

AI時代に求められる エンジニア像と、AI活用の勘所

エンジニアの価値創出とAIとの協調による持続可能な成長戦略



わたり

- 現在はPOS開発
- AIツール
 - 利用歴は約2年半
 - 転職とほぼ同時期
 - ChatGPTが公開されてすぐくらい
 - 社内
 - AI利用のルール作り / AIツール価値検証や布教
 - プロダクトのAI開発最適化
 - プライベート
 - Altuber作成(配信は10回くらい)

AI時代に求められるエンジニア像



AIに置き換えられやすいエンジニア

- 「必要な情報を揃えて渡して」と中心業務以外を避ける
- 情報を揃えたらあとはAIで実現可能
AIに任せた方が効率的



価値を発揮するエンジニア

- AIへの指示出し＝**マネジメント力**を発揮
- コンテキストを踏まえ **自律的に業務を進める**
- 少ない指示で広く動ける **総合力**を持つ

これから残るのは「AIを動かせる人」

状況を理解し、自ら考えて行動できる**総合力**と**行動力**がエンジニアにとって重要な資質となります。
AIという優秀な部下に的確な指示を出し、その能力を最大限に引き出せる人材が価値を高めていく
「**エンジニア総マネージャー時代**」の到来です。

AI時代にエンジニアが価値を出せる領域



AIが得意とする領域

⚡ 短期案件

</> 定型業務

📦 独立したタスク

- 「作って納品して終わり」の短期案件
- 新規の小規模システムをゼロから構築
- 明確な仕様が定義されている開発作業

💡 AIは単発的・短期的なタスクに強いが、長期的な価値や複雑な要素が絡む判断は苦手



人間が価値を発揮する領域

🕒 長期運用

🗨️ 複雑なコンテキスト

👥 関係者調整

- 長期のシステム運用と継続的な改善
- ステークホルダー間の交渉や調整が必要な案件
- システムが成長し、関係者が増え、明文化されていないものが複雑に絡み合う中での判断

👤 人間のエンジニアは複雑な状況判断と長期的な視点を持った意思決定に強み

AIとの協調: AI が理解しやすいコードを書く ポイント



無駄な独自実装を避ける

既存ライブラリと役割が重複するメソッドは極力作らない。標準的な実装方法を採用することでAIが元々持っている知識で対応でき、コードの意図を理解しやすくなります。



暗黙知を減らす

「この部分を修正する際は、あちらも変更する必要がある」等の暗黙のルールは、**コメントとして明記**します。調査や質問を通じて判明したことも、今後のためにコメントに残しましょう。



適切な命名

誤解を招く名前はAI / 人間ともにバグの温床になります。**意図が明確に伝わる命名**を心がけ、コードの可読性と保守性を高めましょう。

人間にとって理解しやすいコードは、AIにとっても解析・拡張しやすいコードです。明確で一貫性のあるコードは、開発効率を向上させ**AI との効果的な協調**にも繋がります。

プロンプトエンジニアリングは「知識」であって「職種」ではない



知識としてのプロンプト技術

プロンプトエンジニアリングの知識は有用ですが、開発業務の中心になるわけではありません。AIモデルによって最適な指示の出し方が異なるのは、**人間に対するコミュニケーション**と同様です。

💡 効果が検証された技術を活用することが重要であり、**小手先のテクニック**に頼るべきではない

余計な情報を与えず、**簡潔に指示を出す**ことが、現状ではAIの精度を高めるコツとなる場合があります。情報量が多すぎると本質が見えにくくなり、AIの回答が浅くなる傾向があります。



モデルによって異なる最適な指示方法



一部のモデルに有効

GPT-3.5/4など

▶ ステップバイステップで考えて

≡ 箇条書きで回答して



むしろ避けるべき例

GPT-o1（公式ドキュメント記載）

▶ ステップバイステップで考えて

⊘ 過度に詳細な指示

“ プロンプトテクニックは「魔法の呪文」ではなく、**コンテキストに応じた適切なコミュニケーション手法**です。AIに対する指示も、最終的には相手に合わせたコミュニケーションが鍵となります。

”

AIツールとの賢い付き合い方



軸足を早めに決める

満点を求めて検討を繰り返すより、**信頼できる軸足となるツールを早めに決める**方が効率的です。検証や乗り換えのコストも無視できません。



プロ驚き屋に振り回されない

まだ実務では使えない技術を過剰に持ち上げる「プロ驚き屋」の情報発信に惑わされず、**本質を見極める**ことが重要です。



情報収集を効率化する

SNSや技術ブログで**名前をよく見るようになってから検証**しても十分間に合います。信頼する企業や開発者の採用状況を判断基準にしましょう。

📁 開発に関わる AIツールの種類(私見)

💬 チャット系: ChatGPT, Gemini, Claude

📄 エディタ統合系: Cursor, Windsurf

🤖 自律Agent系: Devin, CodeRabbit

AIで楽をしたいのに、AIに関する情報収集で疲弊してしまっては本末転倒です。

💡 **最初は多くの人が騒ぎ立て、その後実際に有用なものだけが淘汰されて残っていく**傾向があります。

焦らず、自分のペースで取り入れていきましょう。

まとめ: AI時代のエンジニアが生き残るために

求められるエンジニア像

- ✓ 少ない指示で広く動ける「考える総合力」と「行動力」を持つ
- ✓ AIへの指示出し・マネジメント能力を備えた「マネージャー」としての役割
- ✓ コンテキストを理解し、自律的に業務を進められる能力

エンジニアの価値創出領域

- ✓ 長期運用と複雑なコンテキスト管理が必要な領域
- ✓ システムの成長と関係者間の調整が求められる状況
- ✓ 過去の経緯を踏まえた継続的な改善が必要な業務

AIとの協調方法

- ✓ 無駄な独自実装を避け、標準ライブラリを活用する
- ✓ 暗黙知を形式知化し、コードにコメントとして残す
- ✓ 意図が明確に伝わる適切な命名を心がける

AIツール活用の知恵

- ✓ プロンプトは知識であって職種ではない
- ✓ 満点を求めず軸足となるツールを早めに決める
- ✓ 過度な宣伝に振り回されず本質を見極める

AI時代のエンジニアは、人間ならではの判断力と総合力を武器に、AIを使いこなすことで価値を創出できます。
AIは「競争相手」ではなく、「道具」であり共に成長する優秀な部下であると考えましょう。