3.5 サービス視点での変更管理に関する教訓（T5）

教訓 T5

サービスの視点で、「変更管理」の仕組み作りと「品質管理責任」の明確化を！

問題

サービス系システムの中には、本店ホスト、事業所サーバ、営業端末などの機器を接続したシステムが多い。

A社の使用料請求システムは、本店ホスト／サーバから請求データを営業員が持ち歩くHT（ハンディ・ターミナル）に転送するシステムである（図3.5－1）。

そのシステムから出される請求書の金額が誤ってお客様に渡ってしまい、A社は個別謝罪・請求書の再発行に追われることになった。

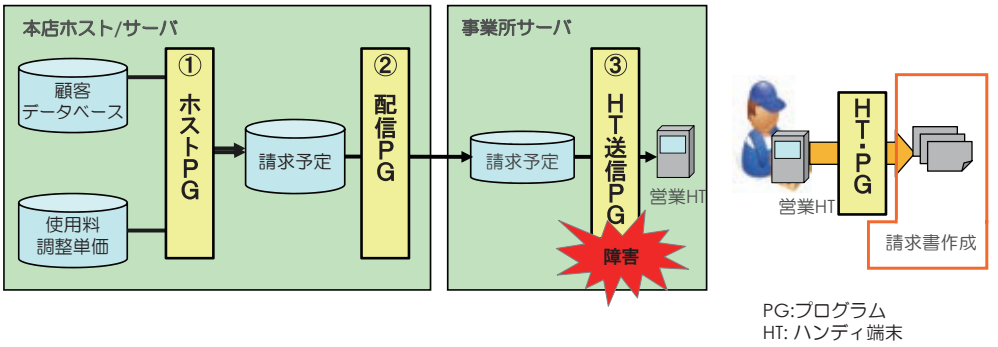

図3.5－1　使用料請求システム

原因

直接の原因は、事業所サーバの③「HT送信プログラム」において、使用料請求データの符号判定処理で、調整額を減算すべきところを加算してしまったことによる。本店ホスト／サーバ上では、正しく減算ができていた。

さらに、障害の原因を分析したところ、以下の2点が判明した。

【原因1】「使用料請求データの符号判定処理で、調整額を減算する」機能は、初期は本店ホスト／サーバ上の①、②では考慮されていたが、当時の状況では、減算となる状況が発生していなかったため、減算を考慮したテストが不十分であった。今回、初めて調整額の減算が発生した。

【原因 2】HT はシステム稼働後、数年が経過した後に新規要件として追加された機能であった。また、インフラ機能であったため、アプリケーション開発部門は参加せず、システム部門だけで更改作業をおこなった。そのため、機能変更に対する影響の認識が甘かった。

この事例のようなサービス系システムのライフサイクルは長い。その間に、途中で新たな要件が追加されたり【原因 2】、使用方法が変わったり【原因 1】する。そのために今まで正常に稼働していたシステムに、突如障害が発生する場合がある。

それを防ぐには、常にエンドユーザへのサービスの視点で「何が変わったのか」をチェックすることである。この事例では、障害を未然に防ぐポイントが 2 カ所あった。先ずは、【原因 2】の HT を導入することによりサービスが変更になったことと、次に、【原因 1】の今まで調整額は、プラスしか発生していなかったが、今回初めてマイナスが発生し、やはりサービスが変更されたことである。つまり、根本原因としては、システムをサービスの視点で見渡した変更管理ができていなかったことである。サービスの視点で見渡した変更管理ができていれば、障害を未然に防ぐことができたと考える。

さらに今回の事例で、「本店ホスト／サーバ→事業所サーバ→ HT」と経由していく途中で、以下の問題が生ずる可能性が考えられる。

① データ整合が取れない可能性がある。
② プログラム仕様、実装、それぞれ整合が取れない可能性がある。
③ テスト仕様・実施整合が取れない可能性がある。

対策

このシステム障害を抑えるためには、サービスの視点での変更点を見落とさない仕組みを作る。つまり、変更があったときに、システム全体のプログラムの整合性、データの整合性、テスト仕様の整合性を保つための変更管理を以下の手順で行う（図 3.5－2）。

① 機能要件は当初からあったとしても、長い間に変わっている場合がある。エンドユーザにとって、初めて使用する機能であれば、新規の要件とみなして対応する。
② 現状分析をおこない現行システムの仕様を確認し、機能変更に対する「要件定義書・設計書・テスト仕様書、プログラム」の影響調査を実施する。
③ 上記①、②の変更を含めたシステム全体のテストを実施する。
④ 修正漏れがないように、「要件定義書・設計書・テスト仕様書、プログラム」の整合性を管理する。
⑤ システム全体のデータ整合性、プログラム整合性、テスト仕様整合性を確認する人を決め、品質管理責任を明確にし、開発フェーズごとの検証（レビュー等）を行う。

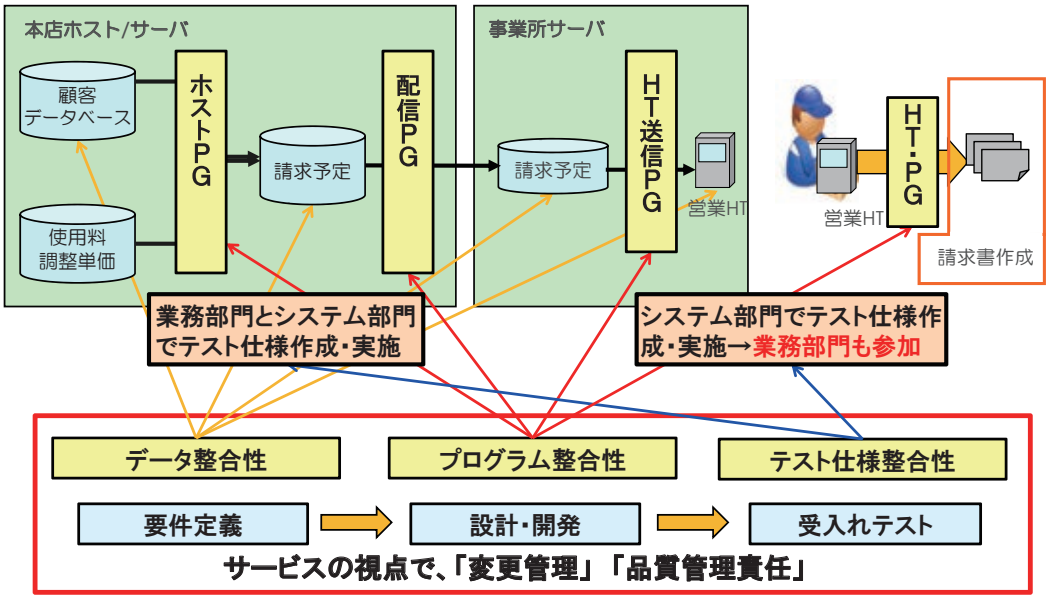

図 3.5－2　システムの概要と障害への対策

効果

変更管理の仕組み作りと開発工程ごとの確認体制を作ることにより、システムの品質を保つことができる。

教訓

サービスの視点で、「変更管理」の仕組み作りと「品質管理責任」の明確化を行うことが重要である。サービス開始時に入れた機能でも、その機能を使用していない状態が長く続くと、いざ使うときに、思わぬ障害を引き起こすことに注意すべきである。