

## タスク一覧: Slide Voice Maker

入力: [/specs/001-Slide-Voice-Maker/](#)からの設計ドキュメント 前提条件: plan.md (必須)、spec.md (必須)、data-model.md、contracts/ バージョン: 1.0.0 開始日: 2026-01-05

### 形式: [ID] [P?] [ストーリー?] 説明

- [P]: 並列実行可能 (異なるファイル、依存関係なし)
- [US1]: ユーザーストーリー1 (解像度選択)
- [US2]: ユーザーストーリー2 (temp上書き更新)

### 実装スケジュール



注意: 2026-01-05 (月曜) を開始日として設定し、土日・年末年始 (12/27-1/4) を除外しています。実際のスケジュールは稼働日にに基づいて自動調整されます。開始日を変更する場合は、最初のタスクの開始日（[2026-01-05](#)）を任意の日付に変更し、他は [after](#) で相対的にスケジューリングされます。

- [P]: 並列実行可能 (異なるファイル、依存関係なし)
- [US1]: ユーザーストーリー1 (解像度選択)
- [US2]: ユーザーストーリー2 (temp上書き更新)

### Phase 1: セットアップ

目的: プロジェクト構造確認と仕様ドキュメント作成

- T001 specs/001-Slide-Voice-Maker/フォルダを作成
- T002 [P] spec.md (機能仕様書) を作成
- T003 [P] plan.md (実装計画) を作成

チェックポイント: ドキュメント準備完了 ✓

### Phase 2: 基盤 (ブロッキング前提条件)

目的: 解像度選択・temp管理の共通インフラ構築

⚠️ 重要: このフェーズが完了するまでユーザーストーリー作業は開始不可

- T004 src/main.pyにRESOLUTION\_MAP定数を定義 (720p/1080p/1440p→幅ピクセル)
- T005 [P] src/main.pyに--resolution引数をargparseに追加
- T006 [P] 環境変数OUTPUT\_MAX\_WIDTHへの変換処理を実装

チェックポイント: 基盤準備完了 - ユーザーストーリー実装を開始可能 ✓

### Phase 3: ユーザーストーリー1 - 解像度選択 (優先度: P1) ⌚ MVP

目標: ユーザーが動画生成前に出力解像度 (720p/1080p/1440p) を選択可能にする

独立テスト: `py src/main.py --resolution 1080p` を実行し、出力動画の解像度が1920x1080であることを FFprobeで確認

#### ユーザーストーリー1の実装

- T007 [US1] src/main.pyで--resolution引数をパースしRESOLUTION\_MAPから幅を取得
- T008 [US1] src/main.pyで取得した幅をos.environ["OUTPUT\_MAX\_WIDTH"]に設定
- T009 [US1] src/processor.pyの\_get\_output\_max\_width()が環境変数を正しく読み取ることを確認
- T010 [US1] 無効な解像度値の場合は720p (デフォルト) にフォールバックするバリデーション追加

チェックポイント: 解像度選択機能が独立して動作 ✓

### Phase 4: ユーザーストーリー2 - temp上書き更新 (優先度: P1)

目標: 毎回の実行時にtempフォルダを自動クリアし、古いファイルを残さない

独立テスト: 2回連続で動画生成を実行し、output/temp内に1回目のファイルが残っていないことを確認

#### ユーザーストーリー2の実装

- T011 [US2] src/processor.pyにclear\_temp\_folder(temp\_dir)関数を追加
- T012 [US2] clear\_temp\_folder()内でshutil.rmtree()とos.makedirs()を使用
- T013 [US2] process\_pdf\_and\_script()の冒頭でclear\_temp\_folder()を呼び出し
- T014 [P] [US2] PermissionError時のエラーハンドリングとログ出力追加

チェックポイント: temp上書き機能が独立して動作 ✓

### Phase 5: Web UI (優先度: P1)

目標: サーバー (src/server.py) と連携するWeb UI

独立テスト: index.htmlでPDF/CSV入力→音声生成→WebM/MP4ダウンロードが可能であることを確認

#### Web UIの実装

- T015 index.htmlにRESOLUTION\_OPTIONS配列を定義 (label, value, width, height)
- T016 index.htmlにサーバー連携機能を実装 (PDF/CSVアップロード、動画生成、ダウンロード)
- T017 src/server.pyにFastAPIエンドポイントを実装 (PDF/CSVアップロード、動画生成、ファイル一覧、ダウンロード)
- T018 CSV文字化け対処をTextDecoderベースに強化 (UTF-8/Shift\_JIS等 + RFC4180最小対応)

チェックポイント: Web UIが独立して動作 ✓

### Phase 6: 仕上げとテスト

目的: E2Eテスト実行、ドキュメント更新、最終検証

- T019 [P] tests/e2e/test\_resolution.pyでCLI E2Eテスト (解像度・非空WebM/MP4確認)
- T020 [P] tests/e2e/test\_local\_backend.pyでバックエンドE2Eテスト
- T021 [P] README.mdを要件/テスト/実行手順に整合
- T022 [P] docs/完全仕様書.mdを現行仕様に整合
- T023 [P] specs/001-Slide-Voice-Maker/{spec,plan,quickstart}.mdを整合 (リンクはGitHub URLへ)
- T024 E2Eを実行し100%成功を確認

チェックポイント: 全機能テスト・ドキュメント完了

### Phase 8: 男声/女声 (話者) 選択

目的: 解像度選択の右に男声/女声ブルダウンを追加し、画像・音声生成で選択した話者を反映する。

- T025 index.htmlに男声/女声ブルダウン (ツールチップ付き) を追加
- T026 src/server.py / src/processor.pyにvoice\_gender受理とEdge TTS voice反映を追加
- T027 E2Eテストにvoice\_gender受理の回帰テストを追加

チェックポイント: UI選択が音声生成に反映される ✓

### Phase 9: GitHub Pages

目的: `index.html` 等の静的成果物をGitHub Pagesへデプロイ可能にする。

- T028 .github/workflows/pages.yml を整備 (dist生成→Pagesへデプロイ)

チェックポイント: ActionsからPagesデプロイできる ✓

### Phase 10: 運用 (手動作業)

重要: 下記はローカル/リモートのGit認証・運用判断が必要なため、自動実行ではなく手動確認を必須とする。

- T029 「pdf以外の不要ファイル」の定義を合意し、削除対象を確定 (生成物のみ等)
- T030 ローカルとリモートの全ブランチをmainへマージし削除 (認証が必要)

### 依存関係と実行順序

#### フェーズ依存関係



#### ユーザーストーリー依存関係

- ユーザーストーリー1 (P1) : 基盤 (Phase 2) 完了後に開始可能 - 他のストーリーへの依存なし
- ユーザーストーリー2 (P1) : 基盤 (Phase 2) 完了後に開始可能 - US1とは独立

#### 並列実行可能タスク

| Phase   | 並列実行可能タスク  |
|---------|------------|
| Phase 1 | T002, T003 |
| Phase 2 | T005, T006 |
| Phase 5 | T015-T018  |
| Phase 6 | T019-T023  |

### 並列例: 基盤完了後

```
# 基盤完了後、2つのユーザーストーリーを並列開始可能:  
チームA: "ユーザーストーリー1 - 解像度選択"  
チームB: "ユーザーストーリー2 - temp上書き"
```

### 実装戦略

#### MVP優先 (ユーザーストーリー1のみ)

- Phase 1: セットアップを完了 ✓
- Phase 2: 基盤を完了 ✓
- Phase 3: ユーザーストーリー1を完了 ✓
- 停止して検証: 解像度選択機能を独立して動作 ✓
- 準備ができたらデプロイ/デモ

#### インクリメンタルデリバリー

- セットアップ + 基盤を完了 → 基盤準備完了 ✓
- ユーザーストーリー1を追加 → 独立してテスト → デプロイ/デモ (MVP!) ✓
- ユーザーストーリー2を追加 → 独立してテスト → デプロイ/デモ ✓
- Web UIを追加 → 独立してテスト → デプロイ/デモ ✓
- 各ストーリーは前のストーリーを壊さずに価値を追加

### タスク進捗サマリー

| 項目    | 数値 |
|-------|----|
| 総タスク数 | 30 |
| 完了    | 28 |
| 未着手   | 2  |

### 注意事項

- Python 3.10.11を使用 ([py -3.10](#))
- UTF-8エンコーディング必須
- 土日・年末年始 (12/27-1/4) はスケジュール対象外
- 各チェックポイントで動作確認を実施
- [P] タスク = 異なるファイル、依存関係なし
- [US\*] ラベルはトレーサビリティのためタスクを特定のユーザーストーリーにマップ

### 完了条件

- すべてのタスクが完了状態になっていること
- CLI E2Eテスト (T019) が成功すること
- バックエンドE2Eテスト (T020) が成功すること
- ドキュメント整合 (T021-T024) が完了すること