

VS仕様説明

2025年9月1日 21:33

VSの2大仕様(XG-Xとの違い)

フラットUI

VSはフラットUIを採用。コンテンツの視認性向上とユーザー操作の手数削減を実現。必要な情報へスムーズにアクセスできる画面設計となっている。

オンライン編集

VSはオンライン編集が可能。実機上で直接設定を編集できる。これにより、動作確認や設定反映が短時間で完了して運用効率性を高めている。

VJのPCアプリで、フラットUI・オンライン編集を実現したが、VSにはVJになかった以下の設計難易度を高める特性があった。

1. データ量の多さ:

ツール数最大2000点、最大検出個数3万点、データセット2000点、シミュレーション画像1000枚以上、などデータ量が多い仕様が多数。それらをフラットUIでは同時に更新する必要があるため、非同期処理が必須となり状態管理が複雑化している。

2. 機能間のデータ連携:

ビジョンダッシュボード（行・列の挿入）、参照関係（入力画像、エラー）、イメージストリップ、
権限、データリスト

- ▶ 機能間の依存関係が強くなり、モジュール間の独立性が低下する。
- ▶ 変更の影響範囲が広がり、横断的な対応が必要になる。

例： イメスト更新・データリスト・検査設定切替

3. リアルタイム性への要求:

- ▶ 高い応答速度が求められ、パフォーマンス最適化が必須となる。
- ▶ 並行処理やキャッシュ戦略など、複雑な実装が必要になる。

4. インライン稼働する実機との連携:

工場で稼働中のインラインの実機とのリアルタイムな連携が求められる。

例えば、コマンド等で実機で検査設定切替やリセットが実行されたときに、VSアプリもそれに合わせて迅速に状態を切り替える必要がある。実機とVSアプリでは状態遷移のずれが生じるため、状態遷移が複雑化する。（設定切替中に次の設定切替要求が

くることもある)

7. 多様な操作方法と表現方法:

- ▶ ショートカット、複数の入力方法、異なる表現方法をサポートする必要がある。
- ▶ 共通化された処理と拡張可能な設計が必要になり、設計要素が増加する。

8. 異種技術間の連携:

- ▶ WebアプリケーションとC#アプリケーションなど、異なる技術スタック間の連携が必要になる。
- ▶ 通信プロトコル、データ形式、など、技術境界を超えるための設計が複雑になる。
- ▶ 複数の外部システムとの連携が必要になり、API連携やデータ形式の変換が複雑になる。
- ▶ システム間の障害対応やバージョン管理が煩雑になる。