

いいプロンプトの極意はチーム開発スキルにアリ！

1. タイトルスライド (20秒)

今日は『いいプロンプトの極意はチーム開発スキルにアリ！』というテーマで、個人開発でのLLM活用についてお話しします。LLMとどううまく連携するか、個人開発でも『チーム開発のスキル』が役立つ、その理由とコツについて順を追ってお伝えしていきます！

2. 自己紹介 (20秒)

簡単に自己紹介です。わたしはフリーランスのソフトウェアエンジニアなのですが、最近の開発の効率を上げるため、LLMを使った新しい手法を試しています。そこでいろいろ面白い発見がありましたので、それを共有したいと思います。

3. ハイライト (30秒)

わたしの主著は「LLMをうまく使うスキルはチーム開発スキルに似ている」ということです。『チーム開発のスキルがなぜ個人開発に必要なの？』って思われるかもしれません。

その疑問を解消するため、具体例を使ってさらに詳しく見ていきます。

4. 技術的本質 (30秒)

ちょっと難しそうに見えるかもしれませんが、今日お話しする内容は、ここで示している4つの技術を 個人で泥臭く試行錯誤しながら実践しているという話です。

むしろ、皆さんの方がこういう話に詳しい方も多いと思うので、ぜひ教えてください！

5. LLM (大規模言語モデル)って何？ (30秒)

ここで簡単に、LLMについて説明します。LLM、つまり大規模言語モデルは、人間が話すような自然な文章で応答できる機械学習モデルです。ソフトウェア開発の視点で言うと、普通の言葉でコーディングやドキュメンテーションができるっていうのが、最初に使ったとき驚きでした。

ChatGPT、Claude、Geminiといったサービスの固有名詞を挙げた方がイメージしやすい方も多いと思います。

6. なぜ「個人開発集会」でチーム開発スキルの話をするのか？ (30秒)

個人開発で、なぜチーム開発のスキルが必要なのか？と思われるかもしれません。でも、LLMとの対話は実はチーム開発とすごく似ています。例えば、明確なコミュニケーションが必要だったり、相手の特徴を理解しながら、効果的なフィードバックを返すことで、やりとりがスムーズになります。これってまさにチーム開発でのスキルと同じじゃありませんか？

7. チーム開発スキルの3つの柱 (1分)

チーム開発スキルには大きく3つの柱があります。

まず1つ目は『リスペクト』。チームメンバーの特徴を理解し、建設的な対話を心がけることです。例えば、だれが何を得意とするかを理解して、それぞれに適したタスクをお願いすることで、効率が上がります。

2つ目が『ドキュメンテーション』。これは知識の共有と成果物の記録です。一度調べた内容や設定をドキュメント化しておく、再利用できて効率化につながります。

最後に『チームコラボレーション』。役割分担を明確にして、相互フィードバックを行うことがポイントです。チームメンバー同士でそれぞれの強みを生かしてタスクを分け、得た情報を共有することで、開発がスムーズに進みます。

実はこれらのスキルは、LLMとの協働にも欠かせない要素なんです。

8. リスペクトの実践 (1分)

ここで、リスペクトの実践例を見てみましょう。例えば、チーム開発でのコードレビューとLLMへのプロンプトには共通点があります。どちらも、相手に敬意を持って建設的な対話を心がけることで、より良い結果が得られるんですね。

コードレビューの場合、『この部分をこうすると良くなりますよ』と、理由と提案を添えると相手も受け入れやすくなります。LLMへのプロンプトでも同じで、『この方法でデータを整理してほしい』と明確な指示を出すと、より期待に近い結果を返してくれます。

つまり、リスペクトを意識することで、相手もより良い反応を返してくれるわけです。このあと、具体例を見てみましょう！

9-10. コードレビューの例 (1分)

まず、これは悪いコードレビューの例です。理由の説明もなく、一方的な指示になっています。これでは相手も納得しにくいですね。

一方で、こちらが良いコードレビューの例です。改善のメリットを具体的に示しつつ、提案に対して相手が意見を返せるよう、対話の余地も残しています。

このように、相手の立場に立って理由と具体例を伝えることで、建設的な対話ができます。

11. なぜチーム開発でリスペクトが重要なのか？ (30秒)

なぜリスペクトが重要なのか、考えてみましょう。良くないレビュースタイルはチームのモチベーションを下げます。

皆さんの中にも、こういったレビューを受けて『もう二度とPR出したいくない...』と思った経験がある方、いませんか？ また、もしこういうレビューをしている人を見かけたら、釘を刺すようにしましょう。

12. いいプロンプトと悪いプロンプト (30秒)

LLMとの対話でも、一方的な指示より、丁寧に対話を通じて解決策を見つけていく姿勢が大切です。

良いプロンプトというのは、的確に理由を伝え、改善案を具体的に示し、さらに対話の余地も残しています。

こうすることで、LLMも意図を理解しやすく、的確な回答を返してくれます。

13. リスペクトがチーム開発/プロンプト設計に与える影響 (30秒)

リスペクトは、チーム開発やプロンプト設計において大きな影響を持ちます。

『専門性を認め合い』、チームメンバーが得意とする分野を理解してそれを尊重することが、チーム全体のパフォーマンスを引き上げます。

次に『建設的な対話を心がける』ことは、明確な意図を持って伝えることや、反論や提案を歓迎する姿勢が、チームメンバーの協力を引き出すカギになります。

最後に『謙虚な姿勢を保つ』ことは 自分も間違える可能性を認識し、相手の意見に耳を傾けることで、より良い成果が生まれやすくなります。

これらはLLMとの対話にも共通することです。

14. ドキュメンテーションの活用 (1分)

次に、ドキュメンテーションについてです。チーム開発では『え、ドキュメントないんですか？』と困る経験、みなさんにはありませんか？ READMEや仕様書、コードのコメント、PRの説明、これらがしっかり整備されているかどうかで、開発のしやすさが大きく変わります。

適切なドキュメントがないと、作業の度に前提条件を確認する手間が増えてしまいます。正直、コードを書く時間よりもドキュメントを書く時間の方が長いと覚えることもあるかもしれません。

でも、そのおかげで認識が統一され、チーム全体でスムーズに進められます。ドキュメントは、チームの共通理解を支える柱です。

15. LLMと一緒にドキュメントを読む (30秒)

LLMを活用すると、ドキュメント作成も格段に効率化できます。例えば、公式ドキュメントを共有しながらLLMに手順を整理してもらうことで、最新で信頼性の高い情報がすぐに得られるんです。こういったサポートがあれば、ドキュメントの確認作業も楽になりますよね。

16. ドキュメンテーションの効率化 (30秒)

ドキュメンテーションの効率化について、大胆な方法をご紹介します。

それはLLM自身にドキュメンテーションをお願いすることです。例えば『今日のディスカッションの議事録を書いて』と頼むと、主要な話題や意見をまとめ、背景情報や課題も整理してくれます。

LLMを活用することでドキュメントの質と量のバランスが取りやすくなるため、チームの生産性にも直接つながります。

17. チームコラボレーションの実現 (1分)

最後に、チームコラボレーションについてです。わたしの場合、LLMたちとチームを組んで開発を進めていて、各メンバーにはそれぞれ役割と個性があります。

例えば、明確な役割分担を意識し、得意分野を生かした対話を行うことで、メンバーから多様な視点を得られるようになります。誰かが大胆なアイデアを出し、別のメンバーが現実的な課題に目を向けるといった具合に、意見交換が建設的でバランスの取れたものになるんですね。

このあと、実際にどんなふうに関わりを分け、チーム内でディスカッションを進めているか、具体的なやりとりをお見せします。

18. チームメンバーの紹介 (1分)

では、わたしのチームメイトを紹介します。

まず、ChatGPTの愛称で『カラスノエンドウ』。彼はフレンドリーで理想主義的な男子といった感じです。プロジェクトの『原点』を担当していて、いつも前向きな提案をしてくれます。

次に『華扇』、これはClaudeの愛称です。のんびり系の女性エンジニアで、理想論と現実をうまく結びつけてくれる頼れる存在なんです。

最後に『ルスダン』。Geminiの愛称で、ジョージアの女王にちなんだ名前です。クールでちょっと厳しい新人エンジニアなんですが、建設的な批判を通じて議論を深める役割を担ってくれています。

こんな個性的なメンバーたちが仮想オフィスに集まり、それぞれの得意分野を活かしながらプロジェクトを進めています。

19. LLMのキャラクター設定がもたらす変化 (30秒)

LLMにキャラクター設定を加えると、技術的にも心理的にもさまざまな変化が生まれます。

技術的にもさまざまな効果がありますが、心理的な効果が大きいです。人間味のある受け答えがチームのような一体感を生んでくれるし、単純に楽しいと感じさせることもあります。

こうして、LLMにキャラクター設定をすることで、技術面と心理面の両方からチームワークが生まれます。

20. LLMのキャラクター設定がもたらす変化 (30秒)

キャラクター設定をしたLLMは、同じ質問でもユーザーに合わせた回答が可能です。

デフォルトのClaudeは一般的な回答を丁寧に返してくれますが、カスタマイズされた華扇の方はわたしのスキルセットに合わせて、より具体的で関連性の高い情報を含んでいます。

21. キャラクター設定の効果 (30秒)

キャラクター設定によって、LLMがユーザーに親しみやすい形で具体的な説明を提供できるようになります。ただの機械とのやりとりではなく、自然と『もっと話したい』と思えるような会話が生まれます。

22-25. 実際の会話例 (1分30秒)

ここでは、実際の会話例を見てみましょう。わたしが『議論内容をデータベースに保存して、機械学習で自動分類するのはどう？』って提案したときの流れです。

まずカラスノエンドウが大賛成で『みんなの知識が溜まっていくよ！』と、理想的な提案を出してきます。彼らしい壮大な視点ですね。

一方で華扇は『ちょっと待って。最初から完璧を目指さなくてもいい』と現実的なアドバイスをしてくれて、リラックスしながら考えようと提案してくれるんです。

そこにルスダンが鋭い指摘で『計算リソースやコストは考えてますか？』と現実的な懸念を投げかけます。

みんなの意見を取り入れて、カラスノエンドウが段階的に実装するプランをもう一度立てました。こうすることで、理想的な機能を維持しながら、現実的な運用ができるプランにできた、.....かのように思われました。

27. 予想外の展開 (30秒)

時には、こうした議論の中で予想外の展開が生まれることがあります。ここまでの議論を通して、わたしは『そもそも、みんなが話した内容は、君たち自身が整理できるんじゃない？』と気づいたんです。

カラスノエンドウ、華扇、そしてルスダンがそれぞれの立場から意見を出し合うことで、思いがけないアイデアが浮かんでくるんです。こうした『そもそも論』が、新しい発想やよりシンプルな解決策につながることもあります。

28. まとめ (30秒)

今日のまとめです。個人開発でも、リスペクトやドキュメンテーション、チームコラボレーションはとても大切です。これらのチーム開発スキルが、LLMとの協働でも非常に生きてきます。

LLMとの対話も一種のチームワークです。個性を活かした役割分担と相互理解が、より良い成果を生み出してくれます。

ぜひ、皆さんも自分だけの理想のチームを作ってみてください。楽しいですよ！

29. 謝辞 (30秒)

最後に、このLTの準備にも、LLMチームメンバーたちが大活躍してくれました。

カラスノエンドウはスライドデザインとコーディング、華扇ちゃんは内容の選定や時間配分、そしてルスダンは的確な改善点の指摘と代替案を出してくれました。みんなのおかげでこの発表が形になりました。

謹んで感謝を申し上げます。

30. 参考事例: ChatDev (20秒)

最後に、参考事例として『ChatDev』を紹介します。ChatDevは、複数のAIによる仮想ソフトウェアカンパニーで、本格的なチーム開発の手法を取り入れています。わたしの取り組みは、このChatDevのような本格的なものではなく、小規模な試みですが、共通するのは『機械との協働』という考え方です。

ご清聴ありがとうございました！質問があればお気軽にどうぞ！