# LLM を用いたソフトウェア開発プロジェクト管理における自動データ分析の評価

研究員：青山　弦太（株式会社ビッグツリーテクノロジー&コンサルティング）
主査：　石川　冬樹（国立情報学研究所）
副主査：栗田　太郎（ソニー株式会社）
副主査：徳本　晋（富士通株式会社）

**研究概要**

ソフトウェア開発プロジェクトの成功がプロジェクト管理者(以下,PM)の勘・経験・度胸に依存している問題がある.本研究では,これを解消するために本論文では生成 AI の大規模言語モデル（以下,LLM）を用いた PM に依存しないプロジェクト管理に関する分析について評価する.実験により,PM による分析の一部を LLM により代替でき,PM の意思決定までの時間短縮,意思決定の質向上に寄与する可能性を確認した.

## 1.　はじめに

コード解析，バグ修正，テスト，プロジェクト管理などのデータを解析し可視化するソフトウェアアナリティクスに関心が寄せられている[1]．ソフトウェアアナリティクスは，対応策の実施へと結びつくアクション可能な情報を成果物から得ることで，ソフトウェア開発の生産性向上，システム品質の確保，ユーザ体験の向上を狙いとしている[1]．特に，属人化された勘・経験・度胸を基にソフトウェア開発プロジェクトのマネジメントを行う場合,プロジェクトの成否は PM の能力に強く依存する.勘・経験・度胸を排除し,再現性のあるプロジェクト運営を行うために「データ駆動のプロジェクト運営」は重要である.

データ駆動のプロジェクト運営に期待感が寄せられている一方で，実体験としてデータに意義を見いだすことができている実務者は必ずしも多くない[1]．データ駆動のプロジェクト運営の方式や，データ駆動という思考様式をソフトウェア開発組織に定着させるには乗り越えるべき課題がある.

データ駆動のプロジェクト運営が定着しない理由の１つに,データ分析・洞察の導出に係る時間,コスト,個人の能力があるのではないかと筆者は考えている.例えば,組織にデータが溜まっていても,どのように分析していいか分からない,分析した結果をどのようにアクションに繋げるかが分からないなど,データ分析及び,洞察の導出にも時間,コスト,個人の能力が必要である.具体的には,プロジェクト運営のための時間やコストが余剰に設けられる機会があまりないため，十分な分析を行うための時間やコストを確保できず,結果としてプロジェクト運営者はデータ駆動のプロジェクト運営に関する知見がつかず能力つかないといった課題が存在する.

その解決のため，LLM を用いプロジェクトに関するデータから洞察,本論文ではその中でも特に,納期遅延なくプロジェクトを完了させるための洞察を自動で見出すことで,PM の力量に依存しないプロジェクト管理が可能かを評価する.

実験により,PM による分析の一部を LLM により代替でき,PM の意思決定までの時間短縮,意思決定の質向上に寄与することを確認した.

## 2.適用方法

プロジェクト管理におけるデータ分析の前提として,以下の２つを定義する.

A)　GQM（Goal, Question, Metric）[2]が定義されている.

B)　定義された Metrics が収集されている.

前提を満たした上で,収集された Metrics をもとに Question に答える行為を LLM にて代替することを考える.具体的には ChatGPT の Advanced Data Analytics プラグインを用いて,以下の要領で作成されたプロンプト[1]にて洞察を引き出す.

(1) 収集された Metrics をファイルとしてプロンプトに添付する

(2) Metrics からは読み取れない分析対象プロジェクトの前提条件を記載する.

例) プロジェクト期間,メンバの稼働,リスク

(3) 以下の要領で GQM の Question への回答を指示する.

Metrics の読み取り

各 Metrics を導出するための入力がされているかの確認を指示する.

各 Metrics の導出を指示する.

Question への回答

各 Question への回答を指示する

(4)任意のフォーマットを指定し出力を指示する.

前提 B で記載された,収集されている Metrics に対して上記の考え方で作成されたプロンプトを適用することで,いつでもプロジェクトの状態,および,それに対するアクションを導出できる.

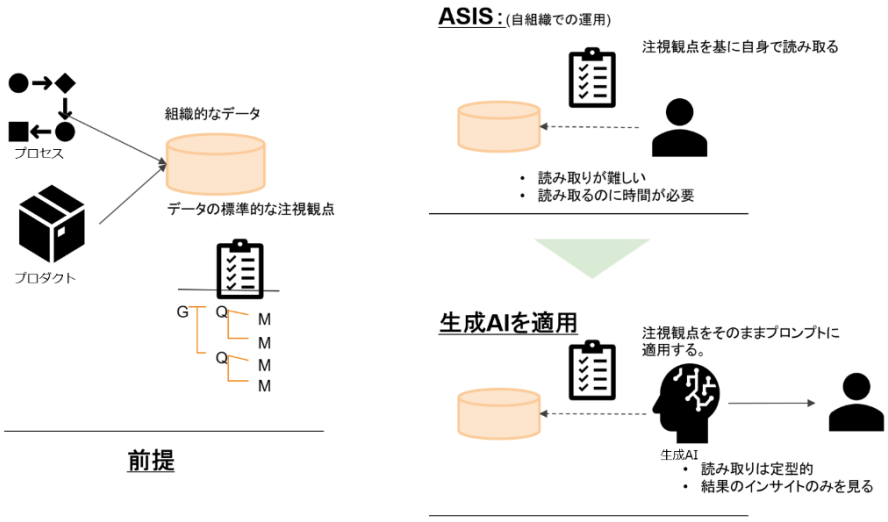


図 1.プロジェクト管理におけるデータ分析への LLM 適用イメージ

なお,LLM は図 2.のように数値計算,集計に弱い欠点がある.そのため,データ分析を行わせる際には,データ分析を行うためのソースコードを生成し実行させる仕組みが用いられる.本研究では ChatGPT の Advanced Data Analytics プラグインを用いてこれを実現している.

[1] 詳細なプロンプトの内容は付録を参照


図 2.単純な数値計算を誤る ChatGPT

## 3. 実験

### 3.1. 実験内容

自組織にて行われている GQM ベースでの分析に対して,プロジェクトマネージャがデータを参照して得られる洞察と,LLM が得る洞察の比較を行う.洞察に対する正解不正解はつけ難いため,以下の要領で洞察の分類し定性的な比較を行う.

組織内の特定のプロジェクトに対して,設計フェーズの特定のタイミング(20%/50%/70%の期間が経過したタイミング)の進捗に関するデータを用意する.用意されたデータに対して,PM と LLM それぞれに洞察を出力させ,その内容を比較する.図 3.における洞察は,以下のような特色があり, A,C が多ければ本手法はより有用,B,D が多ければ本手法の改善点として解釈することができる.

| プロジェクト人数 | データ量（ITS チケット数） | プロジェクト特性 |
|---|---|---|
| 11~20 名規模 | 463 件 | スクラム開発 |

表 1.対象プロジェクトの特性

| 洞察分類 | 特性 |
|---|---|
| A. LLM 独自の有用な洞察 | 人間が気づけない洞察の導出ができているため**付加価値を創造できている.** |
| B. LLM 独自の不要な洞察 | 不要な洞察を無駄に上げているため,**ノイズとなっている.** |
| C. LLM, PM 共通の有用な洞察 | 人間と同様の洞察を**人的コスト無く導出できている.** |
| D. PM 独自の有用な洞察 | 人間が気づく洞察を**漏らしている.** |
| E. PM 独自,また LLM, PM 共通の不要な洞察 | 人間が不要だと考える洞察を人間はみつけないため**存在しない.** |

表 2.洞察と特性のマッピング

※有用,不要は人間が判断する.

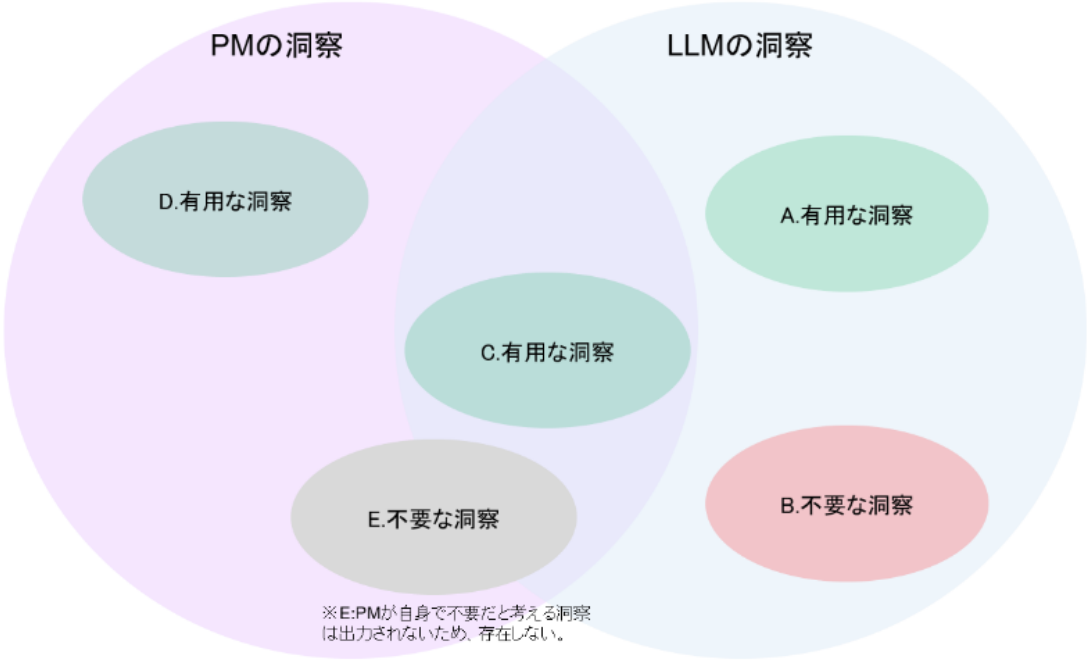


図 3. 洞察の分類

### 3.1.1. ツール

ChatGPT(GTP4)+Advanced Data Analytics を利用した.

### 3.1.2. プロンプト

3.適用方法の項に記載された要領で作成されたプロンプトを用いる.

※サンプルプロンプトの全量は付録に添付

### 3.1.3 データ

筆者の組織に貯められている以下の形式のデータ,及び,GQM を利用する.タスクは Redmine/Backlog/Jira などの ITS(Issue Tracking System)で管理されており,各タスクに対して表 3 の通りのデータが入力/収集されている.それを基に,表 4 の GQM の Question に回答していく.本 GQM は組織横断的に定められたものであるため抽象度は高いものとなっている.

| 項目 | 備考 |
|---|---|
| 開始日 | |
| 期日 | |
| 完了日 | |
| 作成日 | |
| ステータス | 未完了/進行中/レビュー中/完了 |
| 予定工数 | |
| 担当者 | |

表 3.収集されている ITS のデータ

| *Goal* | *Question* | *Metric* |
|---|---|---|
| 納期遅延なくプロジェクトを完了する | 計画は立てられている？ | 予定工数が入力されているタスクチケット数/入力されていないタスクチケット数 |
| | | 開始日期日が入力されているタスクチケット数/入力されていないタスクチケット数 |
| | 立てられた計画は妥当？ | 週ごと,リソース毎の対応予定工数 |
| | | タスクチケット毎の予定工数の大きさ |
| | 計画通りに進んでいる？ | 実績完了チケット工数の和/完了予定チケット工数の和 |

| | | |
|---|---|---|
| | | 予定工数の時系列の増加量 |

表 4. 組織横断的に定められた GQM

### 3.2. 実験結果

設計フェーズにおける実験結果は以下のとおりである.

黒字はプロジェクトが良い状態だという判断.赤字はプロジェクトが悪い状態だという判断である.

**設計フェーズ 20%経過時点**


**設計フェーズ 50%経過時点**


**設計フェーズ 70%経過時点**


### 3.2.1. 考察

C.LLM,PM 共通の有用な洞察が多く,特定の観点を定めた場合において,LLM から得られる導出は PM がデータを分析して得る洞察と多くが重複しており,意思決定の確度向上,洞察を見出す手間を削減し意思決定までの時間短縮へと寄与する可能性を示すことができた.一方,本実験は実験対象の数が少なく,確度の高い結果を得ることができなかった.実験対象数を増やすことや,進行中のプロジェクトに適用することで,より確度の高い結果を得ることが期待できる.

また,一部[A.LLM 独自の有用な洞察],[D.PM 独自の有用な洞察]が見つかった.[A.LLM 独自の有用な洞察]に関しては,多くのチケットの常態を観察しないと見つからない洞察であり,このような洞察は人間ではなく機械的に見つけることが望まれるため PM 業務の補完に値すると考えている.

[D.PM 独自の有用な洞察]としてはプロンプトとして明確に定められていないが問題となる,探索的に気付くことが妥当な洞察であった.本観点をプロンプトに追加することで補完が可能だと考えるが,観点を全てプロンプトに落とし込むと LLM のプロンプトの健忘や,トークン数の上限に達してしまうことが想定される.そのため,以下の 2 点を行うことで改善ができるのではないかと考える.

1. 探索的に気付く洞察は人間,もしくは別のプロンプトでみつけるなど,プロンプトと人間の役割範囲を明確にする.
2. RAG (Retrieval-augmented Generation) を導入し,組織のコンテキストや知識を LLM に参照させることで,組織のコンテキストから探索的に気付く洞察を見つけさせる.

### 3.3. 妥当性への脅威

本実験では,実験の対象となる事例数が極端に少なく,統計的有意性はない.そのため,一部の限られた状態での成果を示したにとどまる.また,評価対象はデータから分かる洞察のみである.PM の頭の中にしか落ちていない情報を鑑みた際に,実プロジェクトでは PM 独自の有用な洞察が増えることが想定される.

## 4. おわりに

PM による分析の一部を LLM により代替でき,意思決定の質向上,意思決定までの時間短縮に寄与する可能性を示すことができた.一方で限られたごく少数データでの評価であり,有意性への評価は十分にできていない.

今後は,以下の 4 点に取り組むことで,実務に即した性能の評価,及び,向上が可能になると考えている.

1. 性能評価
   (ア) 実験対象プロジェクト数の増加.
   (イ) 進行中のプロジェクトへの適用
2. 性能向上
   (ア) RAG (Retrieval-augmented Generation) を導入し,組織のコンテキストや知識を LLM に参照させる.
   (イ) Human in the loop を意識し,実務でどのような運用を行うのかを定義し,それに即した出力を行うようプロンプトを最適化する.

## 5. 参考文献

[1] Svensson, R. B., Feldt, R., & Torkar, R. (2019, May). The unfulfilled potential of data-driven decision making in agile software development. In International Conference on Agile Software Development (pp. 69-85). Springer, Cham.

[2] Zhang, D., Han, S., Dang, Y., Lou, J. G., Zhang, H., & Xie, T. (2013). Software analytics in practice. *IEEE software*, 30(5), 30-37.

[3] Basili, V. R., & Rombach, H. D. (1988). The TAME project: Towards improvement-oriented software environments. *IEEE Transactions on software engineering*, 14(6), 758-773.