無題

2025年11月9日　20:12

AIを活用しやすいソフトウェアの特徴（本質）

1. 状態やフローが “明文化されている”

AI にとって最も扱いやすいのは「状態遷移とデータ構造が明確なソフトウェア」です。

● AIが扱いやすい例

- 状態管理ドメインがあり、状態変化が明示的
- グローバル変数がない／限定的
- 入力 → 状態変化 → 出力 が明文化されている
- モジュール間の依存が明確で、AI が追いかけやすい

● AIが扱いにくい例

- 不特定多数のモジュールが暗黙的に同じ変数を書き換える（←あなたの組織の課題）
- 状態がコード全体に散在している
- 暗黙仕様・慣習・口頭文化が多い

AI はコードやログを読み取ることはできますが、**暗黙仕様は読み取れません。**
したがって、仕様・状態・データ構造の明文化は非常に重要です。

2. モジュールの独立性が高い（疎結合）

AI による以下が容易になります：

- 不具合箇所の特定
- リファクタリング提案
- テストケース生成
- コード補完・改善提案

良い構造

- 単一責任のモジュール
- API 的なインターフェース
- 関数やメソッドが小さい
- 入力と出力が明確

悪い構造

- モジュールが互いに内部実装を知っている
- “神クラス” が存在する
- 1 つの関数が巨大

AI は巨大で絡まったコードを理解するのが非常に苦手です。

3. データが構造化されている

AI による高度化（予測・分析・自動化・改善提案）をするには、**ソフトにデータが残っていること**が必須です。

例：AIが活用しやすいデータ

- 検査ログ
- センサー値
- 設定変更履歴
- 画像メタ情報
- UI 操作ログ
- タイムスタンプ
  —すべて構造化（CSV/DB）されている状態で保存されている場合

NG例

- ログファイルがフリーテキスト
- 画像・値・結果がバラバラのファイルに散在
- ファイル名にだけ情報が入っている（構造化されていない）

データを機械が扱える形にすることは、AI 活用の前提条件です。

4. テスト構造が整っている

AI がテストケースを生成しやすい構造：

- 入力と期待結果が明確
- 仕様がテキストとして存在
- テスト観点が整理されている

もし仕様がソースに埋め込まれていたり、担当者の頭の中にあったりすると、
AI は正しいテストを作れません。

5. API 化・コマンド化されている

AI ツールは API や CLI に対して非常に強いです。

- ワークフロー自動化
- テスト実行
- シミュレーション
- パラメータチューニング
- 実行ログの収集と分析

「AI から操作できる仕組みがある」ソフトウェアは圧倒的に強いです。

□ 本質：AI を活かせるのは “構造化された情報” を持つソフトウェア

AI の能力を最大化するには、

- **状態**
- **仕様**
- **データ**
- **API/操作手続き**
- **ログ**

これらを**構造化して明文化**することが最重要課題です。
実はこれ、
あなたが現在取り組んでいる「状態管理ドメインの導入」「依存性の整理」と完全に一致しています。

□ では、AI を最大限活用できるソフトウェアとは？（まとめ）

項目	AI 活用しやすい	活用しにくい
状態管理	明確な状態管理ドメイン	グローバル変数が乱立
モジュール構造	疎結合・単一責任	依存が複雑・神クラス
データ	構造化されて保存	フリーテキスト・散在
テスト	仕様ベースの整理されたテスト	テストが属人化
API	外部操作しやすい API	内部密結合
ログ	整形ログ・メタ情報	バラバラなログ

□追記：あなたの組織が AI 活用で「最初に取り組むべきこと」

あなたの状況だと、AI 活用しやすいソフトウェアにするためには：

① 状態管理ドメインの導入 → ソフトの構造化

AI が理解できるソフトにするための最上位の対策。

② ログ・計測値・画像を構造化して記録

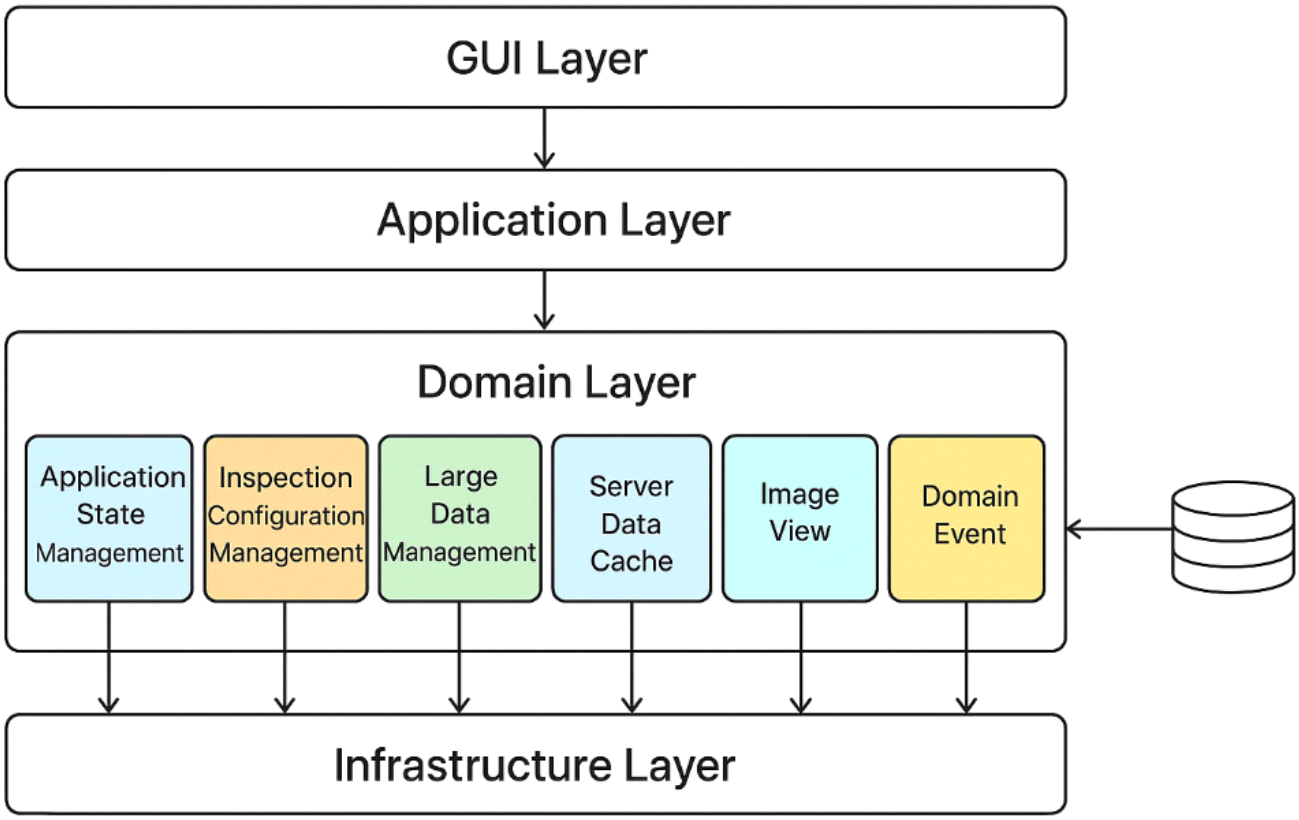
PowerBI を用いた可視化、AI 解析、異常検知などにつながる。

③ 仕様・設計をテキスト化 → AI に読み込ませる

テスト生成・欠陥検知・レビュー自動化が可能になる。

④ モジュールの依存関係を整理（疎結合化）

AI によるリファクタリング支援が可能に。


GUI Layer
Application Layer
Domain Layer
Application State Management
Inspection Configuration Management
Large Data Management
Server Data Cache
Image View
Domain Event
Infrastructure Layer