AIはあなたを
強くしない。

手の早い分身だ。

～チーム開発の型をAIに移植～

AIで増えるのは能力ではない。
手数と速度だ。

自己紹介

黒岩 高弘（TechnoKuRo）

合同会社 TechnoKuRo ｜文系セキュリティコンサル　& Web開発

Think Abstract → Act Realize

- 開発経験 20年以上（元警察官 →エンジニア）
  主に WebApp-Java,JS,TS,Ruby,PHP など
  PM, PL　その他技術顧問的ポジション
- ISMS/Pマーク取得支援 100社以上（審査受審企業 100%通過）
- LINEオープンチャット「フリーランスエンジニア」運営（　990名）

要件定義から開発・運用・セキュリティまで。 "走攻守一体 "で伴走

今日の流れ

1. **AIは分身！あなたの力量がそのまま成果物の品質**
2. **責任は人間、分身の仕事を把握せよ！**
   - 2-1 言語化×得意活用→人にもAIにもフレンドリーなドキュメント生成
   - 2-2 ステークホルダー、AI、全員の合意形成！
   - 2-3 テスト計画もドキュメント化！型化・見える化
3. **開発は「まだ」変わらない →上流ムーブで人の価値 UP！**

誤解の整理: AIは"強化薬"ではない

❌ 誤解

- 能力/スキルが一夜で増える
- AIが勝手に成長させてくれる
- 全部任せればOK

✅ 正解

- 増えるのは「手数」と「速度」
- AIは分身＝自分の力量ままの分身
- 得意を使い、苦手を他が補助する

人間の働き方と似ている：得意な仕事を任せ、苦手な仕事は他者が補助

上流はChatGPTで"必死に詰める"（1）

ChatGPT
powered by OpenAI

■ 人間の役割

- あいまいなアイディアの具体化
- 開発のロードマップ提示
- 今何を作り、この成果物をこの後どんなアクションに使うか

■ AIの役割

「網羅性チェック」「抽出」「整理言語化」

■ やること

何を作るか/受入条件/例外/非目的を言語化

■ コツ

いきなりCLIに投げない！

上流はChatGPTで"必死に詰める"（2）プロンプト例

💡観点抽出・一般チェック（人間がネットで調べて作るイメージと同じ）

「●●が目的のWEBアプリ。■■の構成。▲の計画。君は▽を生成して。」
「人間もAIも把握しやすいmarkdown形式で、概要、目的、機能、非目的、受入条件を生成。」
「●●に似たアプリ。AIでの作成手順としてこの計画の見直しポイントを提示して。」

💡網羅性チェック

「セキュリティ/性能/運用/保守の観点で抜けを指摘して。」
「その他一般的な開発と照合して欠落があれば指摘して。」

💡成果物生成

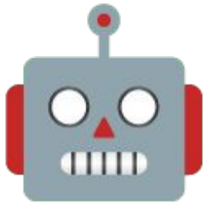
「ここの整理成果を GithubCopilot に伝えるためのプロンプトを生成して。」
「GithubCopilot が生成したmdはこれ。整合チェックして。」
「GithubCopilot に是正させるためのプロンプト作って。」

自分でネットを参考に作業するを、 AIに任せられる
チームメンバーに「レビューお願い」と頼むのと同じ使い方
ChatGPTは、ディスカッション、検索、文章生成が得意！

仕様はMD＋Git管理（1）

人間も読める

Markdown形式で
シンプル



AIも読める

抽出・構造化が得意

差分が見える

Git管理で
レビュー資産に

ExcelではなくMD

AIに網羅性チェックや観点抽出をさせやすいフォーマット

仕様はMD＋Git管理（2）開発フロー

「AIに作らせた」ではなく

「合意済みドキュメントに則って AIに作らせた」と言える

- 合意したのは人間同士で、納得している
- 納得したことが着実にコードになっている（ミス・漏れはない）
- 残りは合意漏れがあった箇所を詰めていく作業（ChatGPTに戻って繰返し）

実装はGithubCopilotに"小さく任せる"（スコープが命）

- 1回の指示は「モジュール or ファイル単位」
- 受け入れ条件と変更範囲を明記（レビュー可能にする）
- PM/PLは人間、実装者がAIに置き換わっただけ

「CLIにファイルをいじらせる前に　Github にプッシュして戻せるようにする！」

テストはCSVで"計画化"（1）

CSV

❌いきなり大量自動生成しない

✅まずテスト計画（CSV）を作る

CSV項目：

- 観点
- 優先度
- ロール
- 自動化可否
- 対象モジュール
- 完了・未完
- 課題有無

JIRAやスプレッドシートで示しやすくしておく
一覧性を持たせて置き、どこまでやったか、どこからやるか、進捗管理

テスト作りつつ、実行結果を記録させる。
バックエンドなら デシジョンテーブル などを作らせる
フロントエンドは アサーション条件を具体化するのはちょっと大変 💦

ID	実行順序	テストグループ	ロール	優先度	カテゴリ	ページ	テスト項目	操作手順	期待結果	テストタイプ	備考	実装ファイル	実装関数	実行結果	エラー内容	修正内容	2回目実行結果	2回目エラー内容	状態	次の状態	可能なアクション	検出された要素	1回目実行結果	1回目エラー内容
FLOW-001	1	Phase0-ShiftRegistration	Staff	最高	前提条件	/staff/calendar	シフト登録	1. カレンダーで未来の日付をクリック 2. 時間帯を選択 3. シフト登録	シフトが登録される	E2E	予約作成の前提条件	e2e/flow/flow-001-020-final.spec.ts		SUCCESS						NONE	シフト登録済み			
FLOW-002	2	Phase1-ReservationCreation	User	最高	予約作成	/user/reserve	ステップ1: ペット選択	1. ユーザー選択プルダウンをクリック 2. ユーザーを選択 3. 次へボタンをクリック	ステップ2に遷移	E2E	予約フロー開始	e2e/flow/flow-001-020-final.spec.ts		SUCCESS						NONE	ステップ2			
FLOW-003	3	Phase1-ReservationCreation	User	最高	予約作成	/user/reserve	ステップ2: コース選択	1. 予約コースを選択 2. 次へボタンをクリック	ステップ3に遷移	E2E		e2e/flow/flow-001-020-final.spec.ts		SUCCESS						ステップ2	ステップ3			

テストはCSVで"計画化"（2）10件ループ

FAILは2回まで。以降は手動に切替
実装途中の四苦八苦も記録→PMとして全体が掴める

10件ずつ実行し、プロンプトは毎回リピート
片手間でテスト作成・実行

まとめ：説明責任と未来予想

説明責任がある限り、
ドキュメントは消えない。

明日から使える3つの型：

① ChatGPTにMD仕様テンプレを作らせる

② Copilotへ"最小スコープ指示"で投げる

③ テスト計画CSV→10件ずつテスト自動化ループ

🎯 **ベネフィット**

上流SEムーブ（合意・設計・説明）が身につく