

# Rocks Coding, Not Development: A Human-Centric, Experimental Evaluation of LLM- Supported SE Tasks

---

WEI WANG, Beijing University of Posts and Telecommunications, China

HUILONG NING, Beijing University of Posts and Telecommunications, China

GAOWEI ZHANG, Beijing University of Posts and Telecommunications, China

LIBO LIU, University of Melbourne, Australia

YI WANG, Beijing University of Posts and Telecommunications, China

Proceedings of the 32nd ACM International Conference on the Foundations of Software Engineering(FSE '24)

# 背景 | 生成AIによるソフトウェア開発

ChatGPTのような最新のLLMは、コーディングパズルで優れた成績を収めている

- コーディングパズル

- プログラミング能力を評価するためのプログラミングタスク
- プログラミングコンテストで見られるような問題、問題文と入力形式、出力形式が与えられる

LLMがソフトウェア開発者にとって代わる可能性があるという主張も出てきている

- ChatGPTがアルゴリズム/コーディングパズルを解いた
- 大手企業のコーディング面接に合格

# 問題 | LLMのSEタスクにおける有効性

LLMの能力は主にコーディングパズルや単純なプログラミングタスクで評価されている

- 実際のソフトウェア開発タスクでの有効性は十分に検証されていない

LLMをソフトウェア開発に利用した際の影響や、開発者との協働の様子は明らかになっていない

- [68]で提唱された6種類の開発活動のうち”MODIFY”操作が大半を占める
- つまり開発には、単なるコード生成能力だけでなく、既存のコードを読む・理解・修正する能力が必要

※MODIFY

既存のコードの変更・修正操作を表す  
他には、ADDやREMOVEなどが存在

[68]Yang Yue, Yi Wang, and David Redmiles. 2023. Off to a Good Start: Dynamic Contribution Patterns and Technical Success in an OSS Newcomer's Early Career. IEEE Trans. Softw. Eng. 49, 2 (2023), 529–548.

# 目的・アプローチ

---

## 目的

- 実際のソフトウェア開発タスクにおけるLLMの有効性を明らかにする
- LLMと人間の開発者との共同の動態を実験的に評価

## アプローチ

- 109人の被験者を対象に実証実験
- コーディングパズル・典型的な開発タスクを解かせる
- ChatGPT使用有り無しのグループに分け実験
- 定量的・定性的データを収集・分析

# 実験概要①

## 参加者

- 109人の参加者(コンピューティング専攻, 1年以上のプロ開発経験)

## グループ分け

- 参加者を4グループにランダムに割り当て

W-CP

ChatGPT**有り**で,  
コーディングパズル  
31人

WO-CP

ChatGPT**無し**で,  
コーディングパズル  
25人

W-TD

ChatGPT**有り**で,  
**開発タスク**  
25人

WO-TD

ChatGPT**無し**で,  
**開発タスク**  
28人

# 実験概要②

---

## タスク内容

- コーディングパズル2問
  - PTAプラットフォームの中難易度の問題5問を取得，うち2問を解かせる
- 典型的開発タスク2問
  - GitHubプロジェクトのIssueに示された2つのバグを修正
  - いずれも一つのファイルを変更すれば修正できるバグ

## 実験プロセス

- 実験説明
- 環境への慣れ：数分
- タスク開始：75分間

# 実験で取得するデータ①

## 定量的データ

- 効率性
  - タスク完了時間(秒), 短いほど効率的
- 品質
  - テストケース通過率に基づくスコア, 100点満点
- NASA-TLX(主観的タスク負荷)
  - 作業負荷を評価するための心理測定手法, 参加者自身が主観的に採点
  - 精神的な要求, 時間的な要求, フラストレーションなど5指標
  - 各指標20点満点
- 主観的認識
  - ChatGPTの生産性向上効果に対する評価
  - 20点満点

## 主にChatGPTの有効性の分析に利用

# 実験で取得するデータ②

## 定性的データ

- 参加者の画面活動を記録・分析

内容

- 参加者とChatGPTの全ての対話
- コーディング活動
- デバッグプロセス
- 情報検索行動

## 主にChatGPTの使用パターンの分析に利用

- **コーディングパズル**・**開発タスク**において、参加者はどのようにLLMを利用するのか

# RQ1 | ChatGPTがもたらす利益①

## 効率性について

- **コーディングパズル** : ChatGPT使用グループが有意に効率的
  - 平均時間 : 2196.7秒 vs 2692.6秒, 統計的有意差あり (M-Uウィルコクソン検定)
- **開発タスク** : ChatGPT使用でわずかに効率的
  - 平均時間 : 2054.9秒 vs 2147.3秒, 統計的有意差なし

## 品質について

- **コーディングパズル** : ChatGPT使用でやや高品質
  - 平均テストスコア : 86.15 vs 77.87, 統計的有意差なし
- **開発タスク** : ChatGPT使用でわずかに効率的
  - 平均テストスコア : 73.68 vs 71.46, 統計的有意差なし

# RQ1 | ChatGPTがもたらす利益②

## タスク負荷(NASA-TLX)

- **コーディングパズル**： ChatGPT使用で一部有意に高評価
  - 大半の側面で有意差なし
- **開発タスク**： 全ての側面で有意差なし

## 主観的認識

- 全参加者がChatGPTの生産性向上効果に**肯定的**(16.12/20)
  - ChatGPTの使用経験・実験結果に関わらず高い評価

## まとめ

- ChatGPTは**コーディングパズル**を解くスピードを向上
- 両タスクで品質の向上にはあまり貢献しなかった
- ChatGPTは、**単純なコーディングタスク**では一部効果的だが、**より複雑な開発タスク**での有効性はかなり限定的

# RQ2 | ChatGPTとの相互作用①

## コーディングパズルでのChatGPT使用パターン

- **ChatGPT完全依存型** (31人中5人)
  - 問題全体をChatGPTに入力し、生成されたコードをそのまま使用
  - ChatGPTの提案をコピーしてそのまま使用
- **ChatGPT部分依存型** (31人中17人)
  - 初期コードをChatGPTに生成させ、その後自分で改良
  - 完全依存型よりも良いパフォーマンスを示した
- **従来型アプローチ** (31人中6人)
  - オンライン検索サービスを主に使用 (ChatGPTをあまり使わない)
  - コーディング、デバッグ、検索のサイクルを繰り返す

## まとめ (コーディングパズル)

- 最初のコード生成が従来のコード検索に取って代わっている
- 一方、従来の方法で行う人も少数存在した

# RQ2 | ChatGPTとの相互作用②

## 開発タスクでのChatGPT使用パターン

- **ChatGPT完全依存型** (25人中8人)
  - 全ての指示をChatGPTに入力し、提案されたコードをそのまま使用
  - 低品質な修正や、修正の断念につながることが多い
- **ChatGPT協働型** (25人中12人)
  - ChatGPTをコード理解、問題解決、改善案の議論に利用
  - ChatGPTの認知的・会話的な性質を最も効果的に活用し、より有用な回答を引き出す
- **最小限利用型** (25人中5人)
  - ChatGPTの使用を最小限に、自力で遂行、高いソフトウェア開発能力を持つ参加者に多い
  - ライブラリに関する質問のような、特定の質問のみに利用、非常に早くタスクを完了

## まとめ（開発タスク）

- 協働型が最も多く、人間の判断とChatGPT組み合わせるのが有効
- 完全依存型は低品質な解決や課題放棄につながるが多かった

# 総括

---

ChatGPTのようなLLMツールがソフトウェア開発に与える影響を包括的に評価

- 現時点では、SEタスクにおいて人間の開発者を置き換えられる性能には至っていない

開発におけるChatGPTの効果的な活用には、LLMとの協働が重要

- タスクの複雑さや開発者のレベルに応じて活用方法を検討する必要

より複雑な開発タスクでのLLMの有効性の向上や効果的な人間-AI協働が今後の課題

- 近い将来では人間の開発者にとって代わるとは考えにくい

# 所感

---

直近のLLMは限界レベルで高性能になってきている一方で、より複雑で実務的なタスクにおける実用性は想像以上に欠けているようで驚いた

単なる生成AIの利用から、SEタスクなど分野に特化した生成AIの開発・活用へ

- 各タスクについてより実用的なツールの開発
- SEタスクに特化したLLMの評価指標

# 補足：コーディングパズル例題

## ・海賊船での生存テスト

満載の船が海を航行中、突然海賊船に囲まれます。海賊たちは船を乗っ取り、全乗組員（合計N人）をデッキに引きずり出します。海賊の船長が言います：「ゲームをしよう。勝者は去ることができる。ルールは次の通りだ：N人で円を作り、最初の人から数え始める。M番目の人が円から出る。そして数え続け、M番目ごとに人を円から出す。K人が残るまで続ける。」与えられたN、M、Kに対して、円から出る順序と残る人を決定するプログラムを作成してください。

入力形式：N、M、Kの値が1行で与えられ、スペースで区切られています。入力は全て正の整数であり、 $N > K$ であることが保証されています。

出力形式：安全に円から出る順序を出力します。例えば、 $N = 6$ 、 $M = 5$ 、 $K = 2$ の場合、円から出る順序は5、4、6、2で、3と1が残ります。

## ・最も簡単な単語

多数の単語があり、学習者はこれらの単語を難易度に基づいてグレード付けします。一定数の学習者が、自分が学習している10個の単語にグレードを付け、異なる人が同じ単語にグレードを付けることもあります。ある単語が複数の人によってグレード付けされた場合、その複合スコアは全てのグレードの平均となります。最も低い複合スコアを持つ単語を見つけるプログラムを作成してください。

入力形式：最初の行にT（データセットの数）。各データセットの最初の行にN（人数）。その後、 $N \times 10$ 行が続き、各行に単語と整数（スコアを表す）がスペースで区切られて入力されます。

出力形式：各データセットに対して、最も簡単な単語を出力します。

# 補足：開発タスクの内容

シーン紹介：

こんにちは、私はWeChat友達の情報を分析するPythonプロジェクトを書きました。それは現在のディレクトリにある「wechat\_friends」フォルダです。あなたが慣れているIDEでプロジェクトを開くと、以下の機能を実現できます：

1. 友達の性別分布を分析する
2. 友達の地域分布を分析し、地図上に可視化する
3. 友達のニックネームを使用してワードクラウドを作成する
4. 友達の個人的な署名の高頻度単語を分析する
5. すべての友達のアバターを1つの大きな画像に結合する

get\_user\_info.pyファイルを実行した後、81人のWeChat友達の情報を取得し、data/friends.jsonに保存しました。WeChatのアバターは「images」ディレクトリに保存されています（実行する必要はありません）。

analyse.pyを実行した後、上記の5つの機能を実装し、分析結果を「analyse」フォルダに保存しました。対応するHTMLファイルをブラウザで開いて閲覧できます。ただし、以下の2つのバグが見つかりました：

バグ1：一部の友達の個人的な署名に絵文字などの認識できない文字があるため、個人的な署名のワードクラウドに「span」、「class」、「emoji」などの単語が混ざっています。以下のように表示されています。この問題を解決してください。

バグ2：一部の友達にプロフィール画像がなく、画像モザイク処理中にスキップされました。そのため、合計81人の友達がいるにもかかわらず、以下の画像は64人の友達のプロフィール画像のみで構成されています。モザイク画像を作成する際、プロフィール画像のない友達には空白のプロフィール画像を使用してください。

# 補足：ChatGPTとの会話例

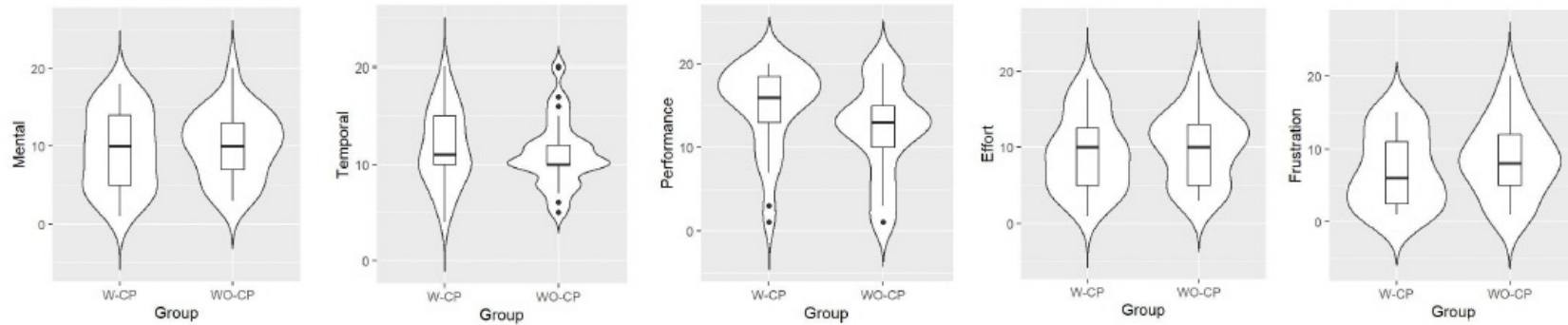
Prompt: Could you mind helping me do an algorithm puzzle?  
Prompt: Please use code solve the problem.  
[Put all instruction in]  
Prompt: Please!  
Prompt: Not right  
Prompt: Test my input  
[Repeated the above steps until give up]

a. The prompt history of a participant relying on ChatGPT for everything.

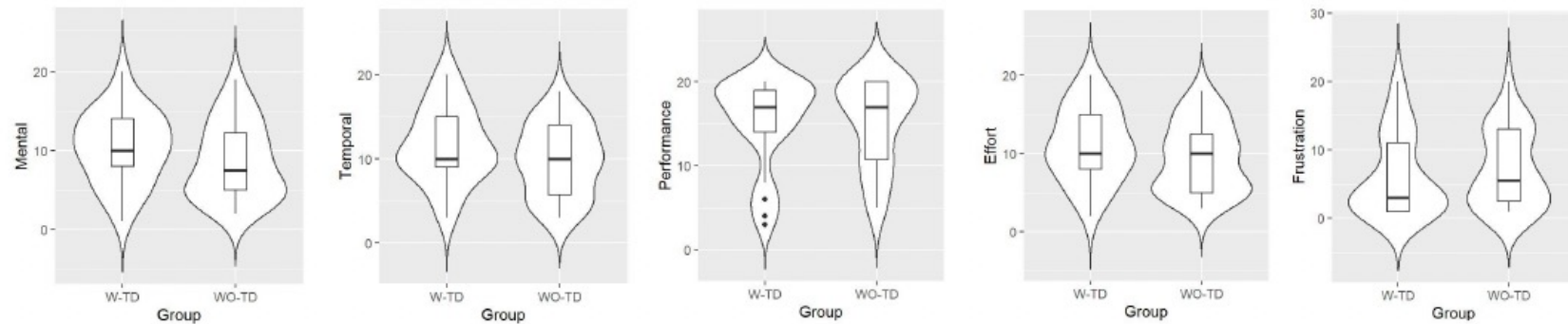
Prompt: Please help me solve the following problem.  
[Put all instruction in]  
Prompt: Please use Python, input N, M, K, output the exit order. Use ACM mode.  
Prompt: The output has problem, for example, if let N, M, K = 6, 5, 2, expected output is 5, 4, 6, 2. But your program's output is 5, 1, 4, 3, 2, 6  
Prompt: You change the index to m-1, could you explain it?  
[Continued communication until problem solved]

b. The prompt history of a participant maintaining active communication with ChatGPT.

# NASA-TLXの結果



a. The distributions of NASA TLX scores in *Coding Puzzles* with(out) ChatGPT



b. The distributions of NASA TLX scores in *Typical Development Task* with(out) ChatGPT