

システムオペレーション自動化

(新規自動化の開発プロジェクト)

プロジェクト。

期間: 2025.07.01 ~ 2025.12.26

現場: P社

所属: 自動化チーム (5人/サブリーダー)

概要。

- 既存部門の定型的な手作業業務を自動化。
- 契約・売上・債権の3チーム向けに、各入力ファイルの確認によるデータ照会・検証、結果レポート作成といった手作業プロセスを、Spring Bootベースのバッチプログラムで自動化。
- 運用効率を最大化し、運用コストなどの削減を期待。

技術およびツール。

Java / Spring Boot / Junit5 / Oracle / MyBatis

S3 / SQS / KEDA / EKS

GitHub / IntelliJ / SQL Developer / Copilot / Confluence

担当。

・ ダンプリスト抽出の自動化

ダンプされた売上データが対象データであるかを検証した上で、売上明細を照会し、CSVファイルとして生成・当該部門へ連携。

- 既存業務プロセスの把握（自動化前の作業内容）
- 仕様書作成：自動化仕様書（実現範囲・可否）、ファイル仕様書。
- 実装：ファイル入出力、検証ロジック、データ照会などのビジネスロジックを実装。
- UT：JUnit5でテストコードを作成し、JaCoCoを用いてコードの安全性を向上。
- IT：テスト仕様書作成およびテストデータ準備。
- ST：本番環境と同等の環境で検証
- リリース

・ 債権流動化データパッチ

入力ファイルの特定項目のパターンをチェックし、パターンが違うと、データベースを照会して

[Developer: 魚洋縁\(Yangrok Auh\)](#)

検証をする。異常データが確認出来たら正常データにパッチ処理を行ってCSV出力。

- 既存業務プロセスの把握（自動化前の作業内容）
- 仕様書作成：自動化仕様書（実現範囲・可否）、ファイル仕様書。
- 実装: ファイル入力、チャンク処理およびマルチパートアップロード、データ抽出の自動化を実装。
2千万件超える入力データを一括処理するため、チャンク処理ロジックを実装。
処理したデータを1つの結果ファイルとしてアップロードするためにMultiPart Uploadを採用。
- UT：JUnit5でテストコードを作成し、JaCoCoを用いてコードの安全性を向上。
- IT：テスト仕様書作成およびテストデータ準備。
- ST：本番環境と同等の環境で検証
- リリース

• 未収入金発生チェック

引き落としの取消が発生した顧客の当月未収金額および契約情報を抽出し、CSVファイル出力。

- 既存業務プロセスの把握（自動化前の作業内容）
- 仕様書作成：自動化仕様書（実現範囲・可否）、ファイル仕様書。
- 実装：ファイル入出力、検証ロジック、データ照会などのビジネスロジックを実装
- UT：JUnit5でテストコードを作成し、JaCoCoを用いてコードの安全性を向上。
- IT：テスト仕様書作成およびテストデータ準備。
- ST：本番環境と同等の環境で検証
- リリース

成果。

- 10人体制で対応していたプロセスを自動化ロジックの導入により、2人体制での運用に転換。
- 業務効率を約80%向上させ、運用コストを大幅に削減。
- データ処理の自動化により、検証時間を90%以上短縮し、エラー発生率0%を維持。
- DDD(Domain-Driven Design)開発を通じてビジネスロジックを明確化し、可読性および将来的な拡張性を向上。
- ファクトリーメソッドの適用と単一責任原則の遵守により、オブジェクト生成の一貫性を確保し、カプセル化を強化。コードの品質、拡張性、再利用性、保守性を向上。

開発アーキテクチャ

[Presentation Layer]

- └ exception(自動化工ラ-定義)
- └ runner(実行クラス/ サービスを呼び出す)

[Application Layer]

- └ exception(サービスエラー定義)
- └ service(ドメインを呼び出す)
 - └ chunk
 - └ download
 - └ contract
 - ⋮

[Domain Layer]

- └ model(ドメインクラス/ ビジネスロジック/ dto)
 - └ card
 - └ contract
 - └ CSV
 - ⋮
- └ repository(DB, S3処理インターフェース)
 - └ contract
 - └ CSV
 - └ sales
 - ⋮
 - └ type(中間産出物クラス)

[Infrastructure Layer]

- └ database
 - └ entity (mybatis)
 - └ impl (ドメインレポジトリ実装)
 - └ mapper (mybatis)
- └ s3
 - └ config (s3 client)
 - └ exception (s3処理エラー定義)
 - └ impl (ドメインレポジトリ実装)
 - └ shared (s3 ハンドリングロジック)

[resource]

- └ mybatis(クエリー)
- └ app.properties
- └ log.xml

自動化システム環境

