

# 低コストで効果的なITIL運用管理支援システムの構築 に関して

とりごえ あきら  
鳥越 彰

株式会社エクサ エンタープライズ開発本部  
ITプロフェッショナル

原稿量  
本文 9,000字  
要約 900字  
図表 8枚

## <要約>

インシデント管理や変更管理といった ITIL ベースの運用管理を実践するには専用の IT サービスマネジメントソフトウェアを用いることが多い。一方、システムの運用管理に対するコスト削減の圧力は年々高まっており、trac や Redmine に代表されるオープンソースの課題管理ソフトウェアを運用管理に使用する事例も見受けられる。本論文では、低コストで効果的な ITIL ベースの運用管理ワークフローシステムを構築するための方法について論じている。

内容としてはまず、費用対効果に優れるツールを選定し、その上で効果的な運用管理システムを構築するための課題を検討した。ツールの選定にあたっては、最初に比較・評価のための評価ポイントを設定した。具体的には①プロセスの実装に関する柔軟性、②外部データ連携、③レポーティング、④Knowledge Base 連携、⑤コストである。次に評価、比較対象のツールとして Redmine, JIRA, LMIS を選んだ。Redmine はオープンソースの課題管理ソフトウェア、JIRA は商用の課題管理ソフトウェア、LMIS は ITIL 運用管理に特化した専用ソフトウェアという位置づけである。各ツールの比較検証を行う上で、①のプロセスの実装に関する柔軟性については、検証用のインシデント管理プロセスを設計し、そのプロセスを実装する過程で各ツールのカスタマイズの柔軟性や各ツールの持つ特徴を明らかにした。②外部データ連携、③レポーティング、④Knowledge Base 連携については、各ツールが備える個々の機能を調査し比較した。⑤のコストについてはソフトウェアのライセンス費用、またはクラウドの場合、ソフトウェア利用費を調べ比較した。最終的に各評価ポイントを総合的に判断し、最も費用対効果が高いと思われるツールを選定した。最後に選定したツールをベースに ITIL ベースの運用管理システムを構築するには何が課題でどのような機能を付加していく必要があるのかを検討した。

今回の検討は、実システム構築前の実現性検討作業であるが、3つのソフトウェアを用いて評価用のインシデント管理プロセスを実装することで、各ツールの機能や特徴を深く理解でき、結果的に ITIL 運用管理システムとして具備すべき機能の抽出が行えたと考える。

## 目次

低コストで効果的なITIL運用管理支援システムの構築に関して .....	1
1. はじめに .....	4
2. 検証用運用プロセス .....	4
3. 比較対象ソフトウェアと製品概要 .....	6
3.1 Redmine(レッドマイン) .....	6
3.2 JIRA(ジラ) .....	6
3.3 LMIS(エルミス) .....	6
4. 評価のポイント .....	7
4.1 プロセスの実装に関する柔軟性 .....	7
4.2 外部データ連携 .....	7
4.3 Knowledge Base連携 .....	7
4.4 レポーティング .....	8
4.5 コスト .....	8
5. 検証結果 .....	8
5.1 ワークフローのカスタマイズに関する各ツールの機能と特徴 .....	9
5.2 ワークフローのカスタマイズに関する考察 .....	13
5.3 外部データベース連携 .....	14
5.3 Knowledge Base連携 .....	14
5.4 レポーティング .....	15
5.5 コスト(ソフトウェア利用費) .....	15
5.6 比較結果のまとめと考察 .....	15
6. 低コストで効果的なITIL運用管理ツールの実現に向けて .....	16

## 1. はじめに

インシデント管理や変更管理といった ITIL ベースの運用管理のワークフローを実装するには専用の IT サービスマネジメントソフトウェアを用いることが多い。しかし一般的に IT サービスマネジメントの専用ソフトウェアは高額である。一方、システムの運用管理に対するコスト削減の圧力は年々高まっている。IT の運用管理コストは大きくモノに依存する部分とヒトに依存する部分からなる。通常、後者を削減すると運用サービスの品質の低下を招く。したがって前者、特にソフトウェアに関するライセンス費用や保守費用をどう削減するかという課題に行き着く。

その課題に対する動きとして最近、trac や Redmine に代表されるオープンソースの課題管理ソフトウェアを運用管理に使用する事例も見受けられる<sup>1)</sup>。しかし、無償のオープンソースソフトウェアで運用管理に求められる必要な機能が実現できるのであろうか？

今回、その疑問を解くために、まず検証用のインシデント管理プロセスを作成し、オープンソースソフトウェアを含むいくつかのツールに対し、そのプロセスを実装する過程で各ツールのカスタマイズの柔軟性や各ツールの持つ特徴を比較、検討した。

ツールの費用が安くても求める機能が実現できなくては意味がない。比較、検討の結果、最も低コストで必要な機能が実現可能なツールを選定した。

最後に選定したツールをベースに効果的な ITIL ベースの運用管理システムを構築するには何が課題でどのような機能を付加していく必要があるのかを検討した。

## 2. 検証用運用プロセス

今回、ツールの比較検証を行う上で、検証用の運用プロセスとしてインシデント管理プロセスを設計し、実際にそれぞれのツールを用いて実装することでツールの機能やカスタマイズの柔軟性を検証した。対象とするサンプルインシデント管理プロセスを図1に示す。

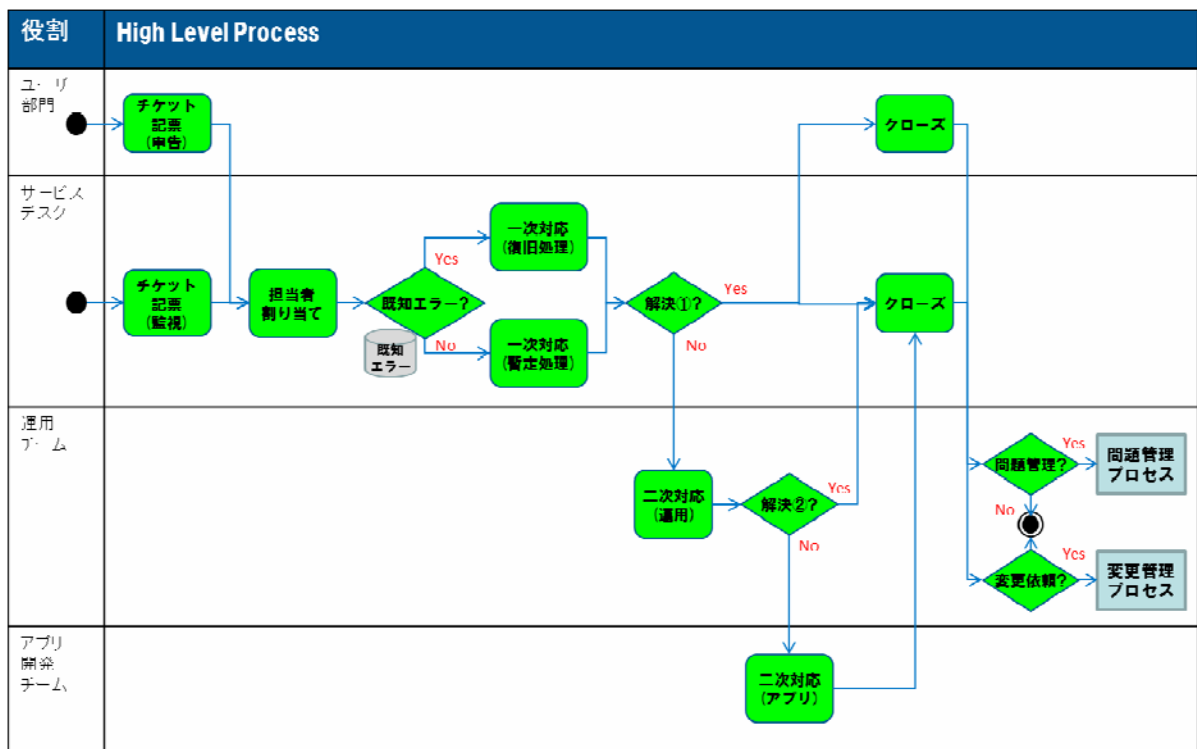


図1. 検証用サンプルインシデント管理プロセス

なお、図1に示す検証用サンプルインシデント管理プロセスでは、簡単にするため最初に起票されたチケットが「障害」か、「問い合わせ」か、「サービス要求」かについての判定と振り分けは省略し、すべて「障害」とであると仮定している。

更に、図1の検証用サンプルインシデントプロセスに対応したユーザーとグループ及び想定したロールと権限を表1に示す。

表1. ユーザー、グループ、ロールと権限

グループ	ユーザー	権限							ロール
		起票	担当者割当て	調査開始	エスカレーション	解決	コメント	クローズ	
ユーザー部門	A	○	-	-	-	-	○	○	ユーザーロール
サービスデスク	B	○	○	○	○	○	○	○	保守・運用ロール
運用チーム	C	○	○	○	○	○	○	○	
アプリ開発チーム	D	○	○	○	○	○	○	○	

### 3. 比較対象ソフトウェアと製品概要

比較対象ソフトウェアとして Redmine, JIRA, LMIS を選んだ。Redmine はオープンソースの課題管理ソフトウェア、JIRA は商用の課題管理ソフトウェア、LMIS は ITIL 運用管理に特化した専用ソフトウェアという位置づけである。以下、それぞれの製品概要を紹介する。

#### 3.1 Redmine(レッドマイン)

Web ベースでオープンソースの課題管理ソフトウェアである。主な使い道はバグトラッキングや課題管理、プロジェクト管理であり、Subversion や Git 等のバージョン管理ソフトウェアと連携可能である。ガントチャート、カレンダー、ロードマップ、ニュース、Wiki 等の機能を搭載している。同じ分類のソフトウェアとして trac がある。両社の違いとして、trac がほとんどの設定を構成ファイルの編集、またはコマンドラインで実施するのに対し、Redmine はすべての設定を GUI から変更可能である。また、Redmine は trac に比較してガントチャート等、標準で豊富な機能が備わっている。

#### 3.2 JIRA(ジラ)

豪州に本社を置くアトラシアン社が開発した商用の課題管理ソフトウェアである。Redmine と同じく主にバグトラッキングや課題管理、プロジェクト管理に用いられる。オープンソースプロジェクトや非営利組織に対しては JIRA を無償提供している。画面やフィールド、ワークフロー等の柔軟なカスタマイズ機能の特徴としている。100 か国 12,000 以上の顧客により使用されている。

#### 3.3 LMIS(エルミス)

日本の株式会社BSPが開発したITサービスマネジメントを支援する専用ソフトウェアである。ITIL v3 ベースのイベント管理、インシデント管理、問題管理、変更管理、リリース管理、構成管理、契約管理、FAQ 管理といった機能を実装している。クラウド版とオンプレミス版があり、クラウド版は Salesforce.com のプラットフォームを使用している。今回はクラウド版である LMIS on cloud を評価

対象とした。

## 4. 評価のポイント

評価にあたり、各ソフトウェアの長所、短所、特徴等を明らかにすべく、以下の評価ポイントを設定した。その他の検討ポイントとして大規模使用に対する拡張性やセキュリティ、監査への対応等あるが使用するミドルウェアやプラットフォームにも関係するのでここでは評価の対象外とする。

### 4.1 プロセスの実装に関する柔軟性

図1に示すインシデント管理プロセスをどれだけ忠実且つ簡単に実装できるかを検証する。具体的には以下の項目がある。

- ワークフロー機能
  - ◇ ワークフロー（状態と状態遷移の組合せ）自由に登録可能か？
  - ◇ 状態遷移の条件や事後操作を自由に記述可能か？
  - ◇ フローの分岐（複数の部署にレビューを依頼する等）は可能か？
- フィールド登録
  - ◇ カスタムフィールドを自由に追加可能か？
- 画面のカスタマイズ
  - ◇ 状態遷移の際に専用の画面を作成可能か？
  - ◇ ユーザーのロール毎に個々のフィールドを表示、非表示に設定することは可能か？
  - ◇ フィールドの表示順序を変更可能か？
- チケット間の関連付け
  - ◇ 間の関連付けは可能か？

### 4.2 外部データ連携

あらかじめ決められた選択肢からユーザーが適切な項目を選択する場合、選択肢の数が少なければツールのリスト型フィールドに選択肢を登録すれば良い。しかし、顧客データや構成管理アイテム等、数が多くてメンテナンスが必要なものは外部のデータソースと連携し値を参照する必要がある。ここでは容易に外部のデータソースと連携するための仕組みがあるかどうか調査する。

### 4.3 Knowledge Base連携

過去の障害の切り分け方法や解決方法に関する Knowhow、または障害の切り分けに使用するシステムやネットワークの構成図等を、運用のワークフローシステムからリンクで参照できるようになっていることが望ましい。ここではそれぞれのツールが Knowledge Base として運用に関する情報を登録、参照できる仕組みを備えているかを調査する。

#### 4.4 レポーティング

インシデント管理でツールに最低限必要なレポーティング機能はデータをエクスポートして Excel 等で障害の一覧を作成することである。次に蓄えたデータを時間軸で管理し、インシデントが持つ各属性、すなわち状態、サービス、障害発生箇所等のさまざまなデータを切り口に、一定期間内の個々の発生件数や比率をグラフ表示できる機能があることが望ましい。更に、レポーティングから一歩進んで SLA や KPI といった管理指標を設定しモニターする機能も欲しいが、こちらは要件が複雑且つ高度になるため今回の評価対象から外す。

#### 4.5 コスト

システムを構築し、運用するには一時費用として、ハードウェアやソフトウェア(ライセンス)の費用、及びシステムの開発費用が必要となる。また毎年定常的に発生する費用としては、ハードウェア、ソフトウェアの保守費用の他にシステムの運用費用、開発したシステムの維持管理費用等が発生する。クラウドのアプリケーションをサービスとして利用する場合はハードウェア、ソフトウェアに関する一時費用やそれらの保守費用は発生しないが、導入時の開発(カスタマイズ)費用及び利用する期間にわたりソフトウェア利用費がかかる。

ハードウェアに関する費用と、ソフトウェアの開発(カスタマイズ)、及び運用にかかる人件費は、どのソフトウェアを選択してもそれほど違いはないと仮定し、ここではソフトウェアのライセンス費用、またはクラウドの場合はソフトウェアの利用費を調査し比較する。

### 5. 検証結果

4章で設定した評価ポイントに関する評価結果を表2に示す。なお、表中の○は機能が確認できた場合、' - 'は確認できなかった場合、△は他の製品との連携で機能が実現できる場合を表す。



表2. 評価結果

大項目	中項目	製品		
		Redmine (2.3)	JIRA (6.0.8)	LMIS (1.2)
ワークフローの カスタマイズ	ワークフローの登録	○	○	— 注1)
	状態遷移の条件設定、事後操作登録	—	○ 注2)	○
	フローの分岐（同時並行処理）に対応可能	—	○	○
	カスタムフィールドの登録	○	○	○
	状態遷移の際に専用の画面を作成可能か？	—	○	○
	ロールと状態の違いによるフィールドの表示、非表示	○	○	○
	フィールドの表示順序変更	—	○	○
	ITILの運用プロセス（インシデント管理や変更管理等）のテンプレート（画面、ワークフロー）が用意されている	—	—	○
	チケットの関連付け（親子、Block、関連等）が可能	○	○	○
外部データ連携	構成管理情報を格納した外部データベースとの連携を容易にする仕組みがあるか？	○ API	○ APIとSQL	○ APIと File(csv)
Knowledge Base 連携	関係者が運用に関する各種 Knowhow を登録し共有する Wiki 等の掲示板と連携	○ Wiki 注3)	△ Confluence 注3)	○ FAQ 管理 注3)
レポートニング	蓄えられたチケット情報のさまざまなデータフィールドを切り口にレポートを生成するための機能またはレポート作成を補助する仕組みがあるか？	—	○ 10種類程度	○ 数10種類以上

注1) LMIS の場合は ITILv3 の運用管理プロセスが実装されており、そのプロセスを必要に応じてカスタマイズする形になる。

注2) JIRA はフローの分岐点でサブプロセスを生成することにより対応可能。

注3) Redmine は Wiki が製品に組み込まれている。JIRA は Confluence という同じ Atlassian 社の Wiki に似たツールを組み合わせ使用。LMIS は FAQ 管理機能がある。

### 5.1 ワークフローのカスタマイズに関する各ツールの機能と特徴

Redmine はワークフローの実装に関して、状態の追加やユーザーのロール毎に遷移先を制限する機能、及びフィールドの追加といった基本的なカスタマイズ機能はある。しかし画面やメニュー等を自由に配置する機能はなく、状態遷移の条件設定や遷移後の事後操作を登録することはできない。ただし、これらがなくてもチケットの編集機能を使って担当者を変更、あるいは、状態を変更することでワークフローのように使うことはできる。

Redmine のワークフロー登録画面を図2に示す。



図 2. Redmine のワークフロー登録画面

また、ユーザーのロールとチケットの状態により、各フィールドを表示するかしないか（Redmine 上では「読み取り専用」と呼ぶ）の設定が可能なので、状態毎に関係のないフィールドを見せなくすることは可能。ただし、カスタムフィールドに「必須」フラグを立てると「読み取り専用」とは関係なく起票時から入力必須になる。

次に JIRA はワークフローの実装に関しては、画面やフィールド、状態遷移のアクションの登録等、非常に柔軟に設定が可能である。特に状態遷移毎に必要なフィールドのみで構成した画面を作成、あるいは、遷移の条件や事後操作も細かく指定できるためカスタマイズの自由度はかなり高い。表 3 に今回 JIRA で実装した検証用インシデント管理システムの状態遷移と画面、フィールドを示す。

表 3. JIRA で実装した検証用インシデント管理システムの状態遷移と画面、フィールド

インシデント管理プロセス							
アクション	チケット 起票	担当者 割当て	調査開始	エスカレー ション	解決	クローズ	
状態	New	Assigned	In Progress	In Progress (Ope or App)		Solved	Close
画面	インシデ ント登録画面	担当者割 当て画面	対応開始 画面	担当者割 当て画面	対応結果 登録画面	クローズ 画面	
入力フィー ルド	課題タイプ	担当者	重要度	担当者	担当者	コメント	
	要約		KB 有無		対処方法		
	説明				対処内容		
	添付 File				添付 File		
	キーワード				解決状況		

表 3 の中で示される JIRA で作成した対応結果登録画面の例を図 3 に示す。

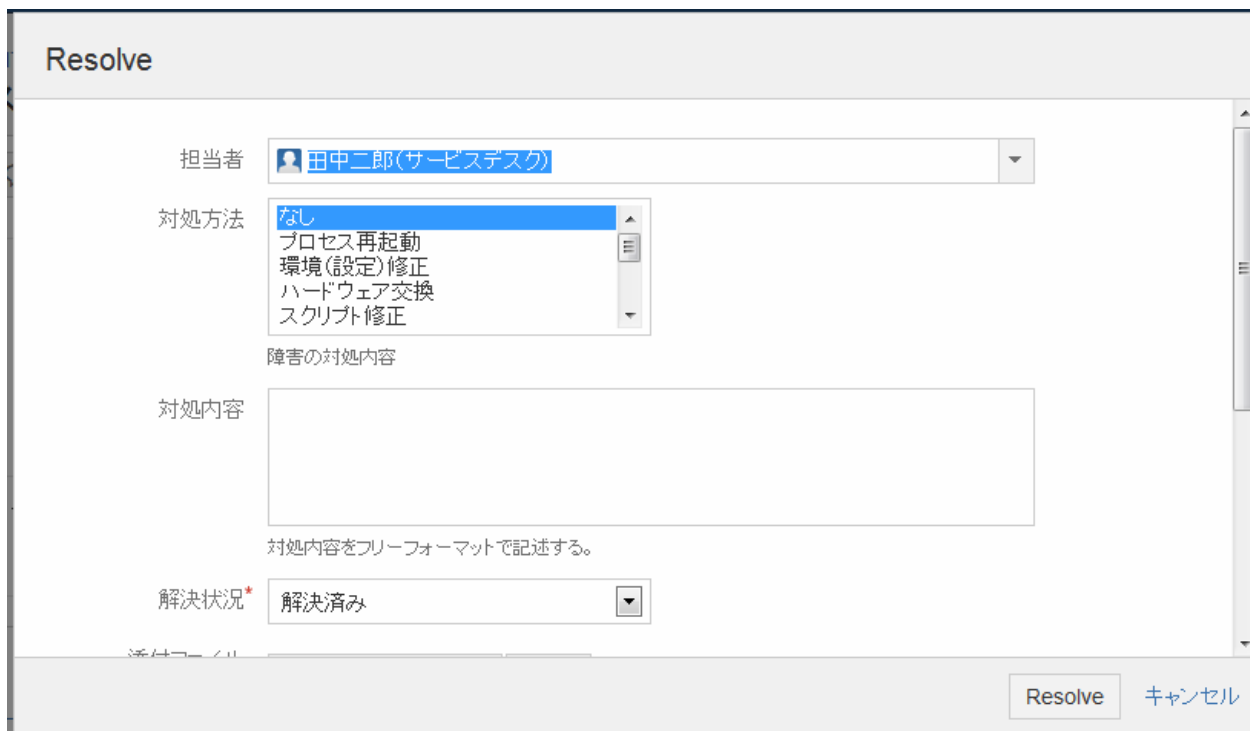


図 3. JIRA で作成した対応結果登録画面

また、ワークフローの処理中に、現在フローのどの段階であるかについて、ワークフローのフロー図がグラフィカルに表示され、現在の状態が色分けで示される機能があるので便利である。JIRA で作成したワークフローのフロー図を図 4 に示す。

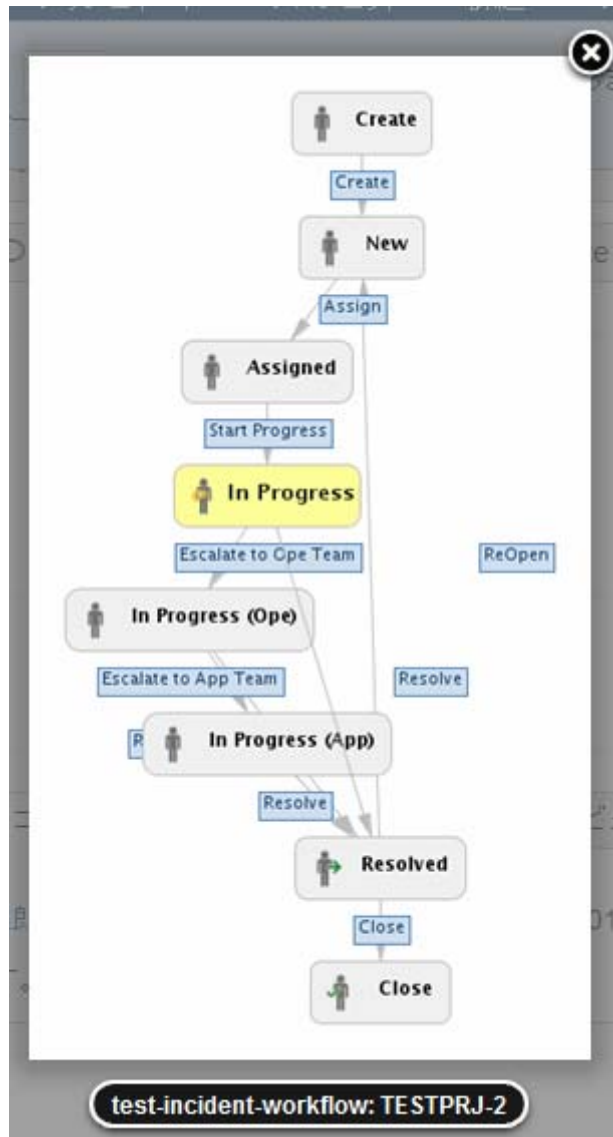


図4. JIRA で作成したワークフローのフロー図

最後に LMIS であるが、LMIS on cloud に標準で実装されているプロセスと今回実装するインシデント管理プロセスとを比較すると状態遷移時の承認（LMIS は毎回承認が必要）や運用チーム、アプリ開発チームへのエスカレーション等（LMIS のデフォルトはそのようなチームがない）で異なる。LMIS で検証用サンプルインシデント管理プロセスを実装しようとしたが、ユーザーやロール、及びワークフローを編集するにはシステム管理者の権限アカウントが必要であり、通常、評価用にはシステム管理者のアカウントは貸し出していないため、こちらで想定していたプロセスを実装することはできなかった。したがって、ワークフローのカスタマイズ機能の評価はマニュアルを参照して判断した。参考までに LMIS のインシデント管理画面を図5に示す。



図 5. LMIS のインシデント管理画面

なお、LMIS では状態遷移を行うアクションは図 5 中にある「調査開始」や「承認依頼」の各ボタンを押すことでステータスが変わる。その結果、その下の「開始」、「調査」といった大きなアイコンの色が変わって現在の状態を示す仕様になっている。なお、内容の更新は常に「編集」ボタンから行う。

状態の遷移に関して LMIS に特徴的な事項は、フローの状態を遷移するには基本的に承認者の承認が必要であることである。デフォルトでは作業を実施した人の上司が承認者にアサインされるが、承認者の決定方法は他のオプションもある。

## 5.2 ワークフローのカスタマイズに関する考察

Redmine は本来の使用用途である課題管理やプロジェクト管理にはこれでじゅうぶんと思うが、ITIL の運用プロセスの管理を想定した場合のワークフローシステムとしては使いづらい。この理由の 1 つとしては Redmine のワークフローの状態は、あくまでチケットの 1 つのフィールドにすぎず、状

態を遷移させてワークフローを回していると感じさせる仕掛けがないからである。

JIRAはその点非常に柔軟性に富むため、思ったとおりのワークフローを実装できる。一方で、JIRAはLMISのようなITIL運用管理の専用ソフトウェアと違い、Redmine同様、課題管理のプラットフォームソフトウェアなので、ITILの運用プロセスはすべて一から実装する必要がある。

LMISはITIL運用管理の専用ソフトウェアであるため、デフォルトで実装されているワークフローで良ければユーザーやサービス、構成管理アイテム等の必要な初期データを投入すればすぐにITILの運用管理ツールとして使用することができる。また、各プロセス間のチケットの連携がスムーズで、イベント管理のチケットを元にインシデントの生成、インシデントのチケットを元に問題管理チケットの生成といった具合に新たに登録画面から入力することなく必要な情報が受け継がれて連携プロセスのデータが登録される。また、それらの関連を簡単に追えるので使い勝手が良い。一方、データの更新には共通の編集画面を用いるため、起票時からすべての属性が表示される。一見情報量過多でエンドユーザーには使いづらいと思うが、フィールドのカテゴリによって基本情報(障害の内容)、種別情報(障害 or 問い合わせ or サービス要求等の種別、優先度等)、調査情報、対策情報等のカテゴリにグルーピングされているので慣れれば問題ないかもしれない。しかし、LMISは実装されているITILの標準的なプロセスが基本にあり、承認フローやフィールドの追加等のカスタマイズは可能だが、独自のプロセスを一から作るという場合は向いていない。

### 5.3 外部データベース連携

運用ツールのフィールドの中にはあらかじめ登録された選択肢のリストから選ぶ場合がある。

選択肢の数が多くなければ、RedmineもJIRAもLMISもリスト形式のカスタムフィールドを登録可能である。一方、顧客データや構成管理データ等、選択肢の数が多く外部データベースの値を参照する必要がある場合は外部データベースと連携する仕組みが必要になる。

JIRAにはDatabase Custom FieldやJIRA Database Values Plugin等の無料のプラグインが存在し、データベースの接続情報を設定し、取得のためのQueryをSQLで記述すれば、JIRA上のカスタムフィールドに外部データベースのデータ連携が可能となる。

RedmineはInternetで検索した限りではそのようなプラグインは見つからなかった。APIが公開されているのでプログラミングを前提で外部システムとの連携は可能である。LMISはcsv形式のファイルでデータを外部から取り込むことが可能である。ファイルの監視機能もあるので定期的にデータを取り込むことも可能である。

### 5.3 Knowledge Base連携

RedmineはWikiが標準で提供されるため、ここに運用の各種Knowhowを体系的に蓄積すれば、インシデント管理等のチケットからWiki上の情報をURLでリンクを張ることでKnowledge Baseの連携が可能となる。同様な仕組みがJIRAにもあるが、こちらは同じAtlassian社のConfluenceという製品

を別途購入する必要がある。LMIS は FAQ 管理機能があるのでここにユーザー用の FAQ、及び運用サービス提供側の Knowhow を蓄積し参照することが可能である。

#### 5.4 レポーティング

レポートの生成に関してはインシデントの情報を csv 形式で外部出力する機能は Redmine, JIRA, LMIS とともにサポートしている。障害の分類や障害原因等のインシデントの属性を元に傾向分析をグラフ等でレポート出力する機能は Redmine にはない。

JIRA はこの点、デフォルトで 10 種類のレポートを生成可能である。また LMIS は構成管理データベースが実装されているため、サービス別や管理対象別等のさまざまな切り口について数十種類以上のレポートを生成可能である。

#### 5.5 コスト(ソフトウェア利用費)

ソフトウェアを利用する上で必要となるライセンス費用 (クラウドの場合はソフトウェア利用費) を調査した。以下はネット等で公開されている定価の情報である。まず Redmine はオープンソフトウェアのためライセンス費用はかからない。次に JIRA は 25 ユーザーの場合、初年度のライセンス費用が 16.8 万円、2 年目以降の年間保守費用が 8.4 万円である<sup>2)</sup>。一方、LMIS のクラウド版は 25 ユーザーで初期登録費用 30 万円、月額利用料 10 万円<sup>3)</sup>になる。これは年額に換算すると初年度合計で 150 万円である。オンプレミスとクラウドの違いはあるが、LMIS は JIRA の約 9 倍のコストがかかる計算になる。なお、JIRA は 10 ユーザーの場合、スターターパックというライセンスがあり、初年度のライセンス費用が 1000 円、2 年目以降の年間保守費用も 1000 円<sup>2)</sup>と破格に安い。機能は変わらないので小規模のシステム用には非常にお得である。

#### 5.6 比較結果のまとめと考察

ITIL ベースの運用管理を実践するツールとして機能面の充実度が最も優れているものは LMIS である。他の 2 つに比べ ITIL ベースの標準的な運用管理プロセスが実装済みであること、また、クラウドベースの Web アプリケーションなのでユーザーカウントの登録等の初期設定を行えばそのまま使えるという利点がある。

しかし、LMIS は他の製品に比較してソフトウェアの利用費が高い。LMIS は ITIL の運用管理プロセスが一通り実装されているため、Redmine や JIRA に比べて運用プロセスの実装 (カスタマイズ) にかかる費用は少ないと考える。しかし、システムの運用プロセスは各社や各システムでさまざまであり、そのまま使えるというケースは少ない。そういう意味では LMIS を用いても運用プロセスの実装にかかる費用は発生する。今回のテーマがいかに費用を抑えて運用管理システムを構築するかであるので LMIS の高額なソフトウェア利用費は不利になる。ではライセンス費用の一番安い Redmine はというと、4 章で比較検討したように ITIL ベースの運用管理に適用する場合、カスタマイズの柔軟性に難がある。一方、JIRA はカスタマイズの柔軟性に富み、想定した画面の遷移を自由に作れるという利点が大い。また、外部データベースの連携も簡単に実現できるアドオンパッケージが無料で利用できるし、レポートングに関しても Web で参照可能な基本的なレポートは用意されている。個別に込み入ったレポ

ートを作成するにはデータを csv でダウンロード後、Excel で作成すれば良い。

以上から、低コストでユーザーの個々の要件に適合可能なシステムを構築するには、JIRA を用いて開発するのがベストと考える。

## 6. 低コストで効果的なITIL運用管理ツールの実現に向けて

3つのツールを調査し比較検討した結果、コストを抑えて ITIL 運用管理のワークフローシステム構築するためには JIRA をベースに運用管理システムを開発するのが適しているとの結論に達した。

今回、インシデント管理プロセスのみを実装したが、必要に応じて残りの問題管理や変更管理といったプロセスも実装する必要がある。今回、JIRA でサンプルインシデント管理プロセスを実装するのにかかった工数はマニュアルを見ながらではあるが3週間程度である(注：簡易な設計書の作成と簡単な動作確認を含む)。状態の遷移毎に画面を作成しているにもかかわらず、この程度の工数で完成できた理由は、基本的にノンプログラミングで開発できるからである。もともと運用プロセスは、それぞれの企業や対象とするシステム、関係する組織に応じてさまざまに変化するものである。したがってどのような運用管理システムを作るにせよ、運用設計は必ず必要であるし、その結果に応じたプロセスを実装する必要がある。

次に外部システムとの連携であるが、構成管理 DB のデータを参照する場合、前述の外部データベースとのアドオンツールを使用すれば比較的容易に連携可能である。また、対象が新規に開発、導入されるシステムの運用管理であるならば、JIRA でデータを保管しているデータベース（通常は Postgres や MySQL 等）上に新たに構成管理 DB を作成するのが効率的である。

更に検討すべき課題として、SLA(Service Level Agreement)や、サービスを提供する側のパフォーマンスを測る KPI(Key Performance Indicator)の扱いがある。例えばサービスデスクの KPI としては以下が考えられる。

1. サービスデスクで解決できたインシデントの数
2. 運用チームにエスカレートして解決できたインシデントの数
3. 起票後、担当者がアサインされるまでに要した時間
4. 解決までに要した平均時間（重要度別 及び 全体）

上記は一定期間内に発生したインシデントに対しての集計機能であるが、継続的なサービス品質の向上のために、このような KPI を算出できるような機能を盛り込むことが望ましい。

以上、低コストで効果的な運用管理ツールの開発の方向性を示した。今後はこの方向性に沿って運用管理ツールを開発し、実際のプロジェクトでその効果を検証したい。



参考文献

- 1) NTTデータ、安定かつ高品質なサービス提供を支援するITSM支援ツールの取組 2012年4月発行、NTT技術ジャーナル 参考URL: <http://www.ntt.co.jp/journal/1204/files/jn201204050.pdf>
- 2) Rick Software JIRA 価格 URL:  
<https://www.ricksoft.jp/product/atlassian/jira>
- 3) (株)BSP LMIS on cloud 価格 URL:  
[http://www.bsp.jp/products/lmis/oncloud\\_price.html](http://www.bsp.jp/products/lmis/oncloud_price.html)

---

Redmine は Jean-Philippe Lang 氏と Redmine の開発に賛同する Contributor が開発したオープンソース・ソフトウェアである。

JIRA は豪州に本社を持つ Atlassian 社の商標、または登録商標である。

LMIS は(株)BSP の商標、または登録商標である。

その他の会社名並びに製品名は、各社の商標、若しくは登録商標である。