運用・保守業務の組織再構築と作業改善

ジスインフォテクノ（株）

■ 執筆者Ｐｒｏｆｉｌｅ ■

大久保　綾香

2003 年　ゼオン情報システム(株)　入社
2005 年　ジスインフォテクノ(株)　会社名変更
2017 年　現在
業務システム開発部第1開発チーム配属

■ 論文要旨 ■

開発チームでは、開発案件と運用保守業務を掛け持ちしていたが、年々運用保守業務の負担が大きくなり、開発案件に専念できないことが問題となっていた。2005 年の会社設立当初はシステム開発者が稼働後も運用保守業務をすることは効率的であった。年月が経ち、他人が開発したシステムの運用保守業務を担当することが多くなり、対応に時間がかかり、プロジェクトが遅延することが起きていた。運用保守管理者を新たに設置し、リーダーへの報告を円滑とするような新たな業務フローを定義した。利用者からの依頼内容の標準化、運用保守業務の作業工数を正しく入力することで、運用保守業務の作業状況を正確に把握できるようにした。その上で、日々の進捗確認における無駄を排除し、チーム全体で 1 ヶ月に 50 時間の工数を、10 時間に削減した。問題発生を見える化し、一つずつの運用保守業務を短縮することができ、全体的な工数削減ができた。

■ 論文目次 ■

１．はじめに……《 3 》
１．１　当社の概要
１．２　ITIL の導入と限界

２．第１開発チームの課題……《 4 》
２．１　運用保守業務が優先され、開発業務が遅延する
２．２　運用保守担当者として協力会社要員を配置しても作業量が減らない
２．３　運用保守業務の管理ができていない

３．運用保守業務の改善取り組み……《 5 》
３．１　運用保守業務全体の見直し
３．１．１　運用保守業務の体制変更
３．１．２　運用保守業務のフロー変更

３．２　日々発生する運用保守業務で実施すべきことを明確化
３．２．１　運用保守管理者による要件確認
３．２．２　依頼内容の標準化
３．２．３　工数を正しく入力する

３．３　運用保守業務の効率化のための改善
３．３．１　日々の進捗確認における無駄の排除
３．３．２　繰り返し作業の協力会社要員への移管

４．取り組みによる効果……《 9 》

５．今後の展望……《 11 》

６．おわりに……《 11 》

■ 図表一覧 ■

図１　ジスインフォテクノ組織体制図 ……《 3 》
図２　第１開発チームのインシデント数 ……《 4 》
図３　管理しているシステム数とメンバー数の推移 ……《 5 》
図４　運用・保守体制の変更 ……《 6 》
図５　体制見直し後の運用保守業務フロー_インシデント管理での例 ……《 7 》
図６　2016年度の開発チームの運用保守業務の作業工数 ……《 10 》
図７　2016年度の開発チームのプロパーの総工数に対する運用･保守工数の割合 ‥《 10 》
表１　運用保守業務の工数入力事例 ……《 8 》
表２　日々の進捗確認項目の見直し ……《 9 》

1. はじめに

1. 1　当社の概要

ジスインフォテクノ(株)(以下　当社)は、製造業の日本ゼオン(株)と富士通(株)の合弁会社で、日本ゼオン(株)グループのシステム開発、運用保守業務を行っている。日本ゼオン(株)は、合成ゴム、合成ラテックス、フィルムなどの製品を製造、販売している化学メーカーで、社員数は約 3,000 名である。

当社では、「システム運用部」と「業務システム開発部」で、システム開発、構築、運用保守業務を行っている。当社の組織体制図を図 1 に示す。当社では、業務システムは約 60 システムあり、館林データセンターに構築されているサーバは約 180 台である。

「業務システム開発部」の「第 1 開発チーム」では、他のチーム同様、メンバー全員が開発案件と運用保守業務を掛け持ちしている。「第 1 開発チーム」は 6 人のプロパーで、46 の既存システムの運用保守業務を行っている。トラブルや問い合せがあった際には、業務影響などトラブルの優先順位に関係なく、システム担当者の判断で、随時対応をしていた。

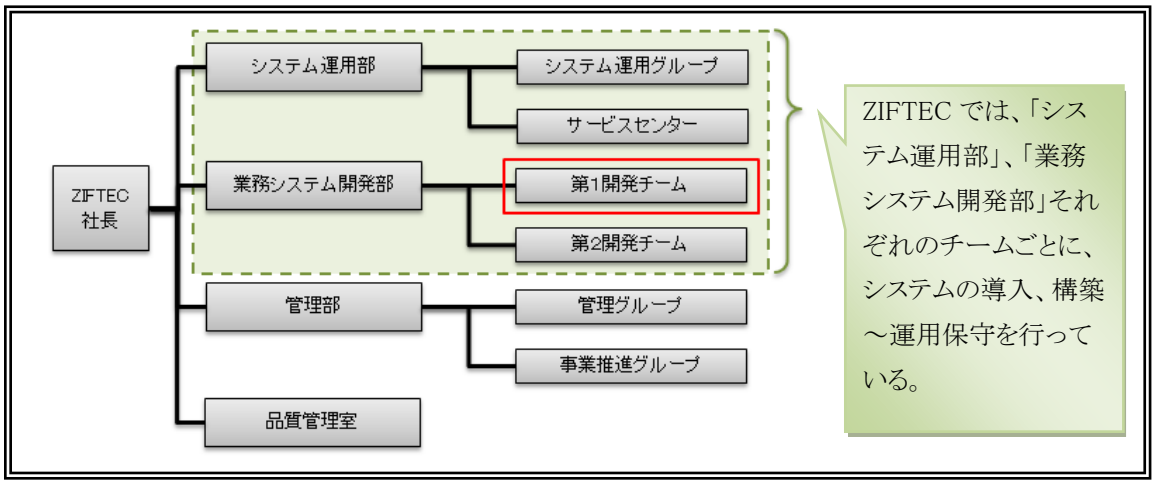

図 1　ジスインフォテクノ組織体制図　※ジスインフォテクノ(株)(略称:ZIFTEC)

1. 2　ITIL の導入と限界

当社では、2006 年の ITIL の導入により、運用プロセスが標準化されたことで、インシデント発生時の問い合わせ窓口は、ヘルプデスクに統一されている。ITIL 導入前の 2005 年以前に比べ、何件のインシデントが発生しているかを把握できるようになった。

運用プロセスは標準化されたものの、年々システムが増加することに比例して、インシデントは増える傾向にある。第1開発チームのインシデント数を図2に示す。2005 年より前は、ITIL 導入前であるため、チームで発生するインシデントの数は、把握できていなかった。ITIL 導入直後の 2006 年は、インシデント数は約 60 件/月だったが、2014 年のインシデント数は、システム数の増加とともに、トラブル、QA 共に、約 80 件/月と増え続けていることが分かり、保守業務の負荷が増加していることが明確になった。

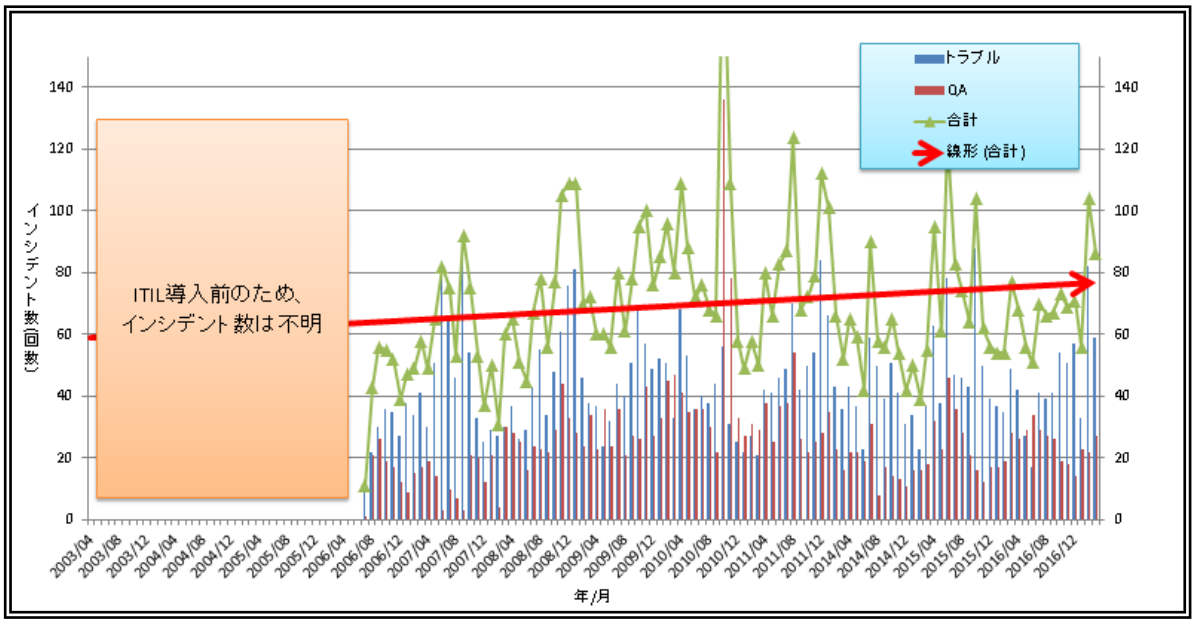

図2　第1開発チームのインシデント数

2. 第1開発チームの課題

2. 1　運用保守業務が優先され、開発業務が遅延する

開発業務と運用保守業務の業務負荷状況の管理は、システム担当者個人ごとで行っていた。2002 年は、システム開発者が稼働後も運用保守をするルールになっており、システム開発者が一番システムの概要を理解しているため、担当システム数が少ない間は、効率的に運用保守業務を行うことができていた。

年月とともに、人の入れ替わりやシステムの増加により、自分で開発したシステム以外のシステムの運用保守業務を担当することが多くなった。2002 年の第1開発チームで管理しているシステムは、15システムあり、全てのシステムにおいて開発に関わった担当者が、そのまま運用保守業務も担当していた。現在は、2002 年に比べシステムは 3 倍以上に増えたが、プロパーの人数は横ばいである。システム担当者がシステム導入時から開発に関わっているシステムは全システム数の 1/3 以下になっている。第1開発チームで管理しているシステム数とチームメンバー数の推移を図3で示す。

このような状況の変化の結果、運用保守業務が発生するたびにシステムの仕様を確認する必要があり、ちょっとした問い合せでも回答に時間を要するようになっていた。 2002 年は、電話で問い合せがあった場合、電話を受けたシステム担当者がシステムの仕様も把握しているため、その場で回答することができた。現在は、一旦問い合せの回答を保留し、設計書から仕様を確認し、場合によっては検証環境で動作を確認するなど、問い合せ1件当たりの対応時間が増えるようになった。

稼動中のシステムで突発的なトラブルが発生した際、システム担当者はそれぞれの対応に時間がかかってしまい、開発案件のプロジェクトに工数を割けなくなり、プロジェクトが遅延する。小規模のシステム改善ですら、納期を守ることができなくなる。といったことがチーム内で起き始めた。チーム内の誰もが自分の担当しているシステムで手一杯になり、他のメンバーを手伝える余裕もない状況であった。スケジュールが立てられないため、利用者への要望の回答ができないという悪循環に陥っていた。

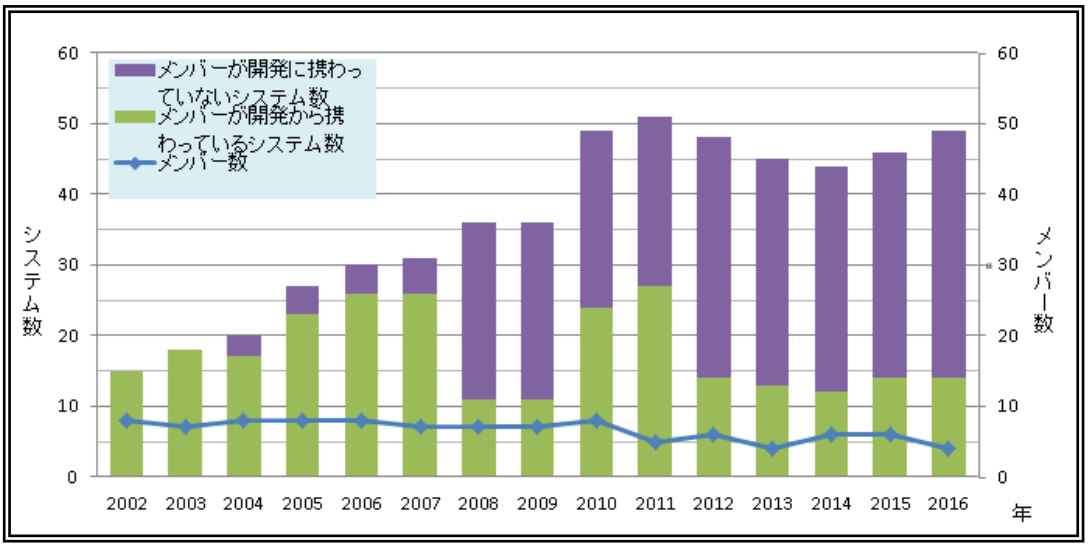

図3　管理しているシステム数とメンバー数の推移

2．2　運用保守担当者として協力会社要員を配置しても作業量が減らない

利用者の要件をとりまとめ運用保守担当者に的確な作業指示を出し、手戻りなく運用保守業務を進めることが課題である。運用保守業務が増えているため、運用保守担当者として協力会社要員を配置したものの工数削減の効果があまり得られなかった。運用保守担当者は利用者から要望された作業をそのまま行うため、利用者の要件が曖昧な場合でも、勝手に判断してメンテナンスすることがあり、後にインシデント発生原因となることがあった。

システム担当者は、運用保守担当者がいるのに、運用保守業務が減らないと不満に感じていた。運用保守担当者は、的確な指示がないまま作業を任されるため、最初のリリースでは不十分で手戻りになることもあり、双方のモチベーションも悪影響が出ている状況になっていた。

2．3　運用保守業務の管理ができていない

チームの運用保守業務が増えているのだが、突発的なインシデント業務が増えているのか、問題管理の原因究明や対策の時間が増えているのか、チームメンバがどんな運用保守業務を行っているかを把握できることが課題である。チームの工数は、社内システムで管理していて、運用保守業務は“インシデント工数”、“問題管理工数”、“メンテナンス作業”…というように、作業ごとに分けて入力するようになっているが、メンバーそれぞれの判断で入力していたため、具体的にどのような運用保守業務が発生しているか分からず、対策が打てなかった。

3．運用保守業務の改善取り組み

3．1　運用保守業務全体の見直し

3．1．1　運用保守業務の体制変更

状況の変化により、チームに運用保守管理者、運用保守担当者を新たに設置し、リーダーへのエスカレーションが円滑となるようにした。システム担当者から運用保守管理者、運用保守担当者へシステムの運用業務の引き継ぎを行った。運用保守体制の変更を図4に示す。

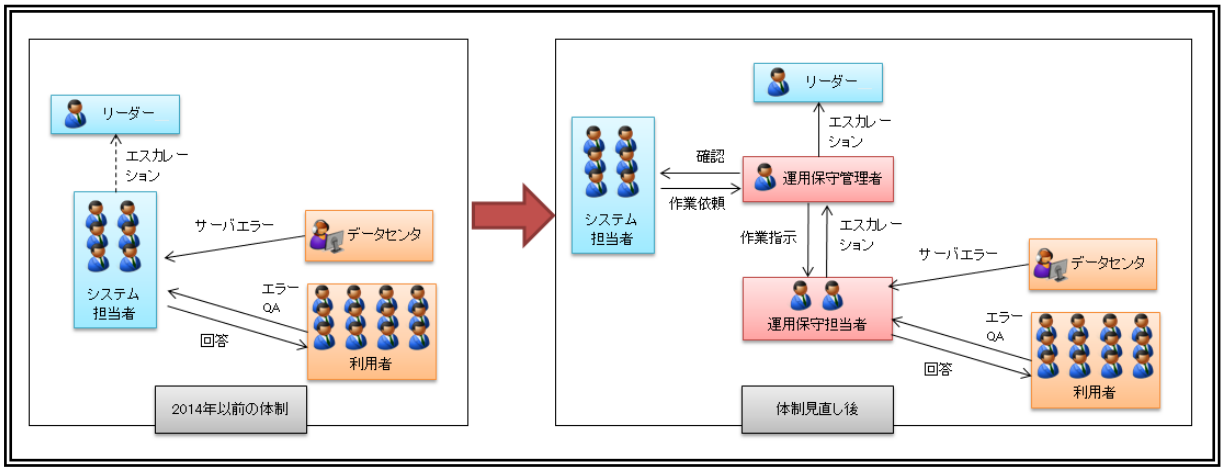

図4　運用保守体制の変更

2014 年度は第1開発チームで管理しているシステムは、46 システムあった。46 システムの中から、運用保守業務数(インシデント数、QA 数、問題管理対応数、小規模メンテナンス数)を集計し、運用保守業務の数が多い上位 16 システムを対象として、システム担当者から、運用保守管理者、運用保守担当者に引継ぎを行った。運用保守業務の数が多い上位の 16 システムは、全体の運用保守業務数の 72.1%を占めていた。システム担当者は、運用保守担当者がインシデントに回答できるか、小規模メンテナンス依頼に対応できるかの観点で引継ぎを行った。運用保守管理者は、利用者との要件確認、運用保守担当者への作業指示を行っていく中で、運用保守業務に無駄がないか、もっと作業を効率良く対応できないかを考えた。運用保守業務全体を見ているため、運用保守担当者の作業負荷を把握でき、必要に応じて、リーダーへエスカレーションを行った。

3．1．2　運用保守業務のフロー変更

インシデントがエスカレーションされた際に、それぞれのシステム担当者がインシデントの対応をしていたが、全てのシステムで運用保守担当者が対応し、判断が必要になった場合は、直接システム担当者へ確認するのではなく、運用保守管理者の指示を仰ぐように業務フローを変更した。体制見直し後の運用保守業務フローのインシデント管理での例を図5に示す。

運用保守業務フローは、極力システム担当者への回送を減らすように考えた。運用保守担当者では対応方法が不明確の場合や、要件確認が必要な場合は、運用保守管理者が対応を行い、運用保守担当者に作業指示を行った。利用者から、直接システム担当者へ問い合せが来た場合は、システム担当者が対応せず、運用保守管理者へ回送してもらい、システム担当者の運用保守業務の負荷を減らした。利用者へはトラブルや問い合せはサポート窓口へ問い合わせいただくように、都度アナウンスをした。

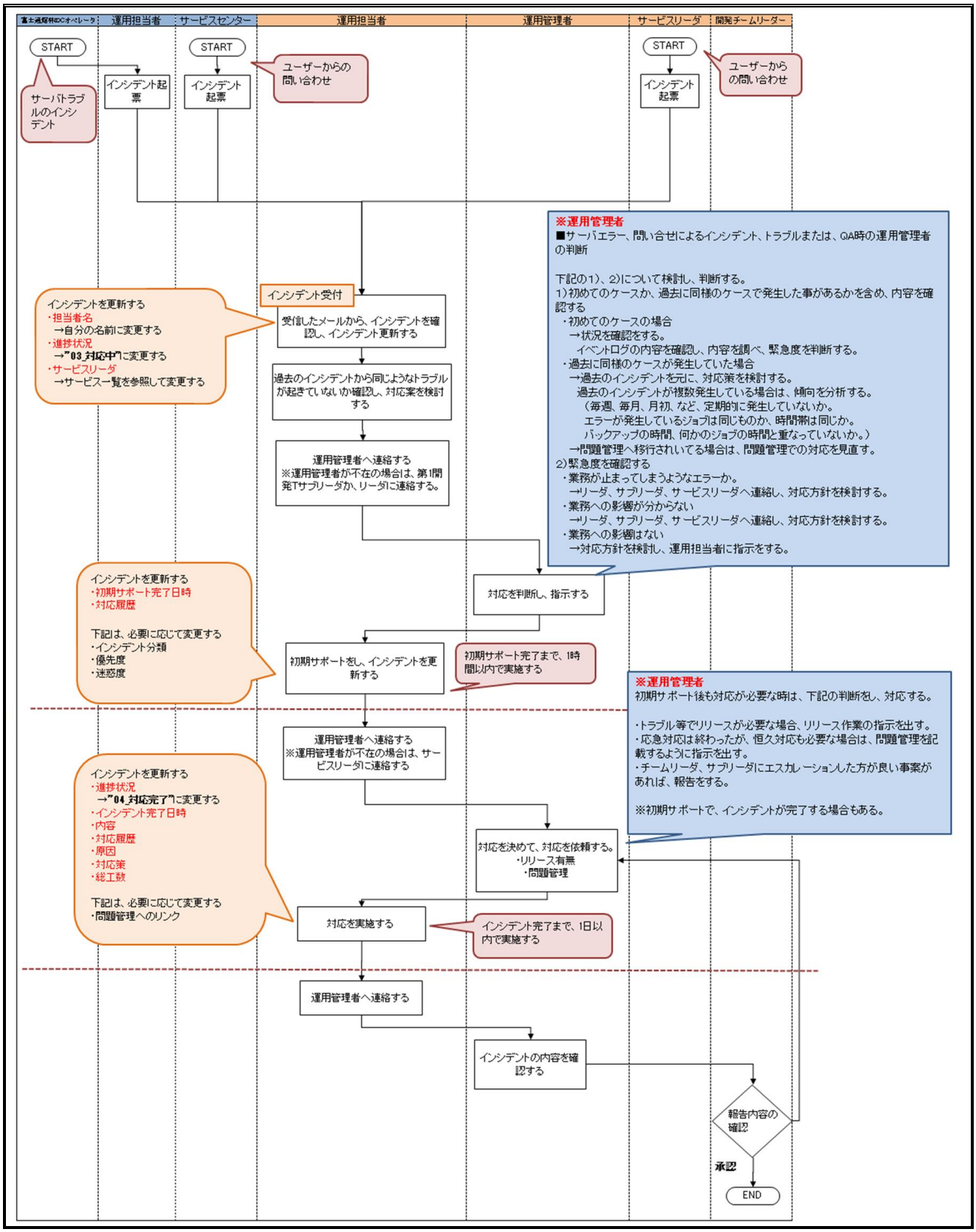

図 5　体制見直し後の運用保守業務フロー_インシデント管理での例

3．2　日々発生する運用保守業務で実施すべきことを明確化

3．2．1　運用保守管理者による要件確認

曖昧なメンテナンス作業の依頼に対して、運用保守担当者が要件確認まで行うことは難しかったため、運用保守管理者が要件確認を行うようにした。運用保守管理者の手間が増えたが、依頼者と運用保守担当者の間で何度も要件の確認をするやり取りが減った。メンテナンス作業の依頼の中には何のために作業をする必要があるのか分からない依頼もあり、要件を確認してみると、当初の利用者の依頼作業とは別な提案で対応した方が利用者も良いし、運用保守業務をする側も作業時間を減らせた。という場合があった。

3．2．2　依頼内容の標準化

定型化している手順のメンテナンス作業は、あらかじめ改修に必要な情報を利用者に連絡しておくように運用を変更し、依頼ごとに確認するやりとりを減らした。以前は依頼があった際に改修に必要な情報が揃っていないことがあり、リリース日の調整やリリース準備が停滞し、希望納期に間に合わないことがあった。利用者は改修に必要な情報を事前に揃えてから、メンテナンスの依頼していただくことで、運用保守担当者は作業計画が立てやすくなり、スムーズにリリースまで業務を実施することができるようになった。

3．2．3　工数を正しく入力する

チームメンバーに運用保守業務の作業工数を的確に入力するように指導し、工数管理を明確にした。工数を入力するオーダーはあるが、「小規模改善」「運用業務」の2つの観点でしか、管理されていなかった。運用保守の工数はすべて「運用業務」となっていたため、メンバーがどんな運用保守作業を行っているかが分からず、対策が立てられなかった。また、メンバーにヒアリングするとどのオーダーに入力したら良いか分からず、「プロジェクトの開発」の工数、「小規模改善」の工数以外は全て「運用業務」の工数としている場合もあった。「運用業務」の内訳を「QA　対応」「障害対応」「問題管理対応」「予防保守」「その他保守」の分類で正しく入力できるように、具体的な事例をあげて、チームメンバーに周知させ、運用保守業務の中で何の作業に時間がかかっているかが分かるようにした。運用保守業務の工数の入力事例の一覧を表1に示す。

大分類	大分類	中分類	小分類	備考・参考例
小規模改善	小規模改善			小口メンテナンス作業 ※見積り後、中止になった場合の工数も、このオーダーにつける。
運用業務	運用業務	QA対応		インシデントのQA対応 ※分かりずらいもの ・E-ラーニングシステムの受講結果抽出依頼の作業
		保守対応	障害対応	インシデントの障害対応
			問題管理対応	問題管理対応
			予防保守	パッチ適用や、SPの適用のオーダー ・月次メンテの検証や、月次メンテ準備など
			その他保守	・手順書修正(リリース手順ではなく、手順書の雛形の修正など。) ただし、障害対応や、問題管理に関連する手順書の修正の場合は、それぞれのオーダーにつける。 ・インシデント外の調査工数(インシデントでも問題管理でもないような、技術調査などの工数など) ・引継ぎ作業 ・運用・保守業務の進捗管理 ・運用・保守業務の受注活動 ※小口や案件であふれた分については、それぞれのオーダーにつけて、その他保守にはつけない。 ※Project点検の資料作成、プロジェクト点検出席もそれぞれのオーダーにつけて、その他保守にはつけない。
		運転業務		・AMSの日次チェック、月次チェック ・昼ミ、昼ミ準備作業 ・健康管理月次作業調査作業 ※セキュリティ対策は、予防保守のオーダーとし、運転業務オーダーにはつけない。
		業務改善		運用保守のプロセス見直し等。運用管理者のオーダー

表1　運用保守業務の工数入力事例

3．3 運用保守業務の効率化のための改善

3．3．1 日々の進捗確認における無駄の排除

(1)進捗確認の内容見直し

毎日行っている運用保守業務の進捗確認の内容を見直し、1 回の進捗確認に約 15 分かかっていた進捗確認時間を約 10 分に短縮した。日々の進捗確認で確認する項目を追加していくうちに、確認項目が 13 個になっていた。内容を見直し、必要な4項目を確認することで、進捗確認の時間を短縮した。日々の進捗確認項目を見直した一覧を表2に示す。

No	件名	今後			
		いつ	確認者	確認事項	事前資料更新者
1	FJWANへのログイン確認	廃止			
2	就労の入力確認	廃止			
3	パスワード変更確認	廃止			
4	携帯電話ローテーションの確認	週ミに移動			
5	手順書規定類の作成改訂確認	週ミに移動			
6	一斉定時退社日の確認	週ミに移動			
7	小口メンテ残件数	なし	運用・保守管理者		自動
8	リリース実施予定	運用昼ミ	運用・保守担当者 ＋運用・保守管理者	本日のリリースの漏れがない事を確認する	リリース実施者
9	第1開発T担当の完了/中止以外の小口メンテの確認	運用昼ミ	運用・保守担当者 ＋運用・保守管理者	回送中のリリースの進捗状況を確認する	運用・保守担当者
10	完了分の工数管理	なし	運用・保守管理者		全員
11	リリース予定日より、3営業日以上経過しているリリースの確認	運用昼ミ	運用・保守担当者	リリース予定日より、3営業日以上経過しているリリースを確認する	自動
12	対応中インシデント	運用昼ミ	運用・保守担当者	回送中のインシデントの確認	自動
13	運用カレンダーで、作業予定及び、作業完了を確認する	週ミに移動			

表2 日々の進捗確認項目の見直し

(2)進捗確認をするメンバーの見直し

チームメンバー全員で行っていた運用保守業務の進捗確認を、運用保守管理者と運用保守担当者だけで行うようにした。以前はプロパー6 名とパートナー4 名の計 10 名で進捗確認を行っていた。しかし、チームメンバー全員で運用保守業務の進捗を確認したところで自分の作業以外は聞いていないため、チームメンバー全員で進捗確認をする必要性がないことが分かった。改善後は運用保守管理者 1 人と、運用保守担当者 2 人の合計 3 人で進捗確認を行うようにした。

(1)と(2)より、チーム全体で 1 日に 2.5 時間、1 ヶ月に 50 時間の工数を要していたが、1 日に 30 分、1 ヶ月に 10 時間の工数になり、1 ヶ月 40 時間の削減に繋がった。

3．3．2 繰り返し作業の協力会社要員への移管

作業手順が明確になっている運用保守業務については、手順書を見直して、標準化を行い、サービスセンターでも作業ができるようにした。作業依頼がきた際、運用保守管理者が依頼内容を確認し、サービスセンターへ移管した作業であれば、サービスセンターで作業を行うことができ、開発チームまでエスカレーションされることなく対応ができるようになったため、依頼に対して早く対応が出来るようになり、運用保守業務の工数も減らすことができた。

4．取り組みによる効果

3．1の運用保守業務全体の見直しより、運用保守体制と運用保守業務のフロー変更を行ったことで、指示系統が明確になり、作業依頼や QA に対しての対応を円滑進めることができるようにな

った。指示系統が明確になったことで、システム担当者が運用保守担当者へ勝手に作業を依頼させないようになったので、いつの間にか運用保守担当者の作業負荷が上がっていたという事がなくなった。

3．2の日々発生する運用保守業務で実施すべきことを明確化したことで、運用保守担当者はスムーズに業務ができるようになり、何の作業が滞ってしまうのかという課題が見え対策がたてやすくなった。

3．3の運用保守業務の効率化のための改善をしたことで、一つずつの運用保守業務の作業工数を短縮することができ、全体的な工数削減をすることができた。

2016 年度の開発チームの運用保守業務の工数の推移を図6に示す。

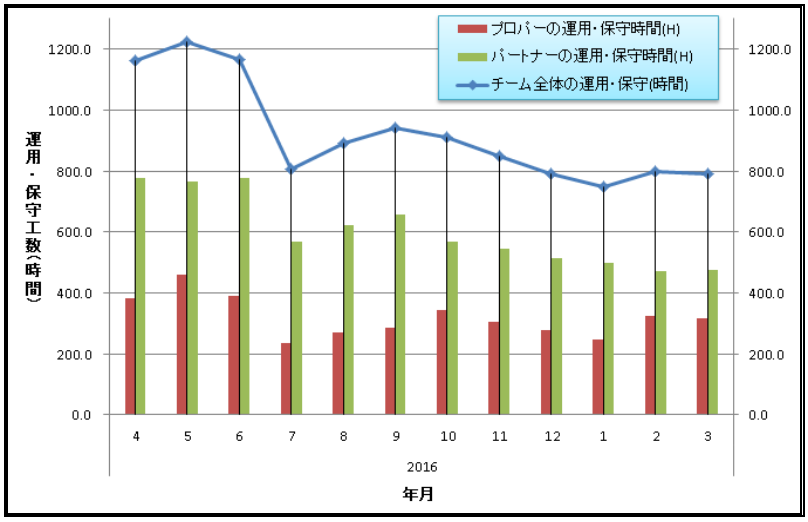

図6　2016 年度の開発チームの運用保守業務の作業工数

開発チームメンバーの総工数に占める運用保守工数の比率は、2016 年 4 月度の 58%から、2016 年 3 月度の 20%へ低下し、チームメンバーの運用保守工数を削減することができた。チームメンバーは開発案件に集中できるようになった。2016 年度の開発チームのプロパーの総工数に対する運用保守工数の割合を図7に示す。

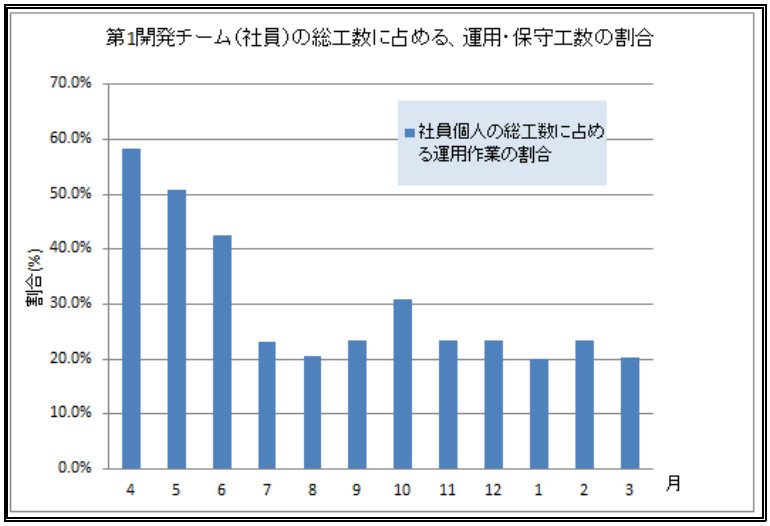

図7　2016 年度の開発チームのプロパーの総工数に対する運用保守工数の割合

運用保守業務の計画が立てられるようになり、利用者へいつ作業が対応できるか連絡できるよう

になった。作業の手戻りも減り、作業完了後に ZIFTEC に問い合せされることが減った。運用保守業務が標準化されたことで、運用保守担当者が変更になっても対応できるため、担当者変更や担当者不在によって作業が遅延することがなくなり、利用者へ迷惑をかけることがなくなった。

5. 今後の展望

開発チームメンバーの運用保守業務を減らすことができたが、今後はシステム担当者が運用保守業務に携わることが少なくなり、運用保守業務の状況や課題が分からなくなることが考えられる。そのため運用保守業務での状況を開発チームメンバーと共有することが課題になっている。この課題に対して、日々の運用保守業務から開発チームメンバーへ周知した方が良い情報をフィードバックするように取り組んでいる。例えば、システムの運用が不明確だったために、利用者との要件調整に手間がかかったものがあった。システム開発時に運用も見据えて仕様を決めるように、フィードバックした。また、リリースのミスによる手戻りが発生した場合は、運用保守担当者と運用保守管理者で対応して終わるのではなく、何が原因でどんな対策をすれば、ミスを防ぐことができるか、チームメンバーにそれぞれに考えてもらい、意見を交換する場を設けた。

6. おわりに

私自身、今まで開発プロジェクトに携わりながら自分の担当していたシステムの運用保守業務を行っていたので、チーム内で何がおきているか理解できていなかった。実際に運用保守管理者の立場で運用保守業務を見てみると、運用保守業務全体の見直しが必要であることが分かった。日々の運用保守業務がスムーズに回っていないこと、運用保守業務をする協力会社に的確な作業指示が出せていないこと、チームメンバーの不満など、細かい問題が多々あった。運用保守業務の中でもっと標準化できることや、無駄な作業を削除することができるので、今後も運用保守業務の改善作業を実施していきたい。

今回は第 1 開発チーム内での運用保守業務の効率化を行ったが、同様の状況は他チームでも発生している。他チームにもこの取組みを展開することができないか考えていきたい。

参考文献

［１］伊藤 正博:“ITIL で“ダメ運用”から抜け出す“, 日経システム構築 (2004.12), P.148

［２］アールワークス:“クラウド時代の正しいシステム運用”, 2012.5, 第1章～第2章

［３］“IT運用管理 攻めのツール活用術”, 日経SYSTEM, 第4章ITサービスマネージメント