

---

---

# 生保大規模プロジェクトにおける Redmine を活用したプロジェクト管理スキームの構築

## 東京海上日動システムズ（株）

---

### ■ 執筆者 Profile ■



原田 幸樹

2013 年 東京海上日動システムズ（株）入社  
生保本部 所属

### ■ 論文要旨 ■

筆者はプロジェクト管理担当者としてプロジェクトの円滑運用のサポートを実施している。Excel の WBS によるプロジェクト管理を社内標準としている中で、大規模プロジェクトにおける導入効果が大きいことから Redmine という WEB ベースのプロジェクト管理ソフトの利用を決定した。本稿においては初期構築から始まり、160 名の利用者に対する運用指針の提示、実情の可視化を目指しての工夫など、現在に至るまでのプロジェクト管理スキーム形成について論ずる。

筆者は本プロジェクトの管理担当としての経験を経て、利用者の視点に立った運用の重要性を学んだ。今後も定期的に自らの経験・行動を振り返ることで成長していきたいと考えている。

## ■ 論文目次 ■

<b>1. はじめに</b> .....	《 4》
1. 1 東京海上日動システムズ株式会社の概要	
1. 2 生保大規模プロジェクトの概要	
<b>2. プロジェクトへのRedmineの導入</b> .....	《 4》
2. 1 Redmine の概要	
2. 2 Redmine の基本用語	
<b>3. 初期構築のポイント</b> .....	《 5》
3. 1 ワークフローの導入	
3. 2 Redmine プロジェクトの分割単位	
3. 3 チケットの粒度	
<b>4. Redmine 活用の促進</b> .....	《 8》
4. 1 ナレッジベースの開設	
4. 2 CSV インポートツールの提供	
4. 3 カスタムクエリの提供	
<b>5. プロジェクトの見える化の追求</b> .....	《 10》
5. 1 プロジェクトの進捗指標	
5. 1. 1 チケット件数管理	
5. 1. 2 EVM 管理	
5. 2 チケットモニタリング	
<b>6. おわりに</b> .....	《 12》
6. 1 今後の課題	
6. 1. 1 Redmine を弊社標準とする上での課題	
6. 1. 2 Excel の作業工数における課題	
6. 1. 3 製造工程とテスト工程への適用における課題	
6. 2 所感	

■ 図表一覧 ■

図 1	Redmineプロジェクト内のイメージ	《 5》
図 2	ワークフローの運用イメージ	《 6》
図 3	Redmineプロジェクト構造	《 7》
図 4	ナレッジベースのイメージ	《 8》
図 5	CSVインポートツールのイメージ	《 9》
図 6	カスタムクエリ一覧	《 10》
図 7	モニタリングのチケット詳細	《 12》
図 8	モニタリングのサマリー表	《 12》
表 1	進捗指標一覧	《 10》
表 2	進捗指標の説明	《 11》

## 1. はじめに

### **1. 1 東京海上日動システムズ株式会社の概要**

東京海上日動システムズ株式会社（以下、弊社とする）は 1983 年 9 月に設立され 10 本部 28 部から成り東京海上グループの情報システムの開発・保守・運用を担う。私の所属する生保本部は東京海上日動あんしん生命保険株式会社のシステム開発・保守・運用を担当している。

### **1. 2 生保大規模プロジェクトの概要**

当プロジェクトはホストコンピュータ中心の現行システムをサーバ環境で再構築することで開発・保守・運用のコスト削減とシステム変更の柔軟性を確保することを目指す。オープン化対象のプログラムは約 1,000,000 ステップの大規模プロジェクトであり、2017 年度内のサービスインを計画している。私は本プロジェクトの PMO の一員として進捗管理スキームの構築とツールの導入を担当している。

## 2. プロジェクトへの Redmine の導入

弊社では、Excel による WBS が標準テンプレートとして提供されているが、複数人が同時に更新できないことなどが課題となっていた。以下のポイントから新規の進捗管理ツールの導入効果が大いだと判断し、製品の検討を行った。

- ・ 参画人数が多いプロジェクトという特性から複数人が同時に更新できること
- ・ 様々な切り口で容易に進捗把握ができること
- ・ 参照制限、権限変更などのユーザー管理が容易であること
- ・ サーバのログを参照することで、更新履歴の確認ができること
- ・ Excel 管理のように複数のファイルが作成され、最新版の所在が不明にならないこと

検討の結果、OSS として国内で多くの利用実績がある Redmine の導入を決定した。さらに Redmine の標準機能のみでなくワークフローの導入、データを出力し進捗報告資料を作成するなどカスタマイズを行い、進捗管理スキーム全体を検討し構築した。

Redmine の概要と基本用語については以下のとおりである。

### **2. 1 Redmine の概要**

WEB ベースでプロジェクトのタスク管理、進捗管理、情報共有を行うことができる、オープンソースのプロジェクト管理ソフトウェアである。（<http://redmine.jp/>）

### **2. 2 Redmine の基本用語**

本論文内で論述する内容であるため、下記のとおり解説を行う。また、Redmine プロジェクト内のイメージを図 1 に示す。

- ・ Redmine プロジェクト  
Redmine でチケットを分類する最も大きな単位である。
- ・ チケット

タスク・課題を記載した作業指示票のこと。  
チケットは必ずRedmineプロジェクトに所属する。



図1 Redmine プロジェクト内のイメージ

### 3. 初期構築のポイント

#### 3. 1 ワークフローの導入

ワークフロー導入の目的は2点ある。1つ目の目的は、タスク着手条件の明確化である。そのために、開始日、期日、予定工数、成果物を必須入力項目とし、その内容をチームリーダーが承認することでタスクが開始される。2つ目の目的は、タスク完了条件の明確化である。タスクには必ず成果物が存在するという前提のもと、終了時にチームリーダーによる成果物チェックを義務付けた。

これらを図2のとおりワークフロー機能でサポートし、タスクの着手前と完了時にチームリーダーの承認をもって次工程へ進めるようにした。

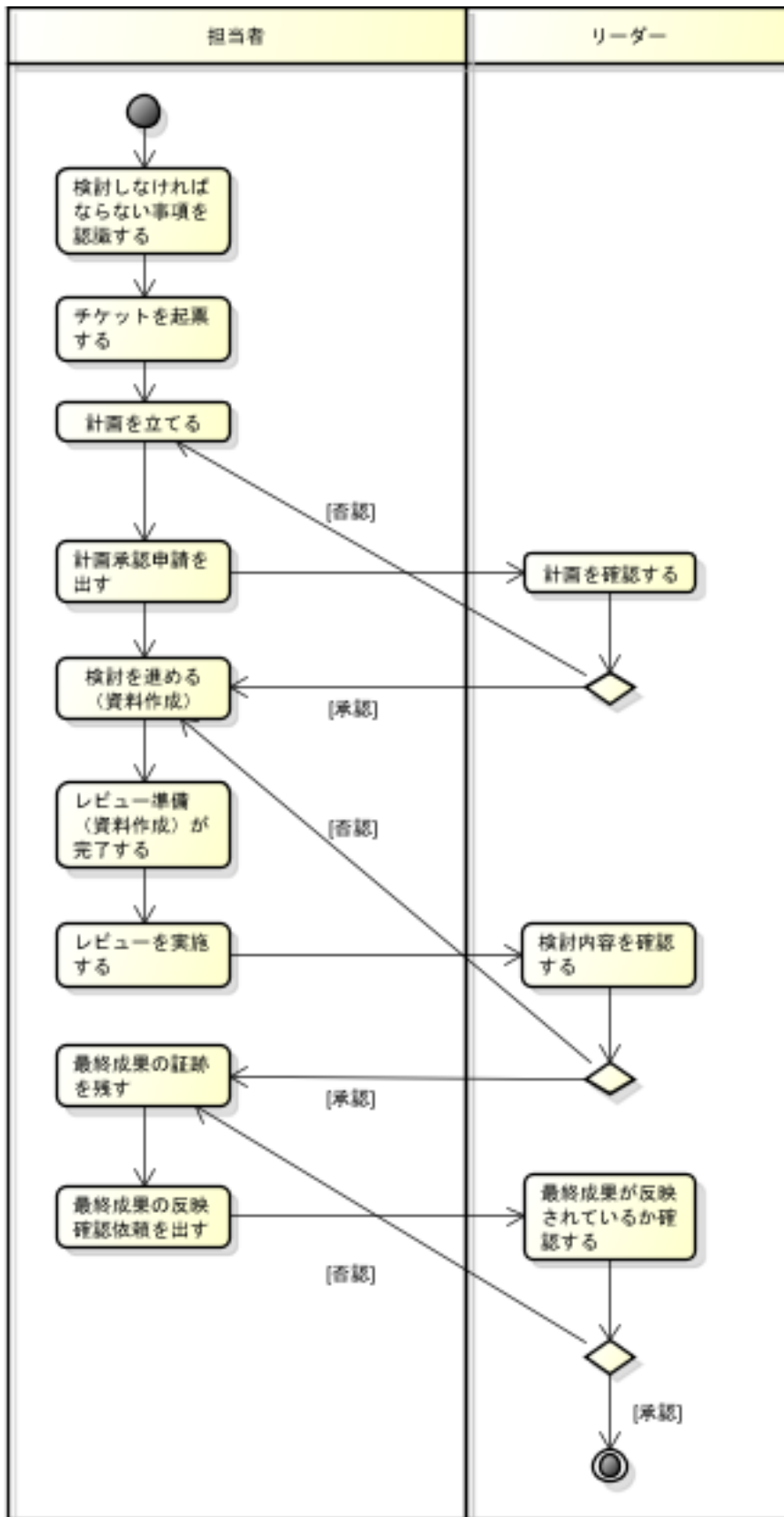


図2 ワークフローの運用イメージ

### 3. 2 Redmine プロジェクトの分割単位

進捗や課題を以下のとおり様々な切り口で把握できるよう、Redmine プロジェクトの階層構造を定めた。

- ① 業務 — 新契約、保全、保険金、稼働時間
- ② システム開発工程 — SA、UC、SS など<sup>注1</sup>
- ③ ワーキンググループ — 要件検討ならびに決定を目的とした IT 部門、ビジネス部門 合同の組織体
- ④ システム機能 — DB モデリング、業務アプリケーション、実行基盤、開発基盤 など

その結果、当初は図 3 のとおり構造を決定した。さらにプロジェクトが進行するにつれて、開発工程の追加、システム機能内でのフェーズ分割など追加で細分化を行い、現在までに 167 プロジェクトを作成している。

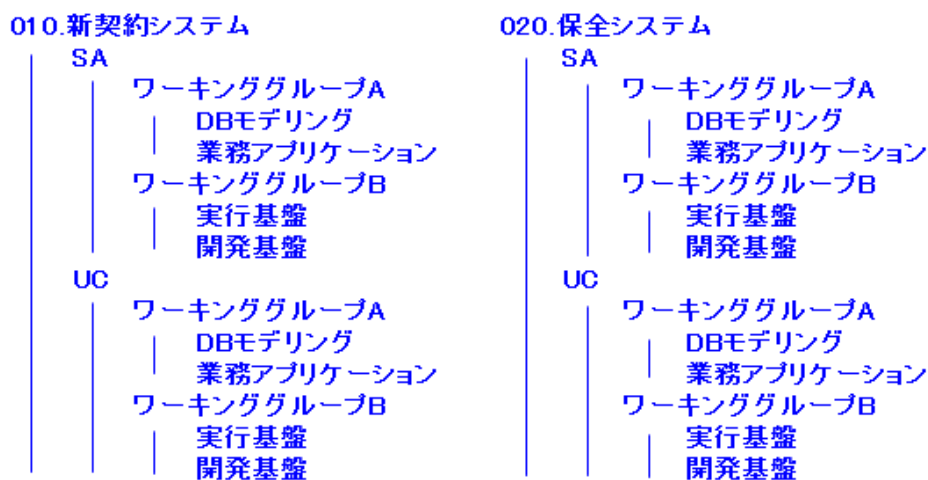


図 3 Redmine プロジェクト構造

### 3. 3 チケットの粒度

特に上流工程においてはタスクの粒度のバラつきが生じやすく、平準化した件数管理ができない懸念があった。それを解決するため 1 チケットあたりの作業期間が最長 1 週間となるように、タスクブレイクを行うよう指示した。また、UC 工程以降については、設計書単位でタスク登録をさせることで、ブレイクの単位を明確にした。

注 1

SA: System Analysis の略語であり、要件定義を行う工程を表す。

UC: User Interface & testCase design の略語であり、

外部設計およびテストケース設計を行う工程を表す。

SS: System Structure design の略語であり、詳細設計を行う工程を表す。

## 4. Redmine 活用の促進

### 4.1 ナレッジベースの開設

Redmine 上で質問・要望を完結できる仕組みを目指し、Redmine の標準機能では搭載されていないプラグインを導入することで図 4 に示すとおりナレッジベースを開設した。開設の際には利用の促進を目的とし、想定される QA を管理者があらかじめ記載した。その結果、同じような悩みをかかえる利用者の円滑な疑問解決に貢献した。さらに、要望についてもナレッジベースに集約することで、要望を元に Redmine のカスタマイズを行い、ユーザー参画型で育てていくことができた。

ホーム マイページ プロジェクト 個人情報 管理 ヘルプ

# 000. 事務局

概要 活動 ナレッジベース チケット 新しいチケット Dashboard ガントチャート カレンダー

Home

## Redmine機能についてのご相談

Redmine機能についてのQAやご要望を記載してください。

Title ▲
EVMによる進捗管理
Q: Dashboard機能について教えてください(作成中)
Q: チケットに親子関係を作りたいのですが
Q: チケットの粒度はどう考えれば良いですか
Q: リーダーの計画承認のポイントは?
Q: 標準で準備されているクエリを教えてください。
Q: 課題一覧表, QA一覧表を作成し、オーナーに送付したいのです
Q: 進捗率/ステータスの考え方を教えてください
Q: Excelファイルをインポートしてチケットを作成する機能はございますか? 今までExcelファイルでタスク管理していたため。
「課題」の定義について
【申請】Redmineユーザ追加
【申請】メーリングリスト追加
【申請】開発ツール導入
チケットのモニタリング要領
チケットビューが見づらいのですが - グループによる親子関係
ユーザーを追加したいです
優先度はどのような観点で記載すればよいですか?
親子関係のあるチケットの子の進捗を進めた時、親もすすんでくれたりしませんか? 「グループ」や子タスクを持つ「課題」も子タスクの進捗に合わせて、手で個別に修正を入れないと進捗していることにならないのでしょうか。(修正が二度手間で見直し)
週次進捗資料作成手順
進捗管理資料 - 2015/5/25 現在

図 4 ナレッジベースのイメージ



## 4. 2 CSV インポートツールの提供

プロジェクト立ち上げ時には大量のチケット登録が必要とされるが、WEB 画面からのチケット登録は作業時間を要するため、一括でチケットを登録できる機能を提供した。具体的には CSV ファイルからのインポートを実現するために機能拡張のプラグインの導入と操作方法のマニュアル整備を行った。10 チケットで時間を測定したところ、6 分かかる WEB 画面からの登録が 2 分で実施でき、単純計算で 3 倍の効率化を実現することができた。

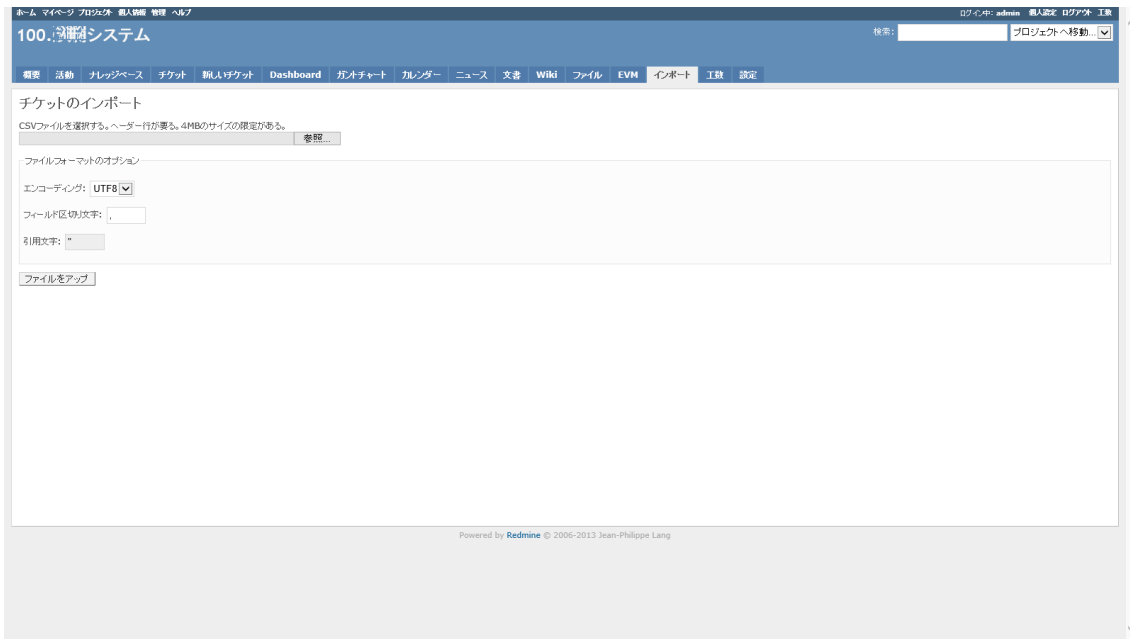


図 5 CSV インポートツールのイメージ

## 4. 3 カスタムクエリの提供

7000 件に及ぶ大量のチケットから特定のステータスや進捗状況の思わしくないものを抽出するために、フィルタリング条件を保存できるカスタムクエリ機能を利用した。全利用者が共通して利用すると想定した図 6 の項目について運営管理者がフィルタ機能を提供することで、均質化したマネジメントを実現した。例えば、「120. 過去 5 日以内に更新がないチケット」のボタンを押下した場合には、チケットを起票したものの作業が行われていないチケットや成果物を作成したもののチームリーダーの確認が行われず完了できないチケットなどを洗い出すことができる。

## カスタムクエリ

000.ステータス=新規  
010.ステータス=承認待  
020.ステータス=最終成果反映確認待  
030.ステータス=却下待  
110.未完了のチケット  
120.過去5日以内に更新がないチケット  
210.期限切れチケット(計画承認前)  
215.期限切れチケット(計画承認後)  
220.期限が近いチケット(期限切れ含む)  
300.『タスク』一覧表示  
310.『課題』一覧表示  
320.『オーナーQA』一覧表示  
500.標準ビュー  
510.進捗資料作成  
520.モニタリング資料作成  
610.開始日入力なし  
620.期日入力なし  
630.予定工数入力なし

図6 カスタムクエリ一覧

## 5. プロジェクトの見える化の追求

プロジェクトの見える化追求のために、Redmine の標準機能ではサポートされていない進捗管理・チケットのモニタリングの仕組み(後述)を構築した。具体的には Redmine からのダウンロードデータを Excel で加工・編集することで実現している。

### 5.1 プロジェクトの進捗指標

進捗報告時はシステム機能ごとに表1のような数値項目の算出を行った。また、算出した数値項目の説明については表2のとおりである。大別してチケット件数と EVM の2つの指標を用いて進捗管理を行った。

	総数 ①	完了予定 ②	完了済 ③	総完了数 ④	進捗率 (③/②)	総完了率 (④/①)	SV (EV - PV)	SPI (EV / PV)
x x システム	112	96	93	101	96.9%	83.0%	-55.03	0.96

表1 進捗指標一覧

数値項目	説明
総数①	チケットの総数。
完了予定②	基準日までに完了しているべき予定のチケット総数。
完了済③	②のうち、完了している総数。
総完了数④	期限到来/未到来を問わず、完了しているすべてのチケット総数。
進捗率 (③/②)	完了予定チケットの完了率 期限どおりに終わるべきものが終わっているか
総完了率 (④/①)	全チケットの完了率 スケジュール対比で、完了度合が妥当か
PV (Planned Value)	基準日時点で終了しているべき出来高 (= 予定工数/時間) Ex) 予定工数10時間の仕事が、 期限の中間点では5時間分出来ていると見込む
EV (Earned Value)	基準日時点の出来高 Ex) 予定工数10時間の仕事が、60%の進捗であれば、 6時間を出来高として算出
SV (EV - PV)	予(PV)実(EV)の差分。 予定どおりの出来高が得られていないと、指標がマイナスになる。
SPI (EV/PV)	予(PV)実(EV)の差分を比率で表現した指標。 予実がイコールであれば、1.00 になり、1.00 未満は実績が予定に至っていない。

表 2 進捗指標の説明

### 5. 1. 1 チケット件数管理

チケット件数は、基準日までに完了しているべき予定のチケット総数や全チケットのうちの完了率を示した。

### 5. 1. 2 EVM 管理

チケット件数管理のみでは、各チケットの工数の大小を加味した管理が不十分になる。例えば、工数が少ないタスクは完了しているものの、工数がかかるタスクが遅れているといったことが挙げられる。このような遅延を把握するために EVM を導入した。具体的には基準日ベースの予定工数(PV)に対する出来高(EV)を算出している。

### 5. 2 チケットモニタリング

先に述べたとおり、カスタムクエリの機能を利用し、チケットの起票者・チームリーダーが主体的に進捗状況を管理できるツールを提供した。しかし、カスタムクエリでは、前週と比較した工数の増減や期限変更が把握できない。チームリーダーが予期していない工数変更や期限変更を捕捉するために、チケットのモニタリングを行い、管理情報を提供した。実施したモニタリング項目は以下のとおりである。

- ・チケットの新規作成
- ・チケットの削除
- ・開始日遅れ、期限切れ
- ・開始日、期日、予定工数の入力有無
- ・開始日、期日、予定工数の前週からの変更

図7のようにモニタリングで検知されたチケットはそのチケット番号と詳細の情報を公開した。そうすることで各チームがとらえきれていなかった潜在的な遅れやタスク管理の不備などを発見することができるようになった。また、図8のようにチームごとのサマリー表を出すことでプロジェクト全体の実状を把握することができた。モニタリングで検知されたアラートについては各チームに是正の指摘を行うことでプロジェクト管理を強化することができた。

モニタリング情報				チケット情報							
連番	モニタリングNO	モニタリング名	#	プロジェクト	トラッカー	ステータス	進捗%	題名	開始日	期日	予定工数
1	9	期限切れ	7587		010.タスク	新規	0		2015/7/31	2015/8/7	4
2	9	期限切れ	5707		010.タスク	新規	0		2015/6/23	2015/7/1	10
3	9	期限切れ	14715		010.タスク	050.最終成果反映確認待	80		2015/10/7	2015/10/23	20
4	9	期限切れ	14716		010.タスク	050.最終成果反映確認待	80		2015/10/13	2015/10/23	20
5	9	期限切れ	14717		010.タスク	050.最終成果反映確認待	80		2015/10/1	2015/10/16	10
6	9	期限切れ	14718		010.タスク	050.最終成果反映確認待	80		2015/10/13	2015/10/23	20
7	9	期限切れ	14719		010.タスク	050.最終成果反映確認待	80		2015/10/13	2015/10/23	20
8	9	期限切れ	14720		010.タスク	050.最終成果反映確認待	80		2015/10/13	2015/10/23	20
9	9	期限切れ	14721		010.タスク	050.最終成果反映確認待	80		2015/10/13	2015/10/23	20
10	9	期限切れ	14722		010.タスク	050.最終成果反映確認待	80		2015/10/13	2015/10/23	10
11	9	期限切れ	14723		010.タスク	050.最終成果反映確認待	80		2015/10/13	2015/10/23	10
12	9	期限切れ	14729		010.タスク	050.最終成果反映確認待	80		2015/10/5	2015/10/23	20
13	9	期限切れ	14730		010.タスク	050.最終成果反映確認待	80		2015/10/13	2015/10/23	20
14	9	期限切れ	14731		010.タスク	050.最終成果反映確認待	80		2015/10/13	2015/10/23	20
15	9	期限切れ	14732		010.タスク	050.最終成果反映確認待	80		2015/10/13	2015/10/23	20
16	9	期限切れ	14481		010.タスク	038.進行中(80%)	80		2015/10/14	2015/10/27	20
17	9	期限切れ	14484		010.タスク	020.計画承認済	0		2015/10/18	2015/10/23	10
18	9	期限切れ	10866		010.タスク	040レビュー待	80		2015/10/1	2015/10/23	32
19	9	期限切れ	10872		010.タスク	040レビュー待	80		2015/10/5	2015/10/23	8
20	9	期限切れ	10874		010.タスク	040レビュー待	80		2015/10/5	2015/10/23	40
21	9	期限切れ	10876		010.タスク	040レビュー待	80		2015/10/5	2015/10/23	64
22	9	期限切れ	10878		010.タスク	040レビュー待	80		2015/10/5	2015/10/23	40
23	9	期限切れ	4670		010.タスク	035.進行中(50%)	50		2015/7/1	2015/9/3	32
24	9	期限切れ	5353		010.タスク	020.計画承認済	0		2015/8/17	2015/8/28	32
25	9	期限切れ	11613		010.タスク	038.進行中(80%)	80		2015/9/28	2015/10/23	8
26	9	期限切れ	11614		010.タスク	038.進行中(80%)	80		2015/9/28	2015/10/23	16
27	9	期限切れ	11615		010.タスク	038.進行中(80%)	80		2015/9/28	2015/10/23	8
28	9	期限切れ	11616		010.タスク	040レビュー待	80		2015/9/28	2015/10/23	16

図7 モニタリングのチケット詳細

			開始日遅れ			期限切れ			開始予定日の変更			期日の変更			予定工数の変更			新規作成したチケット		
			優先度・大	優先度・中	優先度・小	優先度・大	優先度・中	優先度・小	優先度・大	優先度・中	優先度・小	優先度・大	優先度・中	優先度・小	優先度・大	優先度・中	優先度・小	優先度・大	優先度・中	優先度・小
新契約事務局	-	課題	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		タスク	0	5	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0
SA	課題	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	タスク	0	3	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	
UC	課題	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	タスク	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
SA	課題	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	タスク	0	0	0	1	5	0	0	0	0	0	0	0	0	0	0	0	0	0	
UC	課題	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	タスク	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
SA	課題	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	タスク	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	
UC	課題	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	タスク	0	4	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	12	
SA	課題	1	1	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	1	
	タスク	0	2	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	1	
UC	課題	0	3	0	0	1	0	0	0	0	0	3	0	0	0	0	0	0	1	
	タスク	0	74	0	0	16	158	0	40	8	0	48	0	1	1	2	626	95	0	
SA	課題	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	タスク	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
UC	課題	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	タスク	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

図8 モニタリングのサマリー表

## 6. おわりに

プロジェクト管理に Redmine を導入し運用スキームを整備したことで従来主流であった Excel の WBS と比較し、先に述べたとおり様々なメリットを享受することができた。弊社では先進的な取組であり課題や検討事項が多かったものの、利用者とのコミュニケーションを密に取り一つ一つ真摯に向き合うことで繰り返し改善を行い、新しい管理スキームを

構築することができた。

## **6. 1 今後の課題**

### **6. 1. 1 Redmine を弊社標準とする上での課題**

当プロジェクトの Redmine を用いたプロジェクト管理手法については、弊社の品質管理部門へ共有し、特に以下の点で一定の評価を受けている。

- ・誰もがリアルタイムにタスクの更新やプロジェクトの状況把握をできる
- ・Excel で Redmine の機能補完を行っている

弊社内ではまだ Redmine を利用したプロジェクト管理スキームがなく、社内に展開するためには課題がある。そのため3章で述べたとおり利用開始までの検討に多くの時間を要している。プロジェクトの分割基準を決定する際には、多くの関係者の意見を統一することに苦労した。そのため、弊社内での Redmine を用いたプロジェクト管理を推進していくためには、今回のようなプロジェクトの実例を展開し標準の管理手法を定める必要がある。Redmine のダウンロードデータをインプットにして、自動的に弊社標準の管理指標（進捗・品質）や提出資料が算出／作成できる仕組みを構築することで、より導入の効果が高まる。さらに委託ベンダーが独自に有する管理ツールとの重複（二重入力）についても、解決すべき課題である。

### **6. 1. 2 Excel の作業工数における課題**

今回のプロジェクト管理スキームの検討・構築において、管理ツールの融合による効果大きいという気づきがあった。具体的には、大規模プロジェクトの管理に強い Redmine を基礎として、情報の見せ方の汎用性が高い Excel を組み合わせることにより互いの長所を活かした効果的なプロジェクト管理を実現することができた。一方、7000 件を超える Redmine のチケットを Excel でいかに正確に効率よく処理するかについては今回の構築で最も苦労した点のひとつである。現状は Excel による情報の変換処理に多くの工数がかかっているため、作業の効率化は今後の課題である。

### **6. 1. 3 製造工程とテスト工程への適用における課題**

SA から SS までプロジェクト管理を行ってきたが、今後は製造工程と IT・ST 工程のプロジェクト運用を検討する必要がある。製造工程はプログラム単位の管理を行った場合、タスクの粒度が非常に細かくなり、Redmine への入力の作業ロードが大きくなるという懸念がある。また、当プロジェクトにおいては SS 工程で作成するドキュメントの一部を javadoc で代替するため、SS 工程と PG 工程を一元管理することが望ましい。さらに、IT 工程以降については、進捗・課題管理のみでなくテストケース管理、障害管理、品質指標の算出などを行っていく必要があり、さらに Excel への依存が高まることが危惧される。

## **6. 2 所感**

進捗資料やモニタリング資料で算出した数値の妥当性の判断やプロジェクトの潜在的なリスクを捉えられるようになることが今後私に求められる役割であると考えている。これ

までは重要な情報をいかに蓄積しプレゼンテーションするかを追求してきたが、これからは提示した情報を私自身が読み取り、必要に応じて是正していける力を身につけたい。

また本担当の経験を通し、PMO としてあるべきプロジェクト管理スキームの構築を行いながらも、チームリーダーや担当者の目線に立つ重要性を学んだ。この学びは将来担当や役割が変わった際にも役立つ、業務を行う上での基礎となる考え方である。例えば今後アプリケーション開発担当者になった場合にも、IT 部門側の理想を追い求めるだけでなくビジネス部門の要求にも応えられるよう、どちらの視点にも立てている人材を目指す。

さらに本論文を執筆したことにより、業務内で検討を行った内容、そこから得た学びについて体系的に整理でき、自らの行動や選択に対する自信をもつことができた。今後も定期的に自らの業務を振り返り、気付きや知識を見直すことで成長していきたい。

## **参考文献**

なし