
アプリケーション運用業務「ゼロ」に向けた 取り組み

出光興産（株）

■ 執筆者 Profile ■



坂巻 弘織

1991年 出光興産（株）入社
情報システム部 システム開発担当
1998年 商品管理システム開発・保守担当
2002年 経理システム再構築担当
2005年 商品管理システム再構築担当
2008年 運用企画PJ担当
現在 アプリ運用課長

■ 論文要旨 ■

当社では、システム構築・運用業務のQCD向上を目的として、2008年7月に情報システム部内の組織変更を行い、企画・開発・運用を分離した機能別組織に変更した。

業務アプリケーションの運用は、正確かつ低コストで実施することが求められるが、実施内容がみえにくく、なかなか改善されにくい状態であった。

この組織変更を機に、業務アプリケーションの運用業務は「無い」ことが理想的な姿として、アプリケーション運用業務の見える化を進め、適正な品質とコストで運用を行えるような改善を進めている。

当論文では、現在、ITILを手本に当社運用部門で進めている運用業務の見直しについて述べ、論文を読まれた皆様から忌憚のないご意見をいただき、今後の展開へと進めたいと思っている。

■ 論文目次 ■

1. はじめに	《 3》
1. 1 当社及び情報システム部の概要	
1. 2 当社の業務アプリケーションの概要	
1. 3 業務アプリケーションの開発・保守・運用体制	
1. 4 運用業務の概要	
2. 業務アプリケーション運用業務の問題点	《 5》
2. 1 オープン化に起因する問題点	
2. 2 組織構造に起因する問題点	
3. 業務アプリケーション運用業務見直しの目的	《 6》
4. 業務アプリケーション運用業務見直しの内容	《 6》
4. 1 規程類の見直しとアプリケーション運用業務の見える化	
4. 2 問い合わせ、障害の削減（インシデント管理、問題管理）	
4. 3 アプリケーションの適切な管理（構成管理）	
4. 4 手間とコストの掛からない運用設計	
5. 今後の課題	《 11》
6. おわりに	《 12》

■ 図表一覧 ■

図1 システム・部門別組織と機能別組織	《 4》
図2 実務要領における「実績報告と継続的な改善」の抜粋	《 7》
図3 問い合わせ対応業務の月次報告	《 8》
図4 A B E N D対応実績表	《 8》
図5 問い合わせ対応フロー	《 9》
図6 構成管理フロー（概略図）	《 10》
図7 アプリケーション運用設計書（雛型）	《 11》
図8 運用業務の継続的な改善フロー	《 12》
表1 業務プロセスと管理/計測指標	《 7》

1. はじめに

1. 1 当社及び情報システム部の概要

当社は石油、石炭等の資源開発から、石油精製並びに油脂製造、石油化学製品の製造・販売を主力とした事業を行なっている。石油製品ならびに石油化学事業においては、4 製油所・2 工場、国内18 事業所、海外36 事業所を有している。

また、石油製品以外にも、電子機能材料や農業薬品等の製造・販売事業や、各種化学工業用および環境保全用機械設備・機器の設計、施工といった事業を行なっている。

上記のような事業を支える情報システムは、設備機器の管理から各製品の受発注・物流・販売管理、あるいは人事・経理等のすべての事業領域を対象とし、自社計算センターにて、システムの企画・開発から運用・保守までを、約 100 名の情報システム部員が担っている。

特に、石油・石油化学に関する事業における仕入、売上から販売管理等の業務システムや人事・経理等の基幹システムは、1980 年後半にホストコンピュータ（以下、ホスト）にて、設計・開発され、永きに渡り運用してきた。当社では、業務システムをホストにて運用・管理するにあたり、処理のバッチジョブの登録からスケジュール運用・管理までの自動化運用ソフトを自社開発し、そのソフトを社外販売するまでの取組みを行なっており、システムの安定稼動・効率的な運用に積極的な取組みを行なってきた。

1. 2 当社の業務アプリケーションの概要

当社の情報システム部門では、出光興産の石油、石油化学事業の基幹業務システム、製造系システム、販売管理業務システム及び関係会社システムの企画、開発、運用を担っている。

従来は、メインフレームを中心にシステムを構築、運用していたが、2000 年から段階的にオープン化を進め、2004 年に経理システム、2007 年に販売・物流系システムをオープン系で再構築した。

現在、一部の業務アプリケーションがメインフレーム上で稼動しているが、計画的にオープン化を進め、最終的には全ての業務アプリケーションをオープン系に移行する計画である。

1. 3 業務アプリケーションの開発・保守・運用体制

業務アプリケーションの開発・保守・運用体制は、図1のとおり開発と運用を分離した体制（以下、機能別組織という）と開発と運用を一体化した体制（以下、システム・部門別組織という）があり、これまでそれぞれのメリットを考え、以下のとおり体制の変更を実施してきた。

～1994 年度：機能別組織

1995 年度～：システム・部門別組織

2008 年度～：機能別組織

1995 年度のシステム・部門別組織への変更は、開発と運用を一体化することにより、運用を意識したシステム設計を実施することを目的に行った。その結果、開発担当者は、

運用に手間とコストのかからないシステムを構築することを意識して優れたシステム設計ができるというメリットがあった。

一方、①開発担当者が運用業務も担うことにより、開発時のコスト・納期の制約により、運用に必要な一部の機能の実装を省き、運用作業が増える。②オープン化が進んだ期間に横断的に運用を担う部門が無かったことにより、システム毎の運用ルールが作成され、オープン系システム運用を図りにくい状態だった。

前述のような背景をふまえ、昨年度、「アプリケーション運用機能の強化」と「内部統制における開発と運用の職責の分離」を目的に、機能別組織に変更した。

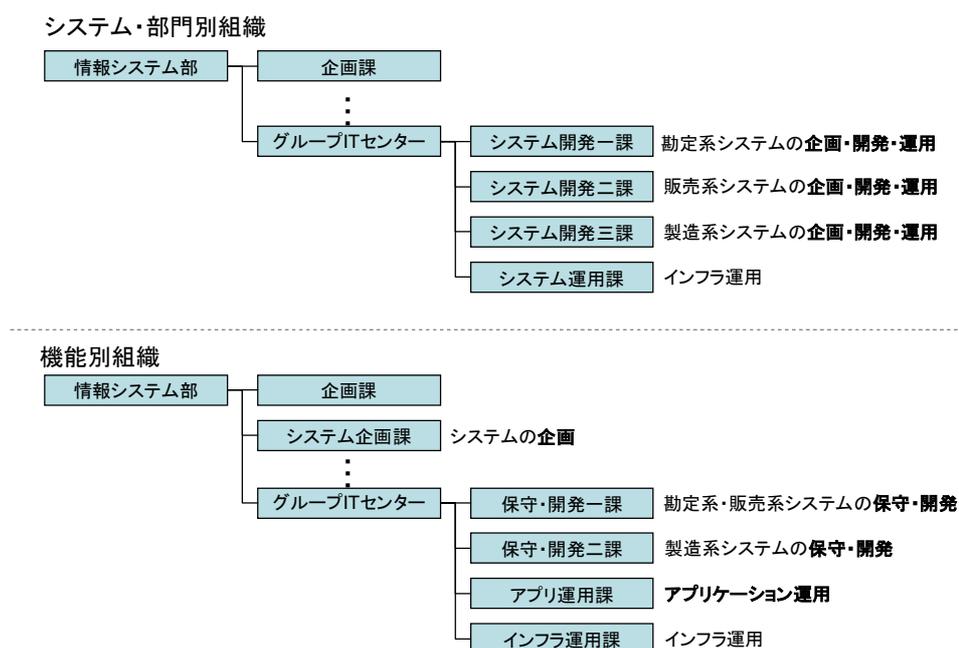


図1 システム・部門別組織と機能別組織

1.4 運用業務の概要

当社では、運用業務を「インフラ運用課」と「アプリ運用課」を2つの組織で担っている。インフラ運用課は、ハードウェア、ミドルウェア、ネットワーク等のインフラ環境管理、キャパシティ管理等を担い、アプリ運用課は、保守・開発担当課が構築した業務アプリケーション運用業務を担っている。

業務アプリケーション運用業務は、「処理スケジュール管理」、「構成管理」、「利用者からの問い合わせ、作業依頼対応」、「障害初期対応」である。バッチ処理は運用管理ツールにより自動化しているものの、一部手作業によるジョブ実行や処理結果の確認も実施している。現在、これらの運用業務に、相当な工数とコストがかかっている。

昨年度、機能別組織になったことを機に、業務アプリケーションの運用業務の個々の作業内容を確認したところ、アプリケーションの改善や、運用ルールを見直すことにより、

作業そのものの削減や効率化を図ることができるものが数多くあることが再確認できた。

また、問い合わせ、障害などは、利用部門の業務効率化の阻害要因にもなっている。

本稿では、「アプリケーション運用業務は、無いことを理想的な姿」として、昨年度から取り組んでいるアプリケーション運用業務の見直しについて述べていきたい。

2. 業務アプリケーション運用業務の問題点

2. 1 オープン化に起因する問題点

メインフレームの運用では長年の経験やノウハウに基づき、運用ルール及び運用ツールが整備されていた。一方、オープン系システムは、ノウハウが少ない中で、段階的に小規模なシステムから構築してきた経緯から、導入タイミング毎に運用ツールが異なり、運用ルールや運用ドキュメントの整備・標準化が進んでいなかった。

2. 2 組織構造に起因する問題点

2. 2. 1 運用業務の実態が見えない。

2007年度までは、「システム・部門別組織」であり、開発担当者がシステムの企画～開発～運用までを担っていた。私の経験では、システムの企画、開発、運用業務を担っている体制では、開発担当者の業務は、企画及び開発業務が中心であり、運用業務は開発の付帯業務という意識になってしまう。過去から開発業務の標準化は進めてきたが、運用業務の標準化は遅れていた。

アプリケーション運用業務の標準として、運用実務を行う上で必要な手続きを定めている社内規定「システム運用管理要領」が存在する。この要領には、運用業務の計画（Plan）、実施（Do）手続きはルール化されているが、定期的に運用実態を把握し（Check）、改善（Action）を図るプロセスについては定めていなかった。

要領に定めが無いからPDCAが回らなかったというわけではないが、これまで組織として「運用を見える化し、改善を図る」取り組みが十分ではなかったことは事実である。

運用業務の実態が見えないがゆえに、組織として計画的に運用の改善を図ることができず、運用の改善は一部の運用に対する意識の高い担当者に依存する状態となっていた。

問い合わせ対応の例では、利用者からの問い合わせは、個々のシステム担当者が受け付け、対応を行っていた。問い合わせの管理は各課、担当者毎にバラバラであり、対応状況や記録の方法が統一されていなかったため、問い合わせ対応実態が見えず、改善が進まないという状況にあった。また、窓口が統一されていないことから、担当者が席を外している場合には、問い合わせがたらい回しにされるケースもあった。

ABEND対応の例では、管理ルールが明確ではなく影響度の大きいものについては、根本原因の究明と恒久対策の立案・実施を行っていたものの、全体像を把握した対応は実施できていなかった。

2. 2. 2 統一的なソース、ドキュメントの管理が不十分

オープン系システムの導入当初は、アプリケーションが少なかったことから、特に構成管理ルールを定めていなかった。その後の新規システム構築においても、そのままの状態であり、個々のアプリケーション毎にソース及びドキュメントを管理していた。

具体的には、オープン系システムのプログラム資産、ドキュメント資産は、各開発担当課又はシステム毎に個別のサーバで管理していた。

このような状況では、最新資産の取り扱いを誤り、低いバージョンを本番登録するなどの信頼性の低下や、個々のアプリケーション毎に作業要員を抱えることとなり、非効率な状態を招く可能性があった。

2. 2. 3 運用要件の実装が不十分

新規システムの構築では、開発の中盤以降に必ずといって良いほど仕様変更要求や品質の問題から納期とコストの制約にぶつかる。その際に利用者からの要求と運用要件を天秤にかけてどちらを切り捨てるか判断をしなければならないのだが、その多くが運用要件である。

また、運用設計の標準化も十分ではなかったため、中には運用設計、実装が不十分なシステムも存在する。

「機能別組織」であれば、構築が終わり運用に乗せる段階で運用部門の審査があり、運用要件を満たしていなければ改善を求めることができる。しかしながら、「システム・部門別組織」では、構築した担当者が運用も担うため、本番稼働後に自分が作業を実施し、徐々に改善を図れば良いという意識があるために、運用要件の実装が切り捨てられるという状態になってしまうと考えられる。

3. 業務アプリケーション運用業務見直しの目的

アプリケーション運用業務は、2で述べたような問題点があり、アプリケーション運用コストは、メインフレーム中心で運用していた当時と比較して倍増、問い合わせや軽微なトラブル対応の件数も増加傾向にあった。

これらの状況を踏まえて、アプリケーション運用業務見直しの目的を以下の2点とし、ITILを手本に改善に取り組んだ。

- ①アプリケーション運用業務の見える化を進め、開発～運用プロセスのPDCAを回すことにより継続的に改善を図る環境を作る。
- ②サービスレベルを明確にし、適正なコストと品質で運用を行う。

4. 業務アプリケーション運用業務見直しの内容

4. 1 規程類の見直しと業務アプリケーション運用業務の見える化

運用業務を見える化し、PDCAを回すために「実績の把握と継続的な改善」に関する内容を、「システム運用管理要領」及び「実務要領」に盛り込み、管理計測指標に基づき、運用実績の把握と改善に取り組んでいる。（図2に実務要領における「実績報告と継続的な改善」の抜粋を、表1に業務プロセスと管理/計測指標を示す。

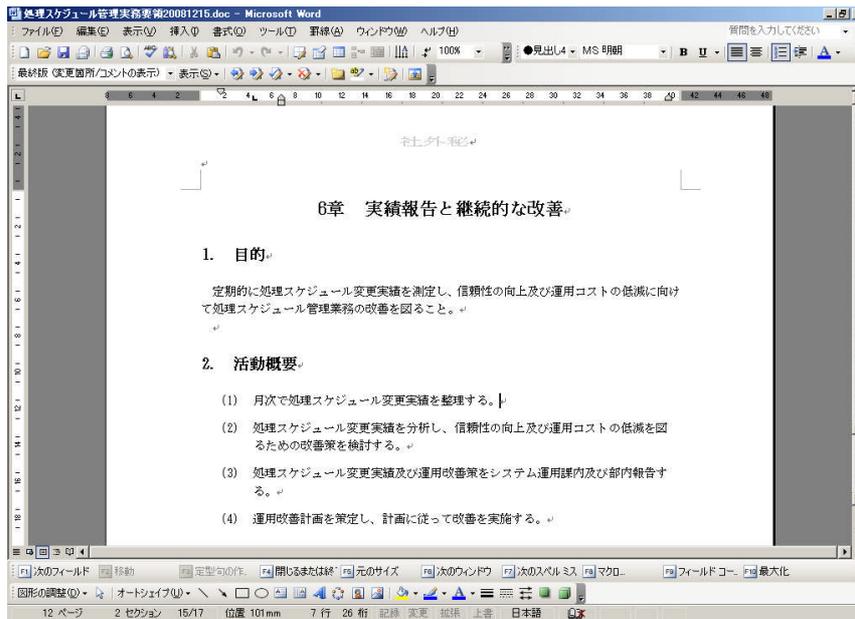


図2 実務要領における「実績報告と継続的な改善」の抜粋

表1 業務プロセスと管理/計測指標

業務プロセス	管理/計測指標
処理スケジュール管理	<ul style="list-style-type: none"> ・ 分類別の処理スケジュール変更件数 ・ 変更の成功件数/失敗件数 ・ 手続き上の不備があった変更の件数 ・ 未処理の変更件数 ・ 緊急変更の件数 ・ 変更に関わるコスト/時間
インシデント管理	<ul style="list-style-type: none"> ・ (コントロールの指標として) インシデントの総数 ・ 段階毎のインシデントの内訳 (記録済み、対応中、エスカレーション中、クローズ済みなど) ・ 現在の未処理のインシデントの数 ・ システム別のインシデントの数と割合 ・ カテゴリ別のインシデントの数と割合 ・ 優先度=1のインシデントの数と割合 ・ 優先度コード毎のインシデントの平均解決時間 ・ インシデントあたりの平均コスト (サービスデスク、2次・3次サポートを含む) ・ 再提起されたインシデントの数及び全体に占める割合 ・ 不適切に分類されたインシデントの数と割合 ・ 最初の窓口で解決し、クローズしたインシデントの割合 ・ 類似のインシデントの数と割合 ・ 上記のほか、信頼性の向上、運用コストの低減のために計測が必要な項目は、適宜定める。
問題管理	<ul style="list-style-type: none"> ・ (コントロールの指標として) 問題の総数 ・ 未解決の問題とその傾向 (変動していないか、増減があるか) ・ 問題の処理にかかったコスト ・ 重大な問題の数
構成管理	<ul style="list-style-type: none"> ・ 構成管理計画 ・ 構成管理変更件数

この見える化の取り組みにより、問い合わせ対応業務の実態、A B E N D対応業務の実態等がわかり、改善を図るためのベースラインを明らかにすることができた。

図3に問い合わせ対応業務の月次集計の一部、図4にA B E N D対応実績表の一部を示す。

現時点での効果は、運用担当者が実態を数値として再認識したことにより、業務量削減、コスト削減の改善目標を定量化することができ、改善意識を一層高めることができた。また、改善の優先順位付けができ、着実に改善が進み始めた。

改善の一例を挙げると、月々の処理スケジュール調整業務を無くするためのシステム改善や、ユーザ登録業務フローの見直しなどにより、年間数百万円程度のコスト削減を図っている。改善対象業務は、数多くあるため今後の改善により大きな効果を見込んでいる。

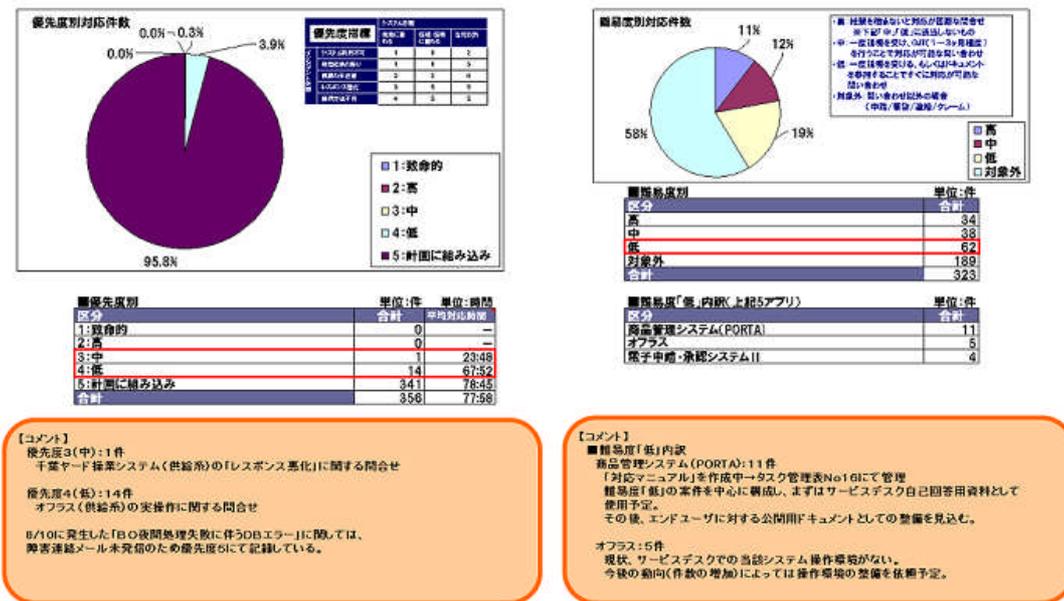


図3 問い合わせ対応業務の月次報告

項目	発生日	発生時刻	メッセージ	対象サーガ名	システム名	PJ名	JOB	JOB	連絡先	所属	処置	処置の	直接原因
1	4月2日	0時06分	Systemwalker[14788] MpJbsch: ERROR: 0330: The job net has abnormal ended. JobNetComment= 支払月報作成 JobNetName=WBI 01000 Code=4 ProjectName=WZ10	0100000C-nr-erth	商品	WZ10	WBI 01	受払		アプリ運用課	【当日】 該当コード以外の帳票については処理されますので、本日の処置としてはアペンド処理をスキップして後続を流しています。 【翌日】 出力されなかった分については日中に帳票基盤のマスクを設定の上、前日出力のCSVより該当分をエディタで抜き出し、本番の出力フォルダに再セットし帳票基盤に連携した。	1.1	在庫支払月報(仮簿)作成処理にて発先のコードが帳票基盤のマスクに存在しないという事でアペンドした。
2	4月2日	23:11	AP MpJbsch: エラー: 9002:ジョブネットジョブネットが異常終了しました。ジョブネット名称=XLID0501 ジョブネット名=XLID0501 コード=1 プロジェクト名=C1100	0001000X	薄清油	CL100	XLID05	XLID0501		アプリ運用課	原因は一時的なメモリ不足(SYSTEM_NO_ROLL)対応についてはジョブ再起動	2.5	一時的なメモリ不足
3	4月3日	1:23	AP MpJbsch: エラー: 9002:ジョブネットジョブネットが異常終了しました。ジョブネット名称=XLID0511 ジョブネット名=XLID0511 コード=1 プロジェクト名=C1100	0001000X	薄清油	CL100	XLID05	XLID0511		アプリ運用課	原因はアダプタが見つからない(CALL_FUNCTION_OPEN_ERROR)でした。対応については時間的に対応に間に合わないかと判断し、放棄(サーバーストップ)致しました。	01	不明
4	4月3日	12:06	AP TWS: エラー: 905:ジョブが異常終了しました。ジョブ名=B400-TWS DHMM00SD_TW400_URUKAKE_RES OVERY(021-400)コード=B400	0020000X	化学	化学	BA00	TWS DHMM 00SD TW400 URUKAKE	秀樹	アプリ運用課	スケジュール誤り CL400については期待外においても稼働不要であったので対象データが存在せずにアペンドした。 対応不要だった。	01	スケジュール誤り
5	4月3日	23:00	AP MpJbsch: エラー: 9002:ジョブネットジョブネットが異常終了しました。ジョブネット名称=XLID0501 ジョブネット名=XLID0501 コード=1 プロジェクト名=C1100	0001000X	薄清油	CL100	XLID05	XLID0501		アプリ運用課	原因は一時的なメモリ不足(SYSTEM_NO_ROLL)対応についてはジョブ再起動	2.5	一時的なメモリ不足

図4 A B E N D対応実績表

ール導入も検討したが、検討するだけでも相当な時間とコストがかかることが見込まれた。本格的なツール導入は、インシデント管理業務プロセスが軌道に乗ってからも遅くないと考え、早期にインシデントを管理できる体制を立ち上げることを優先し、社内ポータルで利用しているツールを活用し、数日間でインシデント記録を残す仕組みを構築した。

これにより、ほとんどコストもかからず、管理開始後の分析要件の追加変更にも柔軟に対応できるツールとして活用している。

問い合わせをを分析した結果、問い合わせ件数上位5システムで問い合わせの7割を占めており、問い合わせ件数上位5システムを優先して対策を検討・実施するように優先順位付けすることができた。今後、問い合わせ件数が上位のものから集中的に対策を行うことにより、問い合わせの削減、対応の効率化を図っていく予定である。

4.3 アプリケーションの適切な管理（構成管理）

統一的な構成管理ルールを「構成管理実務要領」に定め、ソース及びドキュメントの一元管理を実施することとした。

図6に構成管理の概略図を示す。

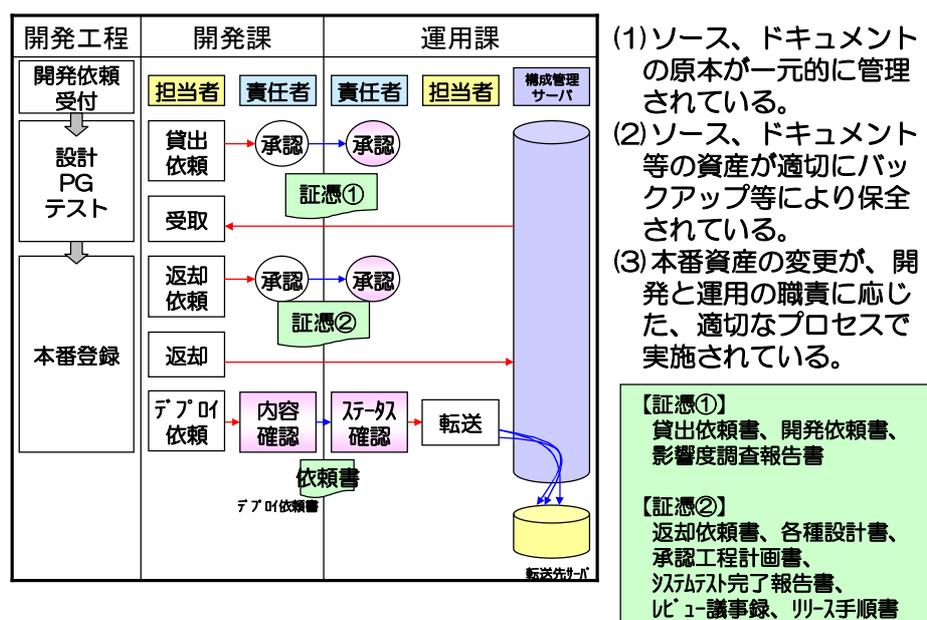


図6 構成管理フロー（概略図）

新たな構成管理ルールは、以下の点に留意して構築した。

- ・ IT全般統制の要件を満たすこと
- ・ 業務の効率性を阻害しないこと
- ・ 開発から運用への引継ぎをスムーズに行えること。
- ・ 利用する構成管理ツールの制約事項を考慮した運用ルールとすること。
 （ツールには手を入れないこと）

現在は、構成管理ツールへの資産の移行中の段階であり、新たな構成管理ルールで運用しているものは一部のシステムであるが、順調に新たなルールへの切替が行われている。

4. 4 手間とコストの掛からない運用設計

4. 1～4. 3の取り組みはシステム稼働後の取り組みだが、適正な品質とコストで運用するためには、設計段階でより良い設計ができていることが重要である。これまで、大規模システムの構築においては、個々の構築PJ毎にアプリケーション運用設計内容の検討がなされていたが、共通的な設計内容は、整備されていなかった。

運用設計の考え方が標準化されていないと、アプリケーション毎に特殊な運用が発生するケースがあり、標準的な考え方として「アプリケーション運用設計書」の雛型を整備し、設計すべき事項の標準化を図った。

図7にアプリケーション運用設計書（雛型）の目次を示す。

目次	システム		サブシステム		プロセス		プログラム		社名		作成日	
	ID		ID		ID		ID					
1章 設計書の対象範囲												
1.1 当該設計書の対象範囲												
2章 処理スケジュール												
2.1 全体スケジュール												
2.2 夜間バッチフェンダー												
2.3 夜間以外のバッチ処理												
2.4 ジョブネットとバッチサーバーとの関連												
2.5 オンライン起動/停止概要												
2.6 定時実行スケジュール変更と運用制御												
2.7 特殊な処理												
3章 外部システム連携												
3.1 連携概要												
3.2 他システム各種フォーマット												
3.3 夜間バッチでのデータ連携時の稼働 非稼働の対応												
4章 業務バックアップリカバリー												
4.1 業務バックアップの概要												
4.2 夜間バッチにおけるリカバリーポイント												
5章 データ回避・削除運用												
5.1 DB保存期間超過データの削除												
6章 運用制限												
6.1 運用に伴う制限事項												
7章 障害対応												
7.1 ジョブ異常時の対応												
7.2 連携異常時の対応												
8章 リラン対応												
8.1 リラン対応方針												

図7 アプリケーション運用設計書（雛型）

また、日々の運用業務の中で見つけた運用上の改善事項、システムの構造上の課題を定期的に整理し、定期的に当該ドキュメントや規程類への反映を推進している。

上記ドキュメント類の作成後に構築したシステムは少ないが、この取り組みにより、運用要件の漏れが無くなり運用業務が削減されることを期待している。

5. 今後の課題

2008年度から仕組みの整備を中心に実施し、2009年度から運用を開始した。現在は、蓄積した運用実績（インシデント、処理スケジュール変更など）を分析し、改善を進めている段階である。

今後は、実務要領に基づくこれらの運用を定着させるとともに、図8に示すフローで運用部門だけではなく企画・開発部門と一丸となって「運用業務を限りなくゼロに近づける」ための施策を検討・実施するように本取り組みを推進していく。

また、適切な品質とコストで運用を実施するためにも、利用部門との間でSLAを明確にして、共通的な指標を持って運用業務を担えるように推進する予定である。

計画的かつ継続的に運用業務及び業務アプリケーションの改善を図り、最終的に運用業務をゼロにするように取り組んでいきたい。

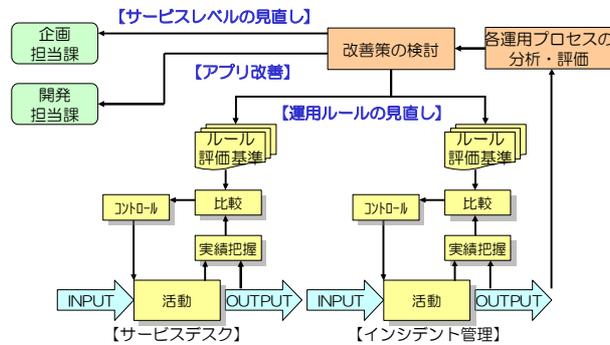


図8 運用業務の継続的な改善フロー

6. おわりに

開発業務は本番稼動をさせれば終わりがくるが、運用業務はそのアプリケーションが存在する間は終わりが無い。また、運用業務は、正しく運用できて当たり前というイメージが強く、企画や開発に比べて「実態が見えず、評価もされにくい」仕事でもある。

今回の取り組みは、まだ始まったばかりであり、具体的な成果を示すことはできていないが、この取り組みを定着させ、企画、開発部門、利用部門へ積極的に提案、提言できる運用部門にしていきたいと考えている。

是非、論文を読まれた方々の忌憚りの無いご意見をいただきたく思っている。

最後に、この取り組み及び本稿の執筆にいろいろとご助言いただいた関係各位に感謝の意を表します。