

2.3 上流工程での運用部門の関与に関する教訓 (G3)

教訓
G3

運用部門は上流工程 (企画・要件定義) から 開発部門と連携して進めるべし

問題

最近、システム運用の現場で以下のような問題が多発している。

- ① A 社では、新システムの運用を開始してからオペレータの操作ミスが多発した。
- ② B 社では、販売店向け発注システムを Web システムに移行したところ、システム移行直後にシステム間の電文データに不具合が発生し、発注処理が行われなくなる事象が発生した。入力データのミスに起因し、電文データが誤って編集されたためである。接続会社間で障害対応時の各種調整に手間取り、最終的なシステム復旧は 1 週間程度かかった。

原因

直接の原因は以下である。

- ① A 社では企画や要件定義段階において、オペレータ操作に関する運用の要件検討が十分されていなかった。運用テストの段階から初めて運用者が参加してテスト及び引き継ぎを行ったがオペレータ操作関連のバグが多発して収束しないまま本稼働を開始した。要件定義段階でのオペレーション要件の検討漏れが原因だった。
- ② B 社ではシステムへのデータ入力ミスを抑止する工夫や仕組みが考慮されていなかった。また、本システムの接続先との間で、有事に関する取り決めや対応範囲などが整理できておらず、コンティンジェンシプランが共有できていなかった。

上記①、②の根本原因は、運用要件の検討漏れあるいは軽視である。これは、運用者が要件定義作業へ参加していないことや、参加していても運用要件が取り込まれなかったことに起因している。

対策

上記の対策として、企画・要件定義作業において、運用者の視点からシステム要件を確認するようにする。運用者が確認する項目の例を表 2.3-1 に示す。なお、これは一例であり、これを参考にして、対象となるシステムに合わせて役割分担表などに表現する。プロジェクト開始前に関係者でレビューして合意し、プロジェクトに適用する。

2

ガバナンス／マネジメント領域の教訓

2.3 上流工程での運用部門の関与に関する教訓 (G3)

表 2.3-1 「運用者が企画・要件定義工程で確認する項目」(1/2)

No	工程	分類	項目	全体で確認する項目	運用部門の関与度合い ◎:責任 ○:担当 △:支援	運用部門が確認する項目
1	企画	起案	■経営戦略を見据えたシステム構築及びシステム構築の目的・目標の明確化	①経営戦略の具現化 ②情報(システム)戦略の具現化 ③システム化の目的・方針 ④納期(スケジュール) ⑤システム利用期間、ライフサイクル概要計画	△	①要件の把握 ②納期(スケジュール)の確認 ③システムのライフサイクル(更新間隔)確認 ④経営戦略の理解 ⑤情報(システム)戦略の理解
2		現状分析	■システム(業務)の現状分析、問題点・課題の抽出と分析	①運用状況 ②問題点・課題 ③最新のシステム動向、技術動向の分析	△	①構築ノウハウの提供 ②現状課題の提供 ③最新技術動向・トレンド等の情報分析提供 ④運用状況の報告
2-2		企画立案	■新システムの企画立案	①全社目標の体系化と施策の定義 ②情報システム要件のまとめ ③システム化の企画立案	○	①運用改善から見た新システム要件の提案
3		投資対効果	■システム構築における投資対効果の明確化	①投資対効果 ②コスト計画の立案	○	①運用要件者側に必要なコストの見積作成
4	承認	■システム構築の承認	①システム構築の承認を得る			
5	要件定義	経営、業務、システム要件の確認	■システム化要件の確認 ■サービス内容・レベルの確認 ■体制の確認 ■計画・スケジュールの確認 ■コストの確認 ■移行要件の確認	①システム化の背景、意図確認 ②現行業務調査と現行システム調査より現状の業務フロー概要、システムフロー概要把握 ③システム化の範囲確認 ④新業務フロー、新システムフロー確認 ⑤システム化要件・機能確認 ⑥問題点・課題確認 ⑦サービス内容、サービスレベル確認 ⑧開発計画・スケジュール確認 ⑨開発(プロジェクト)管理方法確認 ⑩開発/運用/保守体制と規模確認 ⑪概算見積もりを算出し、拘束期間、予算、コスト確認 ⑫企画段階で確立した開発者・運用者の双方の要件を確認し運用要件として確定 ⑬移行時の制限事項などを確認	○	① RFP の理解 ②システム化の狙い・概要・範囲の理解 ③マスタスケジュールの認識 ④サービスレベルの認識 ⑤コストの認識 ⑥ライセンスや守秘義務などの規約や条件の認識
6			■システム構成の確認/提案 ■サーバのハード構成・環境要件の確認/提案 ■性能・拡張性要件の確認/提案	①機能要件を把握し、システム構成が運用に耐えられるか判断 ②システムを構成する機器の性能・容量は処理ピークに耐え拡張性も備えている事 ③システム予算と照らし合わせた最適な構成	○	①システムのハード構成の確認 ・ピーク性能をカバーしているか ・拡張性はどうか ・信頼性はどうか ・サービスレベルを満たしているか ・保守性はどうか ・標準化に準拠しているか ・ルールに準拠しているか ・コストは範囲内か ②システムのハード構成の提案
7			■キャパシティ管理要件の確認	①取得する項目 ②各リソースのしきい値 ③ピーク日のデータ処理量 ④トランザクション応答時間のしきい値 ⑤処理データの増加率を定期的に取得 ⑥パッチジョブの所要時間のしきい値 ⑦パッチジョブの終了時刻の許容範囲、異常を検出できる仕組み ⑧1ジョブで使用する最大テープユニット台数 ⑨管理手順の整備と資源枯渇防止対策	○	①必要データ量の確認 ②必要トラフィック量の確認 ③必要レスポンス値の確認 ④必要キャパシティの把握(ピーク、平常) ⑤現状のキャパシティ状況の把握 ⑥しきい値の確認 ⑦将来値の予測 ⑧制限項目の把握 ⑨ルール・標準化に準拠しているか ⑩システム構成がキャパシティ的に適正かどうかの検討と判断 ⑪システム構成の提案
8			■システム構成の確認 ■サーバのソフトウェア要件(運用系)の確認	①OSはルールや標準化に準拠している運用可能なOSである事 ②待機システムがある場合は待機システムにも各種ソフトウェアを導入する ③各種運用ツールは既存標準ツールに合わせる事	○	①システムのソフト構成の確認 ・OSは運用可能か、 ・運用系ソフト、既存標準ソフト ②システムのソフト構成の提案
9	非機能要件システム基盤	■システム構成の確認 ■パソコン・周辺機の要件の確認	①パソコンの構成 ②プリンタ ③その他周辺機 ④稼働可能OS、ソフトウェア、開発資産、サービスパック、累積パッチのレベル ⑤ID、パスワードの設定、ログイン権限設定 ⑥資産配布	○	①クライアントのソフト構成の確認 ・OSは運用可能か、 ・運用系ソフト、既存標準ソフト ②クライアントのソフト構成の提案 ・ソフト構成を確認しNGの場合は提案する	
10		■信頼性要件の確認(冗長化・二重化など)	①最重要システム業務再開 ②サーバやネットワーク機器の冗長化/分散化構成 ③コールドスタンバイ、ホットスタンバイ構成 ④システムの拠点切替時の他システムの影響 ⑤拠点切替時、連携システムなどでの切替え操作 ⑥単独部品故障では機能停止に陥らない冗長化構成、ホットスベア実装、部品活性化交換 ⑦24時間365日のサービス提供の場合は冗長化構成とし、保守作業等は、待機系に切り替えて作業を行う。 ⑧クラスタ切替え手順自動化、手順書作成	○	①信頼性対応方針の確認 ②信頼性対応システムの確認 ③信頼性対応範囲の確認 ④信頼性対応方法の確認 ⑤対応時間の確認 ⑥信頼性構成の確認 ⑦ルール・標準化に準拠しているか ⑧信頼性構成の提案	

表 2.3-1 「運用者が企画・要件定義工程で確認する項目」(2/2)

No	工程	分類	項目	全体で確認する項目	運用部門の関与度合い ◎:責任 ○:担当 △:支援	運用部門が確認する項目	
11	要件定義	非機能要件システム基盤	■セキュリティ要件の確認	①セキュリティ方針及び規定に対し、運用部門の観点で問題のないことの確認 ②アクセス権限の設定 ③ID管理 ④個人情報ファイルの取扱い ⑤センターポリシーの通知 / セキュリティパッチ適用 ⑥ウイルス対策	○	①セキュリティ対応方針の確認 ②セキュリティ対応システムの確認 ③セキュリティ対応範囲の確認 ④セキュリティ対応方法の確認 ⑤セキュリティ構成の確認 ⑥ルール・標準化に準拠しているか ⑦セキュリティ構成の提案	
12				■システム構成の確認 ■ネットワークの要件の確認	①システム全体のネットワーク構成 ②既存増強(伝送容量アップ)、新規増設(拠点追加)、帯域制御 ③ネットワーク機器、メーカ ④設置拠点の大小によるネットワーク構成と機器 ⑤IPアドレス、コンピュータ名 ⑥冗長化のルール ⑦監視方法(死活監視など) ⑧ファームウェアの更新 ⑨事故停電や計画停電の対応 ⑩帯域制御 ⑪F/W	○	①ネットワーク構成の確認 ・トラフィック量は構成に耐えるのか ・帯域制御の確認 ・冗長化の確認 ・監視方法の確認 ・保守方法の確認 ・停電対策の確認 ・ルール・標準化に準拠しているか ②ネットワーク構成の提案
13				■施設要件	①機器の設置場所 ②設置場所の耐震、電源、空調、セキュリティの仕様 ③設置機器の電源仕様 ④設置機器の物理仕様 ⑤設置機器のネットワーク使用 ⑥機器の設置期間	○	①設置場所は適正か確認 ②耐震仕様の確認 ③電源仕様の確認 ④空調仕様の確認 ⑤入室ルール規則の確認 ⑥施設要件の提案
14		運用・保守	■オペレーション要件	①稼働日及び、業務運用時間 ②ジョブ実行マニュアルの整備 ③コンソール要求とマニュアルの整備 ④入力運用 ⑤出力運用 ⑥バックアップの復旧要件 ⑦変更管理 / リリース / 資産管理運用 ⑧システムの監視範囲 ⑨障害検知及び通知のルール ⑩運用保全用ドキュメントと引き継ぎ期間 ⑪ログの表記、保存期間、削除タイミング	○	システム運用方針の確認 ①新規システムの業務運行時間の確認 ②類似するシステムの保守内容の情報提供 ③保守契約の妥当性の確認 ④類似するシステムの出入口・バックアップ ⑤新規システムの出入口・バックアップ ⑥現行システムと共用できる設備の提案 ⑦運用時の全オペレーションの確認 ⑧運用作業効率化のための提案 ⑨運用保全用ドキュメントの確認 ⑩ログの表記、保存期間、削除タイミング ⑪運用者要員計画の立案 ⑫変更管理 / リリース / 資産管理運用 ⑬標準化に準拠しているか ⑭ルールに準拠しているか ⑮運用要件の提案	
15				■異常時、障害時の要件	①業務停止から復旧までの許容時間 ②異常発生時は既存の通報システムと連動 ③ホットスタンバイ、コールドスタンバイなどシステム構成 ④異常発生時の影響範囲、有事の際の関連部署と連携マニュアル、連絡網、通報 ⑤システム(サーバ、ホスト、ネットワーク機器)の自動監視 ⑥リモートログイン遠隔地から障害対応 ⑦縮退運用の優先順位とその影響内容 ⑧障害DBの整備 ⑨ジョブネットワーク構成	○	①類似システムの情報提供 ②障害対応方針の確認 ③障害検知方針の確認 ④障害時連絡体制の整備の確認 ⑤障害証跡の保持方針の確認 ⑥安定的な業務運営のための提案 ⑦標準化に準拠しているか ⑧ルールに準拠しているか ⑨障害対応要件の提案
16		移行	■保守要件	①定期保守 ②保守の範囲(回数) / 稼働報告の範囲 ③サービス時間が24時間365日の場合の保守時間の確保されている事 ④システムの稼働状況を報告する定例会 ⑤拠点の計画停電対応等の計画停止用の臨時システム	◎	①保守方針の確認 ②保守サイクルの確認 ③保守要件の提案	
17				■移行要件	①本番移行方法・切替日が明確になっている事 ・一括切替本稼働 ・順次切替本稼働 ・平行稼働	○	①影響範囲の明確化のための現行システムの情報提供 ②周知方法、周知先の妥当性の確認 ③移行・切替日の確認 ④移行方法の確認 ⑤影響範囲の確認 ⑥移行要件の提案
18				■内部統制、システム監査要件	①システム証跡を出力する	○	①類似システムのシステム証跡取得方法 ②各種証運用の確認 ③必要資料の確認 ④パッケージの選定、各種証証資料 ⑤内部統制要件の提案
19				■要件定義の合意	①要件定義の全項目を合意する	○	①教育 / 引継計画の確認と修正 ②教育 / 引継スケジュールの確認と修正 ③教育 / 引継資料の確認と修正 ④教育 / 引継環境の構築と確認

2.3 上流工程での運用部門の関与に関する教訓 (G3)

さらに、運用者が企画・要件定義に参加するときの心構えを以下にまとめた。

1: 運用者は経営戦略、情報システム戦略を十分理解すること (企画)

システム要件を確定し開発・運用を進めるにあたり「オーナーが真にシステムに求めているものは何か?」「なぜそのようなシステムが必要なのか?」を理解することが重要である。その根底・背景には必ず企業の経営戦略や情報戦略が存在する。

2: 運用者はプロジェクト企画書の作成段階から企画・要件定義者へのサポートを行うこと (企画)

プロジェクト企画書は企画・要件定義者のみで作成するものと決めつけてはならない。

運用者はプロジェクト企画書を企画・要件定義者や経営から直接説明を受け、適宜質問を行い、意見具申し、その内容を十分理解しながら適切なサポートを行わなければならない。

3: 運用者は運用方針/運用要件を定義すること (要件定義)

運用者は、運用方針/運用要件を「自分の言葉で」提案しなければならない。要件定義に運用要件が盛り込まれていないケースは多く見受けられる。開発者は、業務の要件定義には慣れているが、運用にかかわる要件定義は不得意である。企画・要件定義者も要求が曖昧になりやすい。

また、運用者が社外中心となっている場合は社外の運用者の意見を求めることも必要である。

効果

運用者が要件定義に参加することで、運用要件の漏れに起因する障害の多くが解決できる。

教訓

システムは運用されてこそ、その価値を生み出すものであり、システムの企画、要件定義段階において、無駄のない運用ができるような運用設計が大変重要である。このためには、企画・要件定義作業へ運用者が参加して確認する項目や作業を「見える化」することが重要である。

また、運用者は、要件を十分に説明できていない場合も多いので、開発するものは自分のシステムであることを自覚して、積極的に要件をだす必要がある。