
プロジェクト管理ソフトウェアを活用した 効率的なプロジェクト運営についての考察

株式会社 北見コンピューター・ビジネス 札幌支店

■ 執筆者Profile ■



富田 英治

2006年 (株)北見コンピューター・ビジネス入社
札幌支店システム部 配属

2009年 現在 札幌支店システム部
主任プロジェクトリーダー



藤澤 一人

2003年 (株)北見コンピューター・ビジネス入社
システム部 配属

2009年 現在 札幌支店システム部
主任プロジェクトリーダー



伊藤 加奈

2008年 (株)北見コンピューター・ビジネス入社
札幌支店システム部 配属

2009年 現在に至る

■ 論文要旨 ■

大規模プロジェクト・小規模プロジェクトの輻輳など、システム開発の現場での最大の悩みは、人的、時間的な資源の効率的な管理である。世には様々なプロジェクト管理ソフトウェアが出回っている中で、オープンソースソフトウェアのプロジェクト管理ソフトウェアを導入し、導入コストを抑えた上で現場の問題点がどのように解決できるか、その可能性や効果について、プロジェクト管理ソフトウェアの機能の紹介を織り交ぜながら考察する。

■ 論文目次 ■

<u>1. はじめに</u>	《 5》
1. 1 当社の概要	
1. 2 プロジェクト管理ソフトウェアが必要となった背景	
<u>2. システム開発における問題点と改善案</u>	《 6》
2. 1 現状の問題点と改善案の方向性	
2. 2 プロジェクト管理ソフトウェアの選定	
2. 2. 1 プロジェクト管理ソフトウェアの調査	
2. 2. 2 Redmineの特徴	
2. 2. 2. 1 Redmineとは	
2. 2. 2. 2 動作環境	
2. 2. 2. 3 多言語対応	
2. 2. 2. 4 Webベースの管理	
2. 2. 2. 5 複数プロジェクトのサポート	
2. 2. 2. 6 プロジェクトの管理	
<u>3. プロジェクト管理ソフトウェアの導入・運用</u>	《 9》
3. 1 Redmine の導入環境	
3. 2 評価用顧客, プロジェクトの作成	
3. 3 タスクのチケット管理	
3. 4 プロジェクトメンバー間の情報共有	
<u>4. プロジェクト管理ソフトウェアの導入効果</u>	《 15》
4. 1 Redmine の導入効果	
<u>5. 今後の課題</u>	《 16》
<u>6. おわりに</u>	《 17》

■ 図表一覧 ■

表 1	インストールを行ったハードウェア構成	《 9》
表 2	インストールを行ったソフトウェア	《 9》
表 3	顧客・プロジェクト管理イメージ	《 9》
図 1	チケット作成イメージ	《 10》
図 2	チケット管理イメージ	《 10》
図 3	配信メールイメージ	《 11》
図 4	チケットの時系列管理イメージ	《 12》
図 5	関連チケットの管理イメージ	《 13》
図 6	フォーラムイメージ	《 13》
図 7	フォーラム経過イメージ	《 14》

1. はじめに

1. 1 当社の概要

株式会社北見コンピューター・ビジネス（以下、当社）は、関連会社のシステム開発業務を担うため、平成9年5月に北海道北見市に本社を設立し、平成18年4月に北海道札幌市に支店を開設、従業員数27名（平成21年8月時点）の会社である。

当社では、システム開発をはじめ、パッケージソフト及びハードウェアの販売、ネットワーク事業、テクニカルサポート、システム運用保守などトータルサービスを展開し、富士通パートナー認定及びマイクロソフトパートナー認定を受け、様々な業種のユーザーへそのサービスを提供している。

また、平成15年より顧客満足度向上を目指したCS活動の取り組みや、個人情報保護の観点から、個人情報保護方針を定め、平成17年8月に日本情報処理開発協会(JIPDEC)より「プライバシーマーク」を取得し、情報処理サービス業としてのコンプライアンス経営にも取り組んでいる。

1. 2 プロジェクト管理ソフトウェアが必要となった背景

当社の現状ではプロジェクトごとに管理資料を“それなりに”作成している。“それなりに”当社の基準でドキュメントを作成し、プロジェクト単位に保存しているはずであるが、所在が不明となってしまったドキュメントもあれば、存在しても古いものだったりする。ある一つ的设计書をとってみても、改版履歴を辿っても変更するに至った経緯の一つ一つまで掘り起こすのが非常に困難な文書や、様々な図表での経緯をたぐっていった、ようやく結論に行き着くような文書もある。

昨今当社では、本社・札幌支店間での拠点をまたがる開発も増えてきている。更に他社と共同で行うプロジェクトも増えていくものと予想されている。規模が大きくなるプロジェクトにはプロジェクトを管理するソフトウェアが必要不可欠。単純な手管理によるプロジェクトマネジメントから脱却しなければいけない、と実感した次第である。

2. システム開発における問題点と改善案

2. 1 現状の問題点と改善案の方向性

まず当社のシステム開発の現場を振り返り、問題点の洗い出しを行った。普段はあまり気にしないで「あたりまえ」と思ってやってきた部分もあったが、改めて「これが問題点」というものを列挙してみた。

- (1) お客様と E メールで情報交換を行う場合、メール本文の情報は個人管理になってしまうため、当該業務以外の他のメールと混在してしまい、必要な情報の見落としが起りやすく、情報を掘り起こすのに労力を要している。
- (2) 結果の管理のみに重点を置いてきたため、あとから結果に至るまでのプロセスを図表を交えて時系列化する際に労力を要する。
- (3) 一つのプロジェクトの中では情報は整理されているように見えるが、他のプロジェクトと歩調をあわせた管理を行えていないし、部単位、支店単位などの大きな括りでプロジェクトの総括ができない。
- (4) 個人の進捗を管理する際に、一つのファイルで管理していると他の人が更新中には自分は更新できないことがあり、リアルタイムに反映されない場合がある。
- (5) 他のプロジェクトで蓄積した情報を、プロジェクトを跨いだ類似事例の検索に労力を要する。
- (6) 蓄積した情報量が増えるにつれ、必要な情報を取り出すことに時間が掛かる。

上記の問題点が克服でき、かつ、更に作業効率の向上が見込めるようなツールはないだろうか？

この問題提起が出発点となり、以下に示すような機能を含んだプロジェクト管理ソフトウェアが“無償で”調達、使用できることが望ましいとの結論にいたり、早速事例の調査を開始した。

- (1) 関係情報の共有、検索を容易にし、プロジェクトを跨いだ事例検索機能を有する。
- (2) 結果、プロセスを図表を交えながら時系列で管理することを可能とする。
- (3) 各プロジェクトに統一したユーザーインターフェースを提供する。
- (4) リアルタイムに更新することが可能となる Web ベースでの入力インターフェースを有する。

2. 2 プロジェクト管理ソフトウェアの選定

2. 2. 1 プロジェクト管理ソフトウェアの調査

まずはじめに有償無償を問わずに、プロジェクト管理ソフトウェアにはどのようなものがあるかを調査した。

(1) Trac(<http://trac.edgewall.org/>)

Python で開発されている Web ベースのプロジェクト管理ソフトウェアであり、オープンソースである。また、Wiki^(※1) や BTS^(※2) としても利用可能である。

(2) Redmine(<http://www.redmine.org/>)

Ruby on Rails で開発されている Web ベースのプロジェクト管理ソフトウェアであり、オープンソースである。また、Wiki^(※1) や BTS^(※2) としても利用可能である。

(3) dotProject(<http://www.dotproject.net/>)

PHP で開発されている Web ベースのプロジェクト管理ソフトウェアであり、オープンソースである。

(4) @task(<http://www.proside.co.jp/attask/>)

SaaS(Software as a Service)形式で提供される Web ベースのプロジェクト管理ソフトウェアであり、導入形態、ライセンス形態により有償である。

(5) GanttProject(<http://www.ganttproject.biz/>)

Java ベースで開発されているプロジェクト管理ソフトウェアであり、オープンソースである。

(6) Microsoft Office Project(<http://office.microsoft.com/ja-jp/project/>)

Microsoft® 製のプロジェクト管理ソフトウェアであり、各 Edition により価格が異なる。

他にも様々なプロジェクト管理ソフトウェアが存在するが、検討の結果、今回はオープンソースの Web アプリケーションフレームワークである Ruby on Rails で開発されている Redmine を選択し、考察を行う。

(※1) Web ブラウザから簡単に Web ページの発行・編集などが行える、Web コンテンツ管理システム。

(※2) Bug Tracking System の略。プロジェクトのバグを登録し、修正状況を追跡するシステム。

2. 2. 2 Redmine の特徴

2. 2. 2. 1 Redmine とは

先述のとおり、Redmine は Ruby on Rails という Web アプリケーションで開発されている。そもそも Ruby on Rails は Ruby という比較的新しい言語で実装されている。Ruby とは 1993 年にまつもとゆきひろ氏により開発されたオブジェクト指向スクリプト言語である。

2. 2. 2. 2 動作環境

Windows, Linux いずれの環境にも構築することができ、Ruby, Ruby on Rails, MySQL の環境さえあれば動作することが可能である。Web ベースであるため、クライアント端末にはインターネットブラウザさえあれば使用することが可能となる。

2. 2. 2. 3 多言語対応

日本語をはじめ 20 以上の言語に対応しているため、システムで規定の言語を設定できる他、ユーザーごとに利用する言語を設定することができる。
早速当社の Web サーバにインストールを行う。なお、インストール手順については割愛する。

2. 2. 2. 4 Web ベースの管理

ほとんどの設定・管理を Web ベースで行うことができ、それにより従来の表計算ソフトなどで発生していた管理の煩雑さや排他ロックが発生しにくい。

2. 2. 2. 5 複数プロジェクトのサポート

複数のプロジェクトを管理することが可能であり、複数のプロジェクトに属しているユーザーの場合、ログインする場所は一箇所が良い。プロジェクト間でのチケットの移動、関連付けも可能である。また、プロジェクトごとに Wiki やフォーラムを設置することができる。

2. 2. 2. 6 プロジェクトの管理

新規機能の開発、バグ修正、顧客サポートなど、様々なタスクをチケットとして管理でき、各チケットの状況を一元的に管理することができる。また、チケットに設定した作業開始日、期限日、進捗率から自動的にガントチャートやカレンダーが生成され、チケットに作業時間を記録することで、プロジェクトごと、ユーザーごとなど様々な切り口で集計することができる。

3. プロジェクト管理ソフトウェアの導入・運用

3.1 Redmine の導入環境

当社では以下の環境にプロジェクト管理ソフトウェアをインストールした。インストールを行ったハードウェアの構成については表1，ソフトウェアについては表2に示す。なお，インストール手順，初期設定については割愛する。

OS	Microsoft Windows Server 2003 Standard Edition Service Pack 2
メモリ	1GB
CPU	Xeon™ CPU 2.80GHz

表1. インストールを行ったハードウェア構成

Redmine	Redmine 0.8.0
Web サーバ	Apatche 2.2
データベース	Mysql 5.0.67x
Ruby	Ruby 1.8.6
Ruby on Rails	Ruby on Rails 2.2.2

表2. インストールを行ったソフトウェア

3.2 評価用顧客，プロジェクトの作成

今回は，テスト用の顧客を作成し，テストプロジェクトを作成する。テストユーザーをプロジェクト責任者とする。同様にテスト用の顧客を別に作成し，テストプロジェクトを作成する。

顧客・プロジェクトの管理イメージはファイルシステムでのフォルダ構成と同様になる。管理イメージを表3に示す。

顧客名	プロジェクト名
テスト顧客	テストシステム1
	テストシステム2
テスト顧客2	テスト管理システム

表3. 顧客・プロジェクト管理イメージ

3. 3 タスクのチケット管理

各工程で発生するタスクを、チケットとして管理する。チケットの作成・管理イメージをそれぞれ図1、図2に示す。

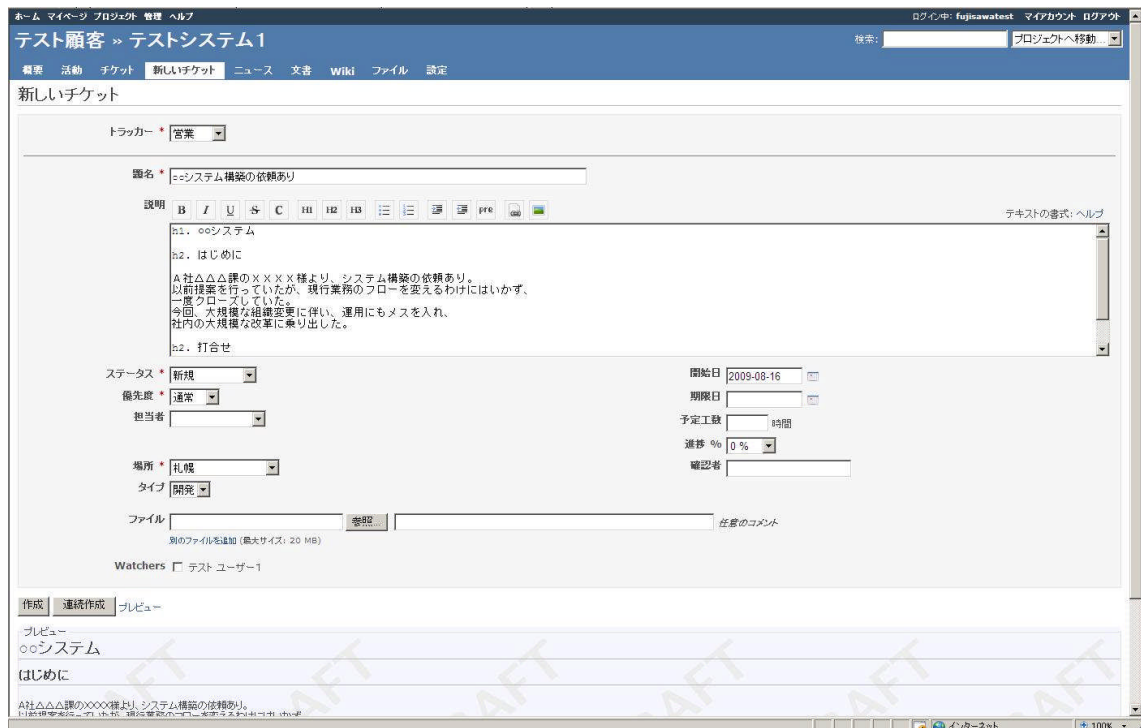


図1. チケット作成イメージ



図2. チケット管理イメージ

チケットを作成した際、プロジェクトメンバーに自動的にメールが配信される。それによりプロジェクトメンバーへ周知する際のコストの低減が可能となる。配信されるメールのイメージを図3に示す。

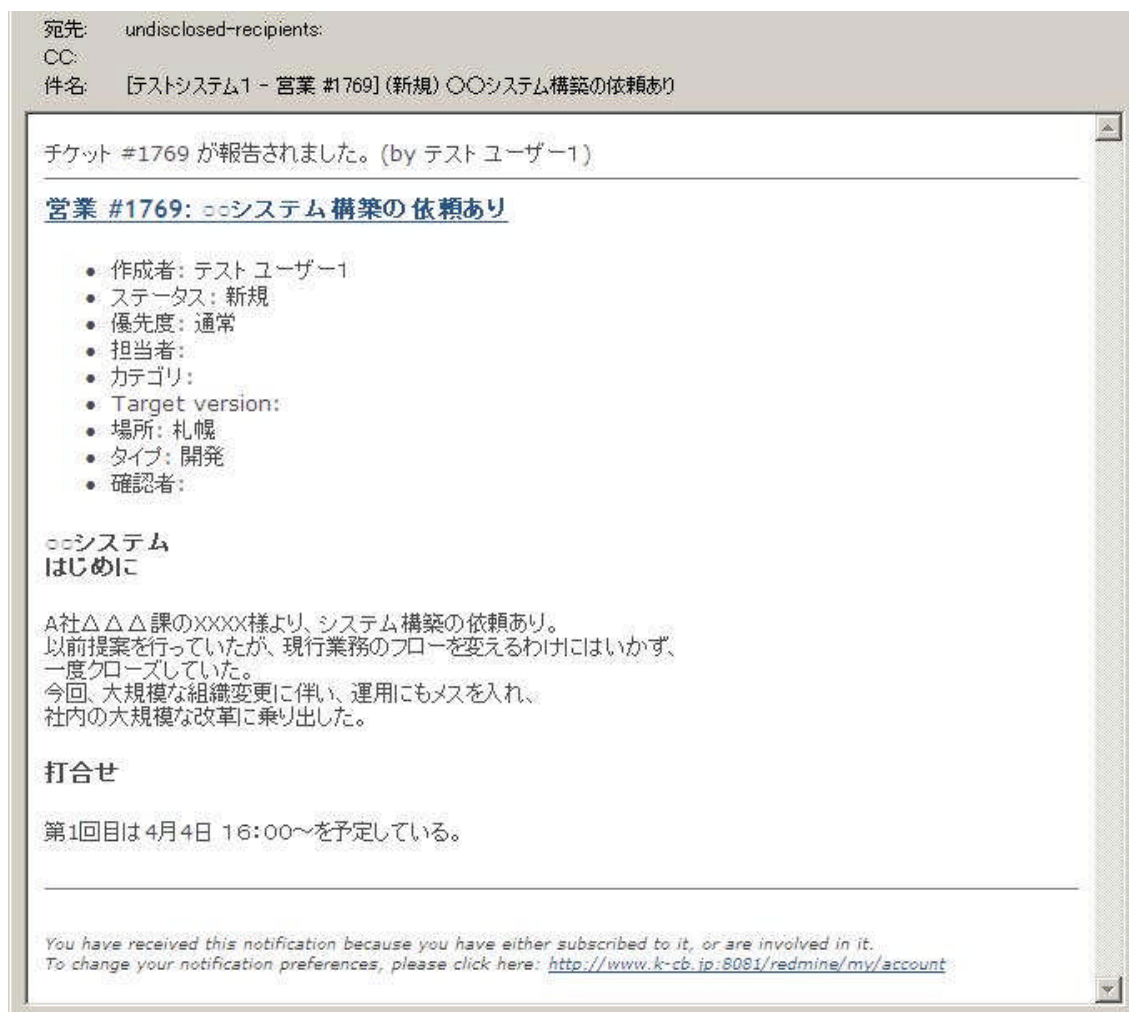


図3. 配信メールイメージ

チケットには、発生順に内容を追加・変更することができ、時系列に管理することができる。チケットの内容を充実させておけば、途中で担当者の変更があった場合にも、内容を整理して柔軟に引継ぎを行えるメリットがあると考えられる。チケットの時系列管理イメージを図4に示す。

The screenshot shows a Redmine interface for a project named 'テスト顧客 >> テストシステム1'. The main content area displays a ticket with the title '○○システム構築の依頼あり' and a status of '新規' (New). The ticket is assigned to 'テスト ユーザー1' and has a due date of '2009/04/01'. The progress bar shows 0% completion. The ticket description includes a section titled '○○システム' and 'はじめに' (Introduction), which mentions a system architecture dependency and a request for a new version. The ticket history shows three updates: 1) A file '業務フロー-081001.xls' was added. 2) The due date was changed from '2009/08/16' to '2009/04/01'. 3) A note was added stating that the user is a regular user and the ticket is a request for a new version. The interface also includes a sidebar with navigation options like 'チケット' (Tickets), '計画' (Plans), and 'カスタムウエリ' (Custom Widgets).

図4. チケットの時系列管理イメージ

また、それぞれのチケットには関連するチケットを持たせることができ、チケット間の参照を補助することができる。それにより、プロジェクト内の類似チケットを視覚的に管理することができる。関連チケットの管理イメージを図5に示す。



図5. 関連チケットの管理イメージ

3.4 プロジェクトメンバー間の情報共有

フォーラム機能を使用することで、情報交換や議論を行うこともできる。それにより議論結果だけでなく、経過も記録に残すことができる。フォーラムのイメージを図6、経過のイメージを図7に示す。



図6. フォーラムイメージ

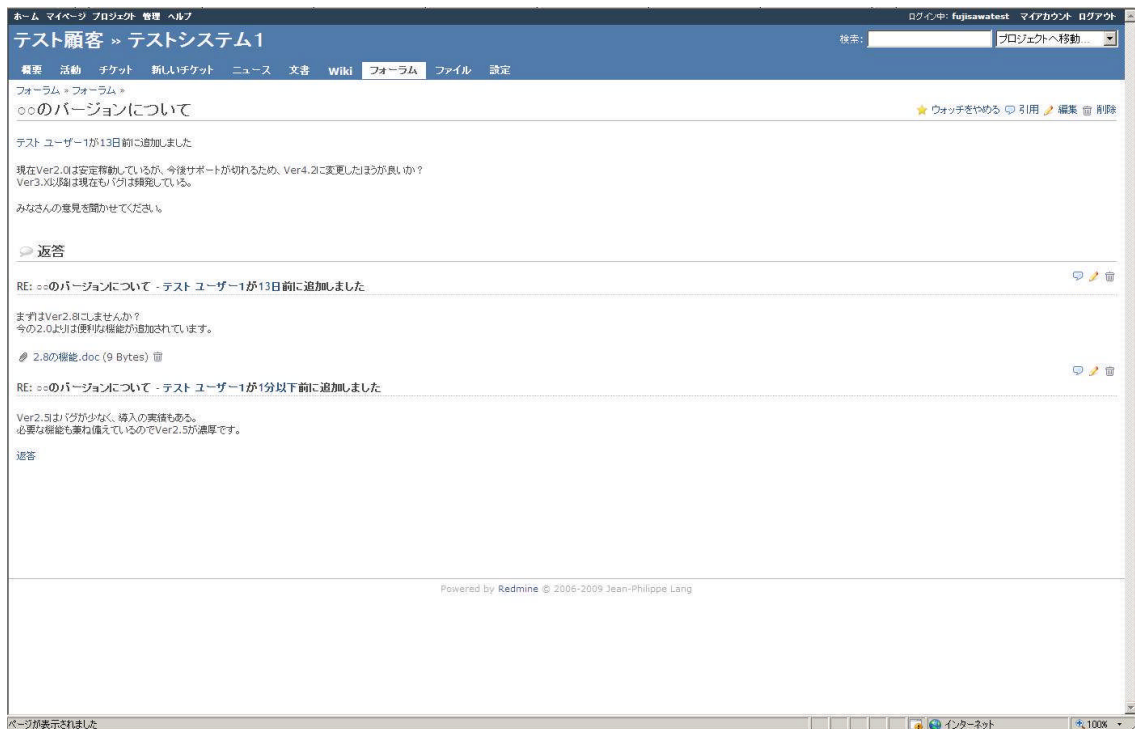


図7. フォーラム経過イメージ

4. プロジェクト管理ソフトウェアの導入効果

4.1 Redmineの導入効果

表計算ソフトベースで情報共有もなく、単純にスケジュールと成果物にだけ重点を置いて管理をしていた時に比べて、試行運用ではあるが、以下の点で参加者から一定の評価が得られた。

(1) タスクの“可視化”が容易に実現できた

プロジェクトメンバーとのチケット情報の共有ができることによって、タスクの状況が誰でも一目でわかるため、リーダーは手作業による“可視化”から解放され、負荷の軽減につながった。

(2) メンバーの意識が高まった

(1)でタスクの可視化が進むことによって、プロジェクトメンバーのプロジェクト管理に対する意識が高まり、進んでチケット情報を更新するようになったことでプロジェクトの一員としての自覚、意識が高まった。これは予想していなかった効果である。

(3) 打ち合わせ、会議の回数が減った

従来の“アナログ管理”では状況の把握が難しかったため、リーダーがメンバー全員、もしくはサブシステム単位にメンバーを定期的に招集し、長い時間をかけて状況報告を受け、表計算ソフトに入力してようやく状況が見え始め、そこから指示を出して打合せをクローズ…という単調かつ労力を要する作業を延々と積み重ね、繰り返していた。

Redmineの導入によってリーダーは一目で状況を把握できるようになったため圧倒的に打ち合わせ、会議の回数・時間が削減された。この結果リーダー自身にも余裕が生まれ、リーダー自身も作業の一部を担当できるようになるほど、リーダーの管理工数削減にも大きく寄与する結果となった。また、プロジェクトリーダーは複数のプロジェクトを担当している場合が多く、上司への報告のためにプロジェクト別の進捗状況をまとめる“儀式”のために時間を割かれることが非常に大きな負担であったが、この点もタスクの“可視化”が進んだことにより状況をまとめる作業が容易になり、負担軽減に寄与する結果となった。

(4) ガントチャート作成機能が意外と重宝した

進捗状況を把握する上で最もわかりやすい（見やすい）表現方法としてガントチャートを作成することが多いが、過去に様々なツール、ソフトウェアを試してみたがなかなかいいものがなかったため、手作業で表計算ソフトを使ってガントチャートを作っていた。Redmineにはガントチャート作成機能があることで、この負荷が軽減され、非常に重宝した。

5. 今後の課題

Redmineを導入，試行してみたことで，その効果については一定の評価ができた．と同時に，当初の期待に反して以下のような課題も見受けられた．

(1) タスク管理の単位が粗い

いろいろと試してみたが，タスク管理の単位が「日」であり，もう少し細かい単位（時間）で管理できないと厳しい．特に当社のような規模の会社では1ヶ月以内で収束するような小規模プロジェクトが輻輳するケースが多く，管理単位が「日」だけでは精度が粗くなってしまう．

またタスク自体も一つでしか管理できないため，プロジェクトをサブシステム単位に管理したい場合はタスクを複数作成するしかなく，タスクの集まりを一つのプロジェクトとして表現する機能が乏しい．結局は，ある部分においては表計算ソフトなどとの組み合わせによる管理が必要で，一長一短であった．

(2) 外部アプリケーションとの連携が弱い

最も残念だったのは，Redmineで“可視化”できるものは外部のアプリケーションと連携できないことだった．Pdfを作成することは可能であったが，最も使用頻度の高い表計算ソフトなどとの連携機能（Import/Exportの機能）が見当たらなかった．

更に，せっかくガントチャート作成機能があるのに自動更新されないのも非常に残念だった．

(3) 用語（日本語訳）の意味が理解できなかった

プロジェクト状況可視化の基本となるのは大日程（マイルストーン）に対してのサブタスクの進捗状況であるが，このマイルストーンを表現する機能（方法）が見つけられず，この機能はないものだと誤解していた．インターネットで検索した結果，あるサイトで「バージョン」という機能の使用法がマイルストーンを表現する方法に酷似していると知り，公式サイトを見ただけでは機能の全容を理解しにくいことを実感した．

もしかすると，他にも便利な機能があるにも関わらず気がついていないものがあるのではないかと感じた．

以上を総合的に勘案した結果，Redmineの使用はプロジェクトの性質によって向き・不向きがあると感じた．当初はプロジェクトの規模が大きくなればなるほどRedmineがその効果を発揮してくれるものと期待していたが，小規模～中規模プロジェクト程度がRedmineの守備範囲ではないと感じた．

6. おわりに

今回のプロジェクト管理ソフトウェアの導入、運用における成果の詳細については、本文の4章「プロジェクト管理ソフトウェアの導入効果」で述べたとおりであるが、顕著な効果としては、従来の表計算ソフトベースの管理に比べてリーダーの負担が軽減された点と、メンバーの意識が高まった点である。ものごとが“可視化”されることで人間の意識は変わるものだということを改めて実感した一方で、プロジェクトの規模によつての向き、不向きがあることもわかってきたので、今後は適用に際しての基準を設ける、といった社内での運用ルールの確立も必要ではないかと感じた。

この他の効果として、業務のマニュアル化というキーワードに対してもプロジェクト管理ソフトウェアの導入は有効ではないかと感じた。

平成16年5月に施行された会社法に「内部統制」についての記載が登場し、組織の業務の適正を確保するための体制を構築していくシステムの確立と、業務のマニュアル化などが求められているが、現実にはきちんと理解されていないようである。

このような現状からも、技術の継承、業務の引継ぎ等を「形」に残していく準備をしておくことは情報産業に携わる企業として急務と考えられ、この点についてもRedmineのようなソフトウェアを利用したプロジェクト管理をより効果的に適用していくことが出来れば業務のマニュアル化ができ、且つ、技術の継承などの問題にもスムーズに対応することができるのではないかと感じた。

最後になるが、プロジェクト管理ソフトウェアの効果については疑問の余地はないが、プロジェクト管理ソフトウェアはあくまでも“ツール”であり、人間の手助けをするものである。“ツール”の導入と共にリーダー候補となる優秀な人材の育成にも力を入れて行く必要性も実感している。