

Redmine による保守サービスの運用について

ソリューションサービス本部 システム開発部

篠塚 隆広

1. はじめに

業務システムの運用開始後に発生する問題として、環境面の問題、ユーザ部門からの改善要求、マスタもしくはプログラムの不具合と思われる異常動作等があり、その対処を「保守サービス」として対応するケースが多い。

保守サービスで発生する作業について、主に以下の特性がある。

- ① いつ発生するかを計画/予見できない
- ② 優先度が異なり、割込みが随時発生
- ③ 作業状況報告が随時必要
- ④ 瑕疵保証内と保証外の作業が混在
- ⑤ 各作業分のリリース予定、実績管理が必要

この保守サービスを定型化し、お客様と当社の双方に対して保守サービスに関する管理作業を軽減させる目的で、BTS(Bug Tracking System) / ITS(Issue Tracking System)として当社内の開発案件でも利用されている Redmine を利用した。

Redmine を利用することで、「保守サービス」について一定の考課と課題が確認できた為、本稿でその結果を報告する。



図 1 チケット管理画面

2. 「Redmine」とは

Redmine とは Web ベースでプロジェクトのタスク管理を行うソフトウェアである。チケットを発行し、タスク管理を行う。チケットとはタスクに対応し、1 つのチケットに 1 つの事柄を記載する。1 プロジェクトに対して複数のチケットが登録され、チケットは必ずプロジェクトに所属する。チケットはステータスを持ち、「新規」「進行中」「フィードバック」「解決」「終了」等がある。

3. 「Redmine」導入前の問題点

Redmine 導入前の問題点について下記に記載する。

- ① Excel 管理の為、作業効率が悪い
毎回、記入⇄連絡⇄確認の手順でチェックする必要があり、運用面で余計な時間がかかった。

ID	担当者	依頼者	依頼内容	依頼日時	対応状況	対応日時	対応者	備考
001	田中	山田	システム運用中に発生しているエラーの調査と修正。エラー発生時のログを収集し、原因を特定する。修正後、テストを実施し、問題が解消するか確認する。	2023/10/01	調査中	2023/10/01	田中	エラー発生時のログを収集し、原因を特定する。修正後、テストを実施し、問題が解消するか確認する。
002	田中	山田	システム運用中に発生しているエラーの調査と修正。エラー発生時のログを収集し、原因を特定する。修正後、テストを実施し、問題が解消するか確認する。	2023/10/01	調査中	2023/10/01	田中	エラー発生時のログを収集し、原因を特定する。修正後、テストを実施し、問題が解消するか確認する。

図 2 Excel 課題管理票(例)

- ② Excel 管理の為、内容が見にくい
1 事象に対して、1 行で管理する為、フィードバックが発生した場合に、時系列が分かり難くなる問題や、複雑な問題を扱う場合に、1 セルに状況を書ききれない問題が発生した。
- ③ 時系列が不明となる
途中経過をメールでやり取りしていた為に、結

果だけが Excel に記載された状態となり、経緯について不明なケースが発生した。

④ 類似の内容が有効活用できない

過去に類似現象があった場合に、途中経過が未記入であったり、資料が散在していることにより、前回と同様の調査を行ったり、資料を検索する為に余計な時間がかかった。

⑤ リリース管理が煩雑

試験環境、運用環境など、複数の環境があり、どの環境に対して、どのバージョンのモジュールを納品したかの管理を Excel で行っていた為にリリース作業前後での管理作業が煩雑になっていた。

4. 「Redmine」導入後の管理実績

4.1 不具合管理

不具合(バグ)に対する問い合わせ内容について、優先度や、進捗率、カテゴリ、要因等の項目を利用し、Redmine 上で管理する。

(1) 運用方法

客先からのメールまたは電話にて不具合に関する連絡を受け、チケット登録し、対応を行う。

処理フローは 図 4 に記す。

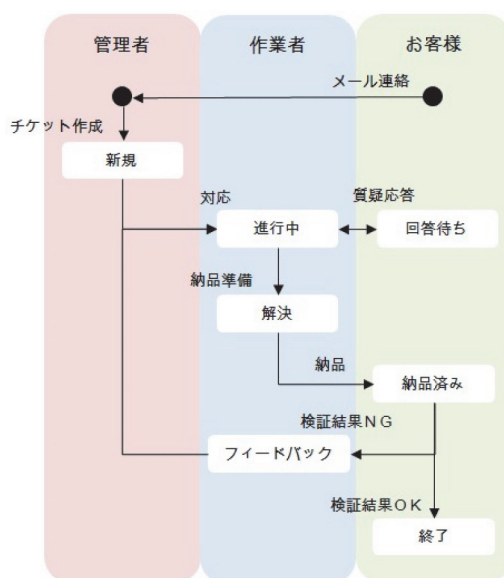


図 4 処理フロー

4.2 仕様変更管理

仕様変更に関する内容について管理番号や予想工数等の項目を利用し、Redmine 上で管理する。

(1) 運用方法

不具合管理同様の処理フローで運用を行う。

質疑応答のやり取りが不具合管理の時よりも増える。

4.3 リリース管理

試験環境、運用環境など複数の環境へ、どのバージョンのモジュールをいつ反映し、どこまで確認が完了しているかをバージョン、試験/運用環境の反映日、動作確認日等の項目を利用し、Redmine 上で管理する。

(1) 運用方法

不具合管理や仕様変更で発行したチケットの作業完了後、バージョンを設定し、リリース作業を行う。

5. 「Redmine」導入によるメリット

Redmine を導入した結果、以下のメリットが得られた。

①チケットの状態管理が分かりやすくなった

フィルタを掛けることにより、未完了のチケットのみを抽出し、該当のチケットの状態がどうなっているか一目で確認できるようになった。

従来の Excel 管理では、リアルタイムの進捗状況が分かり難く、詳細は担当者でしか把握していないことが多かった為、チケットに登録することで、現状の進捗状況の管理が容易になった。

②情報共有のリアルタイム性の向上

従来の Excel 管理の場合、状況を記入後、別途メールにて資料を添付、または共有のリポジトリに対して資料をコミットし、ユーザに通知しなければならなかったが、Redmine の場合、チケット更新と共に自動でメールが通知される為、関係者間にてリアルタイムで情報を共有できるようになった。その結果、発生した不具合対応や、仕様変更の進捗状況がリアルタイムで確認できることとなり、メールや電話で状況を確認する手間がなくなるといったメリットが生まれた。

③過去の類似チケットの検索や関連付が容易

Excel管理の場合、検索を行うことは出来るが、関連する内容に関して紐付ができないといったデメリットがあり、また、類似の内容に関して Excel 管理票や、メール等、情報リソースが複数あるが、Redmineの場合は情報リソースが 1 つに集約されている事と、容易にチケット間で関連付が出来る事により、容易に必要な情報が確認でき、資源の有効活用を行うことが可能となった。

④リリース管理負担の低減

どの不具合対応のモジュールをリリースするか、どの環境にどのモジュールが入っているかといったリリース管理が一元化できる為、リリース計画を容易に行うことができ、結果的にリリース時の管理作業の負担が減った。

また、チケットの関連付が出来る事により、複雑なモジュール構成のリリース時にもトラブルなく、容易に対応することができ、リリース失敗に伴う

切り戻し作業においても、戻し忘れ等の作業ミスの防止にも役立てることが出来た。

⑤管理コストの削減

従来の Excel 管理の場合、Excel への記入工数、関係者間での連絡工数、類似内容の検索工数等、工数面で多大な工数を消費し、また、その分管理コストも多く発生していた。Redmine での管理に切り替えることにより、連絡工数は Redmine が自動で行うことにより削減でき、記入工数については Redmine のみへの記入に統一され、工数の削減ができ、結果的に管理コストの削減が大幅に出来た。

6. 導入後に発生した問題

Redmine 導入により発生した問題を以下に列挙する。

①チケットが放置状態となる

<原因>

担当者の設定ミスやペンディング事項などが発生した場合にチケットが更新されずに放置状態となってしまう。

<解決方法>

カスタムクエリを用いて、チケットの状態を適宜チェックし、放置状態にあるチケットについては担当者に更新を促すことで対応。

②1 チケットに複数の課題が発生する

<原因>

チケット対応中に、対応すべき課題が多く発生し、チケットの中身が煩雑化する。

<解決方法>

複数の課題が発生した時点で子チケットを発行するなど、適切な粒度でチケット管理を行うことで対応。

③入力項目が多い

<原因>

チケットに対する入力項目が多く、設定漏れや設定ミスなどが発生しやすい。

<解決方法>

プロジェクト、トラッカー毎に入力項目を最小限となるように精査を行い、入力作業の簡素化を

行うことで対応。

④メールを見落としやすい

<原因>

Redmine の送信メールが多すぎて、捕捉すべきメールを捕捉できない。

<解決方法>

チケット発行時に適切なメンバーにのみ通知を行うようにルール化。

7. その他の機能

Redmine には、上記に挙げた機能以外にも有用な機能がいくつある。その一部を以下に紹介する。

① Wiki

チケットの運用フローや用語集、共有すべき情報などを、wiki として記述することが可能。運用の初期段階で決定したフローなどを掲載しておくことにより、後から加わったユーザにルールが伝達しやすく、容易に参加できる。

②チケットサマリ

プロジェクト内のチケットの発行状況、終了状況を一覧で表示し、グラフ化して表示することが可能。

③その他

「フォーラム」や「ガントチャート」なども作成することができる。必要に応じてプラグインをインストールすることで機能拡張が可能。

目次	
トラッカーの説明	
ステータスの説明	
トラッカーの説明	
トラッカー	説明
QA	調査依頼として起票する時のトラッカー
バグ	不具合について起票する時のトラッカー
改善要望	改善について起票する時のトラッカー
その他	該当するトラッカーが無い場合に利用
ステータスの説明	
ステータス	説明
進行中	調査中や、不具合修正やバグ、チケットに付いた人による作業中のステータス
解決済	修正、改善の対応が完了した時のステータス
納品済み	納品済みとなった時のステータス
フィードバック	納品後に不具合があった時のステータス。問題解決の時に設定します
終了	対応が完了した時に設定します
却下	起票したチケットを取り下げた時に利用します

図 5 wiki 画面

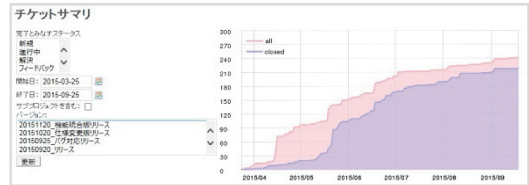


図 6 チケットサマリ画面

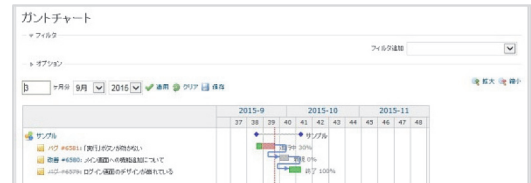


図 7 ガントチャート画面

8. おわりに

このように Redmine を利用した保守サービスを行うことで、管理業務の煩雑さの軽減や、工数の削減、作業の効率化、情報資源の有効活用につながり、お客様、当社双方にとって非常に有益な効果をもたらす。

開発業務だけではなく、納品後の運用面でも大いに活用できる Redmine を利用し、これからもお客様への品質向上につながるソリューションを提供していきたい。