

基幹システムの保守

(P社システム)

プロジェクト。

期間: 2025.01.06 ~ 2025.06.29

現場: P社

所属: 共通チーム(3人/チーム員)

概要。

- P社の基幹システムは、中核業務を処理するシステムであり、本システムの保守・性能改善・障害対応・新機能追加を担当。
- 運用中に発生するシステム障害、パフォーマンス低下、データ連携の不具合などを改善し、システムの信頼性および処理効率を向上。

技術スタック及びツール。

Java / Interfarm / Junit5 / Oracle / Shell Script

GitLab / IntelliJ / SQL Developer / A5 / Sonar Qube

Confluence

担当。

- **ファイルのエンコーディングの時、発生する文字化け改善**
SJIS ↔ UTF-8 の変換処理をUTF-8に統一し、UTF-8で非対応の文字についてはコード表を作成して対応。 --Shell Script
- **新規画面を追加。**
他システムの画面13件を再利用する際に発生する影響範囲とロジックの修正箇所を調査。
他システムの13画面を連携し、業務ロジックの改修によって機能を拡張。 --Java(API)
- **ファイル連携構造を改善し、不要機能を廃止。**
対象ファイルの送受信を停止する際、既存処理への影響範囲および反映可能性を調査。既存ロジックから当該ファイルの送受信処理部分のみを削除。 --Shell /Java
- **金融明細バッチのメモリを最適化。**
メモリ不足発生の原因を特定するため、処理ロジックを確認・調査。
受信した数十万件のデータごとに生成していたインスタンスを再利用構造に変更。 --Java

• 契約情報返却エラーを修正。

契約情報データが返却されない顧客データをもとに、設計書と処理コードを確認。
分岐処理部分で不具合が発生していることを特定し、分岐条件を追加することで問題を解決。 --Java

• DBキャッシュロック問題を解決。

異なるパーティションで削除処理を実行中に、データ投入処理が同時に発生していることを確認。
キャッシュロックが発生しないよう、削除処理ロジックを変更(truncate → delete) –Oracle(Procedure)

• CSVファイルのフォーマット検証機能を追加。

13種類のファイルごとに既存のフォーマットチェック処理コードを調査・把握。
アップロード時、項目ごとの桁数・ダブルクォーテーションの有無などを検証ロジックを追加。
ファイル別項目の必須入力有無やデータ存在に応じたフォーマットチェック機能を追加。
どの項目にどのような問題があるかをメッセージとして出力する処理を実装。 --Java(API)

成果。

• ファイルのエンコーディングの時、発生する文字化け改善

SJIS ↔ UTF-8 の変換処理を最適化し、文字化けを100%解消。

• 新規画面を追加。

新規開発と比べて約60%以上実装期間短縮およびコスト削減。

• ファイル連携構造を改善し、不要機能を廃止。

使用されてない送受信プロセス削除で保守および運用効率を高める。

• 金融明細バッチのメモリを最適化。

インスタンス再利用構造へ改善し、メモリ使用量を90%以上削減、バッチ処理の安定化を実現。
従来のメモリ使用量は195,000KBから0.195KBへ大幅に削減。

• 契約情報返却エラーを修正。

ロジック分岐の不具合を修正し、欠落問題を解消し、正常データ返却率100%を達成。

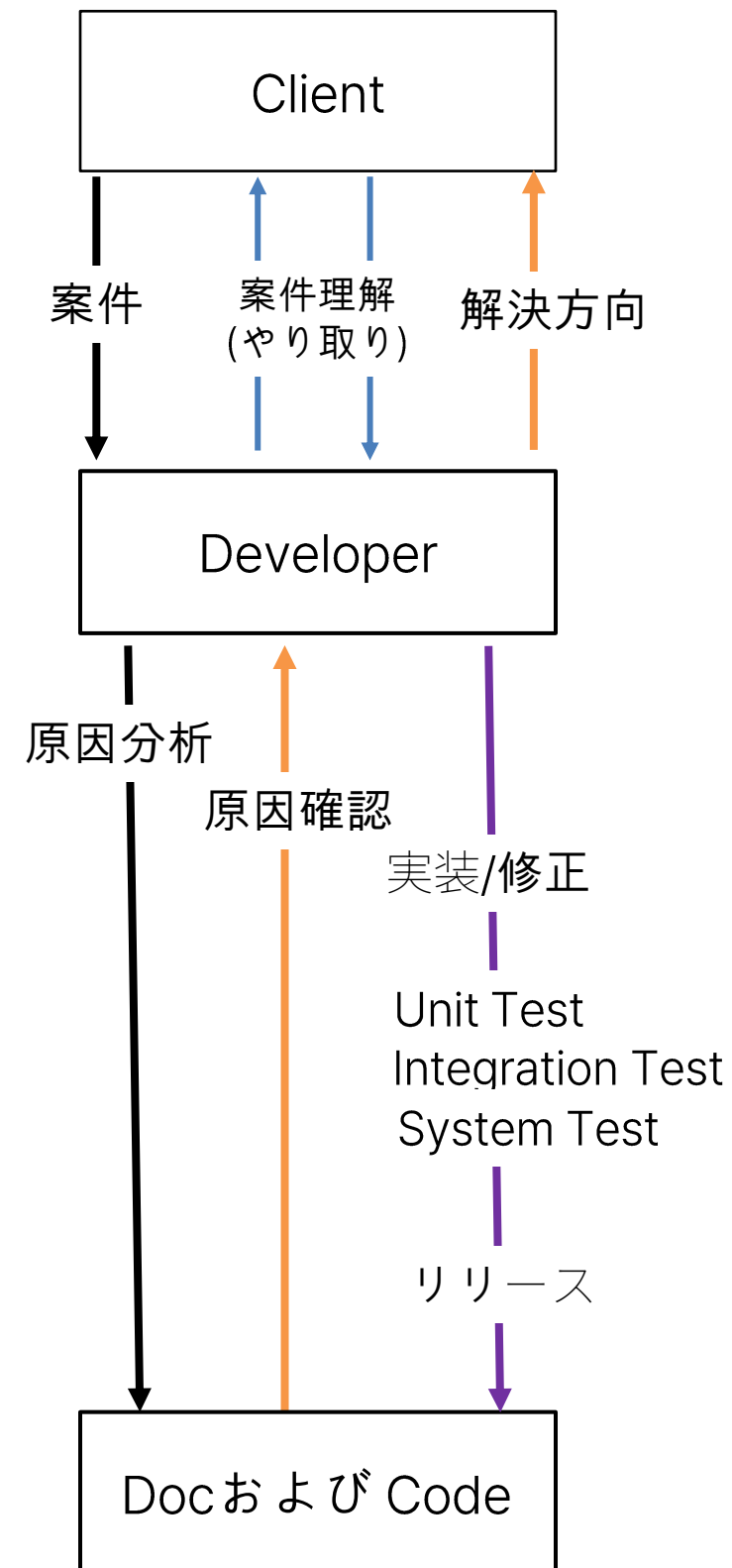
• CSVファイルのフォーマット検証機能を追加。

アップロード段階でフォーマットおよび桁数を自動検証することで、事前のエラーを防止し、
後続処理の工数を削減。

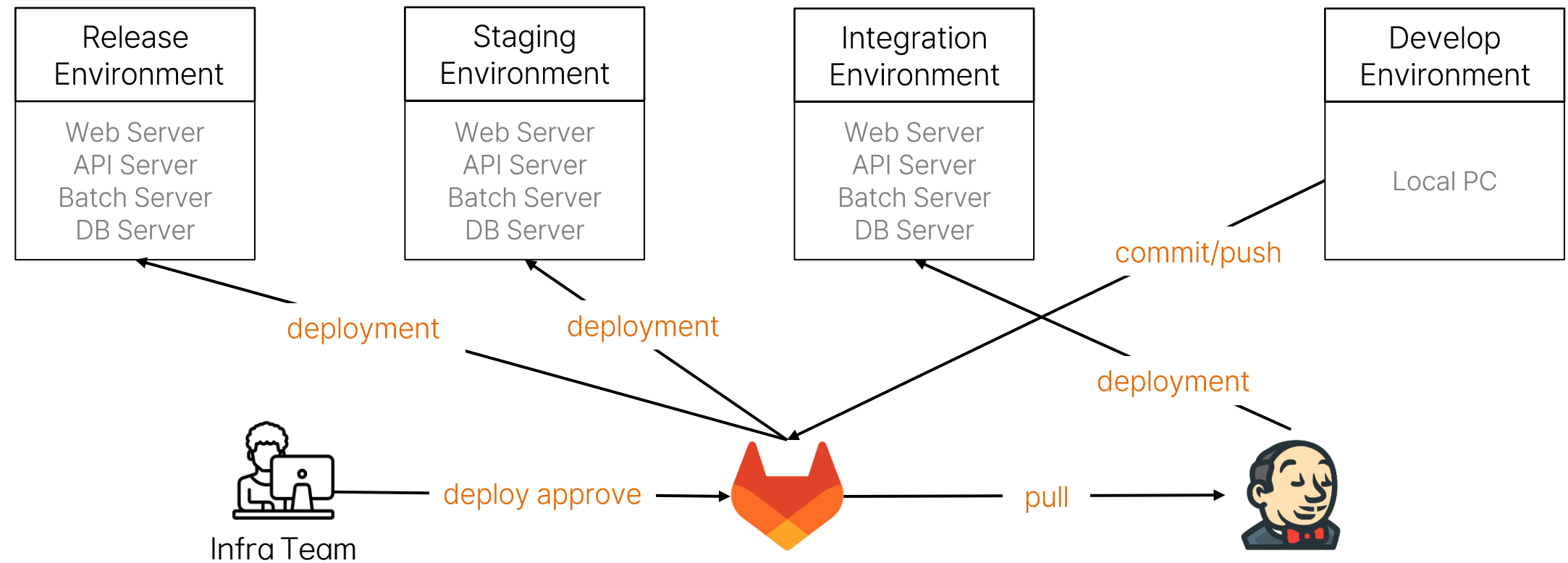
• DBキャッシュロック問題を解決。

パーティションへの同時アクセス時に発生していたキャッシュロック問題を解消し、データ処理の安定性を向上。

案件処理過程



システム環境



環境構成

