LLM を使って生産性を上げる

自己紹介

- 株式会社コルモアナ
- ソフトウェアエンジニア
- 谷口幸宏
- 寿司
- 肉
- 酒

	ユーザー数	エンジニア数
	4 億 5000 万人	32 名
	3000 万人	13 名
Dropbox	10 万人	1 名

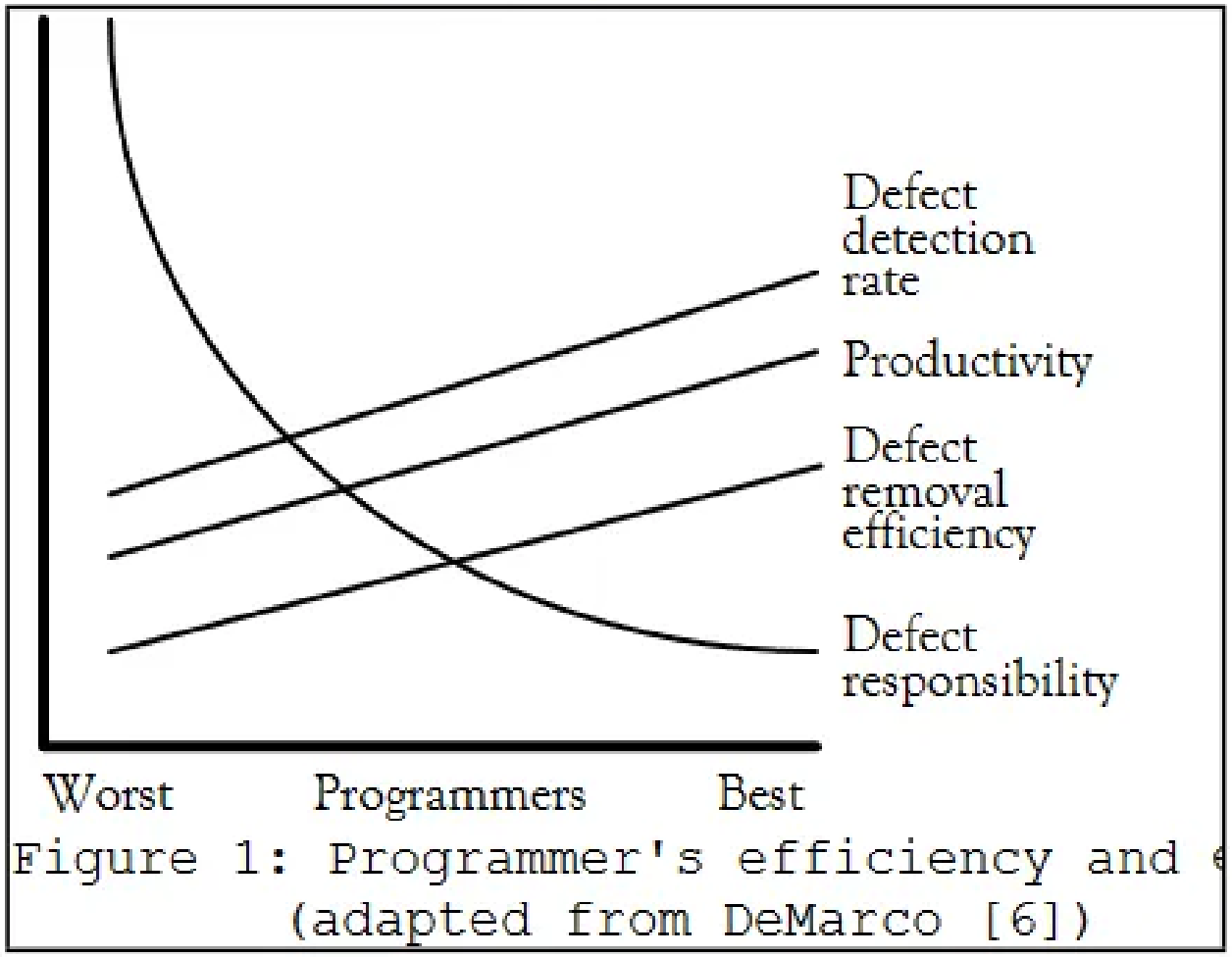

Figure 1: Programmer's efficiency and
(adapted from DeMarco [6])

- 横軸: プログラマーのレベル
- 欠陥検出率 (Defect detection rate):
- 生産性 (Productivity):
- 欠陥除去効率 (Defect removal efficiency):
- 欠陥原因 (Defect responsibility):

Refs: The origins of the 10x developer

なんですか?

- LLM を使ってエンジニアとしての生産性を上げる話をみんなでしたい
- 本当に人の何十倍も生産性の高いエンジニアは存在するのか?
  - 存在する
- まずはエンジニアの生産性について考える必要がある

10x プログラマーという神話 by antirez

「一流のプログラマーは、普通のプログラマーの 10 倍の生産性を持つ」

- antirez さんのブログエントリー
- antirez さん
  - Salvatore Sanfilippo（通称 antirez）
  - オープンソースソフトウェアの開発者
  - Redis の Founder
  - 6 歳の頃からプログラミング

Refs: The mythical 10x programmer

10x プログラマー

- 1968 年の研究
- 最も優れたプログラマーが他のプログラマーよりも 10 倍生産的であることが示された
- 「人月の神話」で紹介され、広く知られるようになった
- プログラミングの複雑さやチームワークの重要性はあまり考慮されていない

10x プログラマー

> 非常に優秀なプロのプログラマーは、下手なプログラマーの 10 倍の生産性がある。
> たとえ、同じ訓練と 2 年間の経験を経ているとしても。

Refs: 人月の神話 - Wikipedia

生産性において大きな違いを生み出す資質

- 基本的なプログラミング能力
- 経験
- 集中力
- 設計の妥協
- シンプルさ
- 完璧主義
- 知識
- 低レイヤの理解
- デバッグスキル

Refs: The mythical 10x programmer

LLM を使ってどのくらい改善可能性があるか?

資質	LLM による改善可能性	
基本的なプログラミング能力	◎	明らかに人間よりできる。ほぼ完璧
経験	○	多様なパターンを提案してくれるが、完璧とは言えない
集中力	△	LLM に時間管理のアドバイスをお願いできるが、まずやらない
設計上の妥協	○	トレードオフの分析ができるが、最終判断には人間が判断することになる
シンプルさ	○	シンプルなソリューションを提案できるが、仕様が壊れることもある
完璧主義	△	効率的アプローチを提案できるが、人間の個人の性格特性に依存
知識	◎	枯れた技術に関してはほぼ完璧だが、新しい技術に関してはそれほど強くない
低レイヤの理解	○	完璧だが、人間が低レイヤに対する理解を持っていることが重要
デバッグスキル	○	LLM は一般的なバグパターンの識別や修正方法を提案できるが、複雑なバグ解決には人間の経験や直感も重要

各項目における生産性の向上倍率を概算してみる

シナリオ	生産性倍率	理由
LLM を使わない場合（ベースライン）	x1	基準となるケース
基本的なプログラミング能力にのみ LLM を使用	x2-x3	コード生成の高速化 基本的なアルゴリズムの迅速な実装 エラーの減少
すべての項目に LLM を適用	x5-x10	基本的なプログラミング（x2-x3） 経験に基づくパターンマッチング（x1.5-x2） 設計の最適化とシンプル化（x1.5-x2） 理論的知識の利用（x1.3-x1.5） デバッグ時間の短縮（x1.2-x1.5）

10x は可能?

- やはり問題はある
- 最新の技術について LLM が知らない
  - 教えてあげないといけない
  - 面倒なのでやらない
- 複雑な仕様や大きなコードベース
  - 教えてあげないといけない
  - 教えてもコンテキストが大きすぎてうまくいかない
  - そもそも LLM に投げてよいのか問題(BYOAI)
- ソフトスキル
  - コミュニケーションスキル
  - リーダーシップ
  - チームワーク

10x は場合による

> ある状況では 1.5 倍、時には 100 倍の能力があるということです。
> しかし、ソフトウェアの基盤の構築後では、ソフトウェアは職人の熟練の技の成果というよりは、Lego に似てきます。
> そこにおいて、基盤をもとに「10x」エンジニアが取り組んでいる仕事の成果の相対的倍数は、1x に近づきます。

Refs: 10x エンジニアの幸福な終焉 (a16z)

結論

- LLM を活用すれば確実に 10x の生産性を発揮することができる状態を作ることができる
- ただし、その状態は場合による
- だからといって、LLM を使わないのはあまりにももったいない

今からできる 10x の LLM 活用

- 設計
- コード生成
- デバッグ
- ドキュメント作成