

2月末完了スプリント計画 (印刷用)

作成日: 2026-02-10

期間: 2026-02-11 ~ 2026-02-28

目的: **Clean Architecture 習得 + Dify 習得を2月末までに最低ライン達成**

1. 完了条件 (2月末時点)

- Python側の構成整理が完了（特に Presentation の旧参照解消）
- Dify 連携を Port/Adapter で1本動作
- 主要フローのテストが緑（pytest）
- 変更した機能ごとに「説明できる」状態

2. 必要時間 (現実的見積)

- 必須タスク合計: **34時間**
- バッファ: **4時間**
- 合計目安: **38時間**

2月末完了ペース（目安）：

- 平日: 2時間 × 14日 = 28時間
- 週末: 5時間 × 2週 = 10時間
- 合計: 38時間

3. 残タスク一覧 (優先順)

T1. Presentation の旧依存解消 (最優先)

- 目安: 6時間
- 対象: `src/presentation/webhook_server.py`
- 内容:
 - `src.base` / `data.schema` 依存を現行構成へ置換
 - 依存注入方針を明確化
 - TODO 部分の最小実装方針を確定
- 完了基準:
 - 旧 import が消えている
 - ルート処理が UseCase 呼び出し中心になっている

T2. Presentation 周辺の空/未整理ファイル対応

- 目安: 3時間
- 対象: `src/presentation/controllers/webhook_controller.py` ,
`src/presentation/dtos/webhook_request_dto.py`
- 内容:
 - 実装するか、廃止するかを決める
 - 決めた方針を `docs` に残す
- 完了基準:
 - 未使用ファイルの扱いが決定済み

T3. Presentation テストの実構成化

- 目安: 5時間
- 対象: `tests/presentation/test_webhook_server.py`
- 内容:
 - モジュール注入前提テストから脱却
 - 現行構成に合わせたテストへ更新
- 完了基準:
 - テストが実構成ベースで通る

T4. Dify 連携の Port 設計

- 目安: 4時間
- 内容:
 - Dify 呼び出し用 Port を定義
 - 入出力を Domain 型で固定
- 完了基準:
 - Port が定義され、UseCase から呼べる

T5. Dify Adapter + Client 雜形実装

- 目安: 6時間
- 内容:
 - Adapter で入出力変換
 - Client で Dify API 通信
- 完了基準:
 - Adapter 経由で Dify API 呼び出し成功

T6. UseCase への Dify 接続 (最小1本)

- 目安: 4時間
- 内容:
 - 既存 UseCase の1本を Dify 経由に接続
- 完了基準:
 - UseCase -> Port -> Adapter -> Client の経路が通る

T7. 異常系テスト追加 (最低3本)

- 目安: 4時間
- 内容:
 - タイムアウト
 - API失敗
 - 不正レスポンス
- 完了基準:
 - 異常系テスト3本以上追加

T8. 学習ドキュメント統合（説明可能化）

- 目安: 2時間
- 内容:
 - 変更点を docs に追記
 - 図とコード参照を一致させる
- 完了基準:
 - 3分説明が可能（自分向け読み上げで確認）

4. 2月末までのスケジュール（日付固定）

Week 1 (2/11～2/14, 10時間)

- 2/11: T1-1 import 整理 (2h)
- 2/12: T1-2 依存注入/ルート整理 (2h)
- 2/13: T2 空ファイル方針決定 + 反映 (2h)
- 2/14: T3-1 Presentation テスト刷新開始 (4h)

Week 2 (2/15～2/21, 12時間)

- 2/15: T3-2 テスト完了 (3h)
- 2/16: T4-1 Dify Port 設計 (2h)
- 2/17: T4-2 Port 実装 + テスト (2h)
- 2/18: T5-1 Dify Client 雛形 (2h)
- 2/19: T5-2 Dify Adapter 雛形 (2h)
- 2/21: T5-3 疎通確認 (1h)

Week 3 (2/22～2/28, 16時間)

- 2/22: T6-1 UseCase 接続 (3h)
- 2/23: T6-2 接続修正 + 回帰確認 (2h)
- 2/24: T7-1 異常系テスト1本 (2h)
- 2/25: T7-2 異常系テスト2本 (2h)
- 2/26: T7-3 異常系テスト3本 (2h)
- 2/27: T8 docs 統合 (2h)
- 2/28: バッファ/仕上げ (3h)

5. 実行ルール（重要）

各タスクで必ず以下を実施する:

- 実装
- 3分説明（声に出して説明）
- テスト1本追加
- docs/worklog 1行追記

説明できない場合:

- コードを戻す前に、責務（UseCase/Domain/Adapter）を再整理
- AIに「どの責務か」だけ質問して修正

6. AI活用ルール（速度と理解の両立）

使ってよい:

- 雛形生成
- テストケース洗い出し
- 命名・責務レビュー

使ってはいけない:

- 理解なしマージ
- テストなし反映
- docs未更新で完了扱い

7. 毎日のチェック欄 (印刷して使用)

日付: _____

今日の対象タスク番号 (T_)

実装完了

3分説明できた

テスト1本追加

`pytest` 実行

`worklog`更新

明日の着手点を1行残した