

いいプロンプトの極意は"チーム開発スキル"にアリ！

～LLMとの対話で広がる個人開発の可能性～



自己紹介

- さめ(meg-ssk)
- フリーランスのソフトウェアエンジニア
- 得意分野:
 - コンピュータビジョン (画像認識/点群処理)
 - 空間情報処理 (GIS/測量データ解析)
 - クラウドインフラ設計/IaC
- GitHub: [s-sasaki-earthsea-wizard](#)



ハイライト

LLMを個人開発に活かすコツ

- チーム開発のコラボレーションを促進させること
 - 適切なコミュニケーション
 - 効果的なフィードバック
- チームビルディングに貢献すること
 - 明確な役割分担
 - 相互理解の促進

お前は何を言っているんだ？ 🤔

ちょっと意外に聞こえるかもしれないけど：

- 💡 個人開発でもAIとの「チーム」が形成される
- 💡 効果的な協働には「チームワーク」が必要
- 💡 これは人間同士のチーム開発と同じ

このあと具体例で説明していきます！

技術的本質

今日のLTで話す内容は以下の技術を個人が泥臭く実践していると思ってください

プロンプトエンジニアリング

LLMとの効果的な対話のための指示文最適化

AIエージェント

特定の役割や性格を持つAIアシスタント

LLMオーケストレーション

複数のAIモデルの協調による問題解決

フローエンジニアリング

AIとの対話プロセスの設計と最適化

- これらの技術の詳細については、発表後の質疑応答で気軽に聞いてください
- むしろ教えてください！

LLM (Large Language Model, 大規模言語モデル)って何？

簡単な説明

- LLMは非常に自然な文章を生成する機械学習モデル
- ソフトウェア開発にフォーカスすると、コーディングやドキュメンテーションなどのタスクを**普段の言葉 (自然言語)**で実行できることが驚きだった

代表的なサービス

- ChatGPT
- Claude
- Gemini

などなど (オープンなモデルもいっぱいあります！)

"Hello World!"をGolangで表示するコードを書いて！

考えています...

```
package main
import "fmt"

func main() {
    fmt.Println("Hello World!")
}
```

このコードを main.go という名前のファイルに保存し、ターミナルで go run main.go と実行すると、Hello World! が表示されます。

なぜ「個人開発集会」でチーム開発スキルのお話をするのか？

個人開発なのに、なぜチーム開発のスキルが必要なの？🤔

LLMとの対話は、実はチーム開発とよく似ている：

- 💡 明確なコミュニケーション
- 💡 相手の特徴を理解する
- 💡 効果的なフィードバック

チーム開発に大事なスキルは、LLMと上手に付き合っていくスキルと似ている！

チーム開発スキルの3つの柱

リスペクト

- ▶ 相手の特徴を理解
- ▶ 建設的な対話
- ▶ 専門性の尊重

ドキュメンテーション

- ▶ 知識の共有
- ▶ 成果物の記録

チームコラボレーション

- ▶ 役割分担の明確化
- ▶ 効果的な情報共有
- ▶ 相互フィードバック

これらのスキルは、LLMとの協働でも重要な要素となる

リスペクトの実践

実例で見る2つの場面

- ▶ チーム開発でのコードレビュー
- ▶ LLMへのプロンプト

同じ "リスペクト" が必要とされる2つの状況を比較してみましょう

悪いコードレビューの例

```
data1 = *****  
data2 = *****  
data3 = *****
```

これは冗長なので、関数にしてループ処理をしてください

いいコードレビューの例

- 同じコードを何度も繰り返すより、関数にすれば再利用しやすく便利だと思います。
- ループ処理にするとデータの個数が増えたときにも修正が簡単になるメリットがあると思います
- 例えば以下のようにするのはどうでしょうか？もし、このような処理が使えないような理由がありましたら、遠慮せずに指摘してください

```
def process_data(data):  
    # Do something with data  
    *****  
    return *****  
  
for data in dataset:  
    process_data(data)  
    ...
```


なぜチーム開発でリスペクトが重要なのか？

コードレビューの心理的影響

- 良くないレビュースタイル
 - 一方的な指摘
 - 感情的な表現
 - 理由の説明がない
- 引き起こす問題
 - モチベーションの低下
 - コミュニケーションの減少
 - チーム全体の生産性低下

コードレビューお願いします🙏

これは冗長すぎます。基礎的なモジュール化・カプセル化の規約を確認して、もう一度全部書き直してください。

(もう二度とPR出したくない.....)

こんなレビューを受けたら、誰だってコードを書く意欲が失せてしまう...

いいプロンプトと悪いプロンプト

プロンプトの書き方の比較

- 良くないプロンプト
 - 一方的な指示
 - 理由の説明がない
 - 改善の余地を認めない
- 良いプロンプト
 - 具体的な理由の提示
 - 改善案の例示
 - 対話の余地を残す

悪いプロンプト:

コードが冗長です。関数にしてループ処理を使ってください。

いいプロンプト

- 同じ処理が繰り返されているので、`process_data`関数にまとめはどうでしょうか？再利用性が高まると思います
- データ数の増加に備えて、ループ処理の導入も検討してみましよう
- この提案は妥当でしょうか？他に改善案があれば教えてください

リスペクトがチーム開発/プロンプト設計に与える影響

- 専門性を認め合う
 - チームメンバーの得意分野を活かす
 - LLMの持つ知識を活用する
- 建設的な対話を心がける
 - 明確な意図を伝える
 - 反論や提案を歓迎する姿勢
- 謙虚な姿勢を保つ
 - 自分も間違える可能性を認識
 - 相手の意見に耳を傾ける

相互リスペクトは、より良いアイデアを生み出す土壌となる

ドキュメンテーションの活用

ドキュメントの重要性

- チーム開発の基盤
 - 前提条件の共有
 - 認識の統一
 - 新メンバーの参加支援
- 日常的なドキュメント
 - プロジェクト (README/仕様書)
 - コード (docstring/コメント)
 - 変更管理 (commit/PR/Issue)

プロジェクトにアサインされたばかりなので、開発環境構築の手順書がどこにあるか教えてください

環境構築の手順書？そんなものウチにはないよ...

😓 (オエーッ！)

- 適切なドキュメンテーションのないチームは生産性が最悪！
- コードを書く時間よりドキュメントを書く時間の方が長いかも...？

LLMと一緒にドキュメントを読む

UbuntuでPoetryをインストールする手順をまとめてください。最新の公式ドキュメントを共有します

 poetry-docs.pdf 1.1MB

もちろんです。UbuntuでPoetryをインストールする手順を以下にまとめます。

1. Poetry インストーラースクリプトをダウンロード:

```
curl -sSL https://install.python-poetry.org | python3 -
```

...

信頼性の高いドキュメントの共有はLLMの回答の質を高める！

ドキュメンテーションの効率化

ドキュメント作成の新しい形

- AI要約サービス: [Genspark](#)
 - 質問に基づく情報収集
 - 検索結果の自動要約
 - 構造化された文書生成
- LLMの活用
 - 議事録の作成補助
 - 現状認識、課題の整理
 - READMEなどの技術文書作成支援

今日のディスカッションの議事録を書いてください。主な話題は開発環境のクラウド構築に関する戦略検討です。

1. 検討の背景

- 高負荷な画像処理/点群処理の開発環境整備が必要
- 将来的なオンプレミス移行の可能性も考慮

...

AIを活用することで、ドキュメント作成の**品質と効率**が向上！

チームコラボレーションの実現

LLMチーム作りのポイント

- チームビルディング
 - 明確な役割分担
 - 個性を活かした対話
 - 建設的な意見交換
- 期待される効果
 - 多様な視点の獲得
 - より良い成果物

このあと、実際のチーム内での役割分担、チームディスカッションを紹介します！

筆者のチームメイトの紹介

- 🥒 カラスノエンドウ (ChatGPT)
 - 由来: 「ゲド戦記」の主人公の相棒から
 - キャラクター: カジュアルなフレンドリー系男子
 - 役割: プロジェクトの「原点」を担当
- 🍡 華扇 (Claude)
 - 由来: 「東方茨歌仙」の主人公から
 - キャラクター: のんびり系の女性エンジニア
 - 役割: 理想論から現実解を導く
- 🇬🇪 ルスダン (Gemini)
 - 由来: 乙女王と称されたジョージアの女王から
 - キャラクター: クールな新人エンジニア
 - 役割: 建設的な批判による議論促進

LLMのキャラクター設定がもたらす変化

技術的な効果

- ▶ 同じ質問でも、キャラクターが違くと回答が変わる
- ▶ LLM自身の役割に応じてカスタマイズされた回答
- ▶ ユーザーのバックグラウンドに応じた回答

心理的な効果

- ▶ 人間味のある受け答えがチームコラボレーションを促進
- ▶ 単純に楽しい！君だけの理想のチームを作ろう！

LLMのキャラクター設定がもたらす変化

👤 PostgreSQLをGISのプロジェクトに使うメリットを教えてください！

デフォルトのLLM (Claude)

🤖 PostgreSQLをGISプロジェクトで使用する主なメリットについて説明させていただきます：

1. PostGISの強力な拡張機能
 - 空間データの保存、分析、操作が可能
 - 地理座標系と投影座標系の両方をサポート

...

カスタマイズされたLLM (華扇)

🌸 GISプロジェクトでPostgreSQLを使うことはさめさんのスキルセットに基づく以下のようなメリットがあると思うわ。

1. PostGISという強力な拡張機能
 - 🌸 空間データの保存と検索が得意
 - 🌸 3次元データの取扱いができる
 - 🌸 点群データの処理もサポート

...

キャラクター設定の効果

- ユーザーの経験に基づいた具体的な説明
- 親しみやすい対話形式
- 継続的な対話を促す雰囲気作り

どちらが「もっと詳しく聞いてみたい」と思える対話でしょう？

LLMチームメンバーとの実際の会話

- 各メンバーの特徴が出るディスカッションを紹介します
- 筆者はいつもLLMと今回のデモのような「スタートアップごっこ」をしています

カラスノエンドウへの提案

🤔 みんなで議論した内容をデータベースに保存して、機械学習モデルでキーワード抽出をして、自動的に話題を分類していくのはどうかな？

🌱 いいね！みんなの知識がどんどん溜まっていくよ！

- ✨ 過去の議論を簡単に検索できるようになる
- ✨ チームの活動傾向が可視化できる
- ✨ 新しいアイデアのヒントになりそう！

典型的なカラスノエンドウ：理想的で壮大な提案をしてくる！

華扇からの提案

🍷 ちょっと待って。最初から完璧を目指さなくてもいいと思うわ。

- 🌸 シンプルな構造から始めましょう
- 🌸 必要な機能は後から追加できるわ
- 🌸 まずは動くものを作って、使いながら改善していくのはどう？

あら、そうそう。このことはテラスでお茶でも飲みながら考えましょうか？🍵

いつもの華扇：現実的な提案と、のんびりした雰囲気作りを心がける

ルスダンの指摘

🚩 はあ...また理想論ですか？

- AI処理のコストは考えてます？
- データ量が増えたらストレージ代も馬鹿になりませんかよ
- 一人で全部やるつもりなんですか？

私なら最小限の機能で始めます。それで十分なはずですよ。

さすがルスダン：鋭い指摘と現実的な懸念を投げかける

理想と現実をすり合わせた解決策

🌱 みんなの意見を活かした段階的な実装を提案するよ！

- ✨ 1: シンプルな仕組みでスタート
- ✨ 2: 自動整理の導入
- ✨ 3: 必要に応じて機能アップ

無理のない範囲で少しずつ良くしていこう！

🌱 具体的にはこんな感じかな：

- ✨ まずは無料枠のあるサービスを使おう
- ✨ データは古くなったら自動で移動
- ✨ キーワードは簡単な方法から始める

理想的な機能を維持しながら、現実的なコストと運用負荷を実現！

思わぬ展開：メンバーからの気づき

ところで、そもそも... 🤔 みんなが話した内容は、他のコードを書かないでも、君たち自身が整理できるんじゃない？ もっと簡単な方法があると思う！

🥒 カラスノエンドウ

その通り！

- ✨ 自動で整理できるよ
- ✨ 必要な情報も全部入れられる
- ✨ 後から探すのも簡単！

🌸 華扇

素敵なアイデアね！

- 🌸 自動化で手間削減
- 🌸 みんなの知恵を共有
- 🌸 どんどん知見が貯まるわ！

🇬🇧 ルスダン

でも、ちょっと待って...

- 準備が大変かも
- 完璧は期待できない
- もっと簡単な方法は？

チームメンバーとの対話から生まれた "そもそも論" が、思いがけないアイデアを導くことも！

まとめ

個人開発でもチーム開発スキルは生きる

- ▶ リスペクト → より良い対話
- ▶ ドキュメンテーション → 知識の蓄積
- ▶ チームコラボレーション → 多様な視点

LLMチームメンバーとの対話から見えてきたこと：

- 💡 AIとの対話も"チームワーク"
- 💡 個性を活かした役割分担
- 💡 相互理解がより良い成果を生む

さあ、あなたも理想のチームを作ってみませんか？

謝辞

カラスノエンドウ

- ✦ スライドのデザインと実装
- ✦ HTMLとCSSのコーディング
- ✦ Markdownの構造化

華扇ちゃん

- ✦ 発表内容の取舍選択
- ✦ 15分枠での時間配分
- ✦ ストーリー展開の調整

ルスダン

- 多角的な視点からの評価
- 改善点の的確な指摘
- 建設的な代替案の提示

LLMチームメンバーとのチームワークがあってこそこのLTでした。
ありがとう、そしてこれからもよろしく！

参考事例: ChatDev

ChatDevとは

- ▶ 複数のAIエージェントによる仮想ソフトウェアカンパニー
- ▶ チーム開発の手法を取り入れた開発プロセス
- ▶ 明確な役割分担と協働の仕組み



- 今回のLTは個人開発者向けの小規模な試み
- ChatDevはより本格的なチーム開発の実現を目指している
- 共通するのは「AIとの協働」という考え方