://arxiv.org/pdf/2506.21521>



松本和芳 2001 年入社
生産技術部
静的解析ツール適用、テスト自動化等、ソフトウェア開発の生産技術全般。近年生成 AI の開発プロセスへの適用推進に従事

*4 AI が実際には存在しない情報を生成してしまう現象。

*5 AI が表面的には正しいように見える回答をするが、実際には深い理解が伴っていない状態。