

## プロジェクト管理ツールとプロセス管理ツールの連携の実現

山本里枝子 吉田裕之

lisun@flab.fujitsu.co.jp

富士通研究所 ソフトウェア研究部

〒 211 川崎市中原区上小田中 1015

我々はソフトウェア開発環境におけるプロセスの位置付けとプロセスにより提供し得るサービスを検討し、プロセス管理ツールはそれと同じようなサービスを実現する既存ツールと関係を図るべきであるという立場を提案した。本稿はその提案に基づくプロセス管理ツールとプロジェクト管理ツールの連携の実現を報告する。品質管理ツール及び進捗管理ツールがプロジェクト管理ツールの二つの主要なコンポーネントである。品質管理ツールとの連携はその狙いを既定の品質管理作業手順の確実な運用とし、各種のプロセステンプレート・連携ツールから成る連携システムを示す。進捗管理ツールとの連携はその狙いを作業の並列化による工期の短縮とし、作業支援テンプレートとその作業方式を示す。

## IMPLEMENTATION OF COOPERATION OF PROJECT MANAGEMENT TOOLS AND PROCESS MANAGEMENT TOOLS

Rieko Yamamoto and Hiroyuki Yoshida

Software Laboratory ,Fujitsu Laboratories Ltd.

1015, Kamikodanaka, Nakahara-ku, Kawasaki, Kanagawa, 211 Japan

We have already proposed process management tools should cooperate with other existing tools which provide similar services. Based on the proposal, we implemented cooperation of process management tools and project management tools with data conversion . The object of the cooperation with the quality management tools is that staves can apply the quality management procedure surely. We implemented a cooperative system with process templates and tools. The object of the cooperation with progress management tools is shortening development term with concurrent works. Process templates and working methods are discussed.

# 1 はじめに

我々は、ソフトウェア開発環境におけるプロセスの位置付けとプロセスにより提供し得るサービスを検討し、プロセス管理ツールは同じようなサービスを実現する既存ツールと連係を図るべきであるという立場を提案した[1]。そこでは、プロジェクト管理ツールとプロセス管理ツールに求められている機能に共通点があることに着目した。既存のプロジェクト管理ツールの機能を活かすことによって重複開発を避けて、データ形式コンバータを経由した連携方式を提案した。プロジェクト管理ツールはプロジェクト全体を管理する諸機能を提供するが、プロジェクト内の様々な案件の一件ごとの状況を詳細に把握する機能は提供されない。プロセス管理ツールと連携することにより、各案件ごとの適切な管理を提供することが可能となる。プロセス管理ツールは各案件について、作業手順、進捗把握手段、フォーム（帳票）やツールなどの作業環境、等を提供する。また、各個人に対して全案件中で自分がやるべき作業の一覧（ワークリスト）が提供される。連携では、管理者側をプロジェクト管理ツールが、開発者側をプロセス管理ツールが主に支援する（図1）と考える。

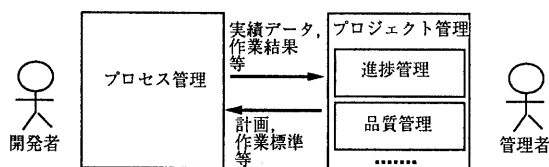


図1: 連携の概観

本稿は提案した方式の実現に関して、プロジェクト管理ツールの二つの主要なコンポーネントである品質管理ツール及び進捗管理ツールのそれぞれについて、連携の狙い、連携システムの実現と作業方式を報告する。プロセス管理ツールにワークフローシステムを利用する。「プロセス」は作業ステップのフローととらえ、各作業ステップは、フォーム、ロール（役割）を持つとする。フォームはその作業ステップの情報を表示するユーザインタフェースであり、ロールはその作業ステップを担当者の仮の名前である。また、定義したプロセスをプロセステンプレートと呼ぶ。プロジェクト管理ツールに、PMVIEW/Q[2]、PMVIEW/P[3]を利用する。

## 2 品質管理ツールの連携

品質管理ツールは、プロジェクトの品質目標を定量的に設定し、その目標にそってプロジェクトの品質データを収集し分析する。品質管理ツールは、品質データとして「何を集めるか」を規定しているが、「どう集めるか」については規定できない。プロセス管理ツールと連携することで「どう集めるか」について支援が可能とな

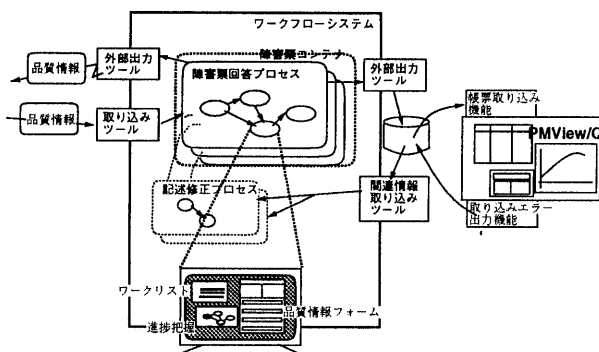


図2: PMVIEW/Q との連携

る。また、品質管理ツールはプロジェクト全体の管理を可能とするが、実際の品質データ収集作業からは直接起動されないで各件の正確な実績が把握できない。プロセス管理ツールと連携することで各件の作業状況が把握可能となる。逆に、プロセス管理ツールは各件の管理が細かくできる半面、プロジェクト全体での状況把握にはそのための機能を作りこむ必要がある。

以降では、品質管理ツールとプロセス管理ツールは互いを補完するように役割分担した連携の実現を示す。連携の狙いを既定の品質管理作業手順の確実な運用とする。品質管理ツールが、各品質情報の全体的な把握、分析に各種インタフェースを提供することを担当し、プロセス管理ツールは品質情報の収集の経路を記述管理、各件の実際のルーティングと実績収集を担当する。これにより品質情報収集の経路が記述管理されることで情報がもれなく収集されることを保証され、品質情報収集の各件ごとの進捗を容易に把握できる効果がある。

### 2.1 連携システム概要

品質管理ツール PMVIEW/Q は障害票、仕様変更票、検討項目票、レビュー票、QA票の各種品質情報を対象として、それらの帳票の帳票雛型（定型フォーマット）を提供している。帳票雛型に障害情報等を入力するだけで管理台帳、管理帳票、グラフが自動的に生成される。各帳票データは PMVIEW/Q とは独立なファイルとして存在している。PMVIEW/Q は帳票のレイアウト情報をもとに必要な情報をそれらファイルから取り込む。また、帳票を PMVIEW/Q に取り込む際に、その記述内容が指定フォーマット等に合致しているかチェックし、合致しない場合に読みとった情報とエラー内容を出力する。この出力をブルーリストと呼ぶ。

障害票をターゲットに PMVIEW/Q とワークフローシステムによる連携システムを実現した。連携システムの構成を図2に示す。障害票以外の帳票についても同様に実現される。

プロセス管理ツールでは、帳票管理作業を、帳票全

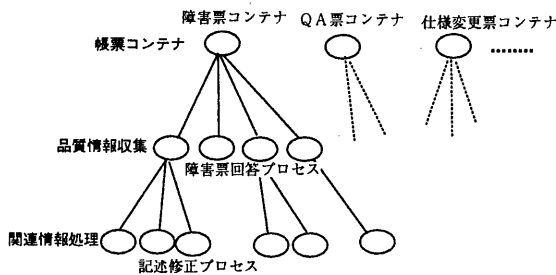


図 3: 帳票管理作業のプロセス構成

体の管理、各帳票の管理、各帳票に関連する作業の管理、の3階層で捉えて、以下(図3)を定義する。

**帳票コンテナ・プロセス** 対象帳票全体の管理。

**品質情報収集プロセス** 各帳票の処理作業の定義と進捗管理、各帳票に記入された品質情報管理。

**帳票各件の関連情報処理プロセス** 各帳票に関連する作業を支援。

連携システムのデータ変換は、品質情報収集レベルについては、プロセス管理ツールから品質管理ツール、品質管理ツールからプロセス管理ツールの各方向を、帳票データファイルを仲介としたデータ変換ツールで行なう。

**取り込みツール** 帳票データを解析しプロセス管理ツールが理解可能な形式に加工する。

**外部出力ツール** プロセス管理ツールが管理する帳票データをPMVIEW/Qが取り込み可能な形に加工する。

また、関連情報処理レベルについては、品質管理ツールからプロセス管理ツールへの以下の一方向のデータ変換ツールで行なう。

**関連情報取り込みツール** ブルーリストの内容を解析し、適切な障害票回答プロセスの下位プロセスとして、記述修正プロセスを作成する。

帳票コンテナレベルのデータ変換は、ツールとしては用意しないが、品質管理ツールの持つプロジェクト管理情報の一部を直接プロセス管理ツールが理解可能な外部テキストファイルへ持たせている。

## 2.2 品質管理作業プロセスの運用

プロセス管理ツール及び品質管理ツールは障害対処担当部署で運用されている。障害は担当部署の内外から報告される。作業全体は、「帳票回答/管理担当組織の外部/内部で帳票を作成し、外部作成されたなら、帳票

をプロセス管理ツールに取り込み、各担当者がワークリストやフォームを通じて品質情報を入力する。」となる。帳票は適宜、外部出力ツールを通じて品質管理ツールへ取り込む。以降は品質管理作業プロセスの運用方法をプロセスを中心として報告する。

### 2.2.1 帳票コンテナ・プロセス

各帳票毎に帳票コンテナ・プロセスを用意する。このプロセスの主な機能は、そのプロジェクトの全障害票の一括管理、品質情報収集プロセスの新規作成、障害票の運用に関して規定された役割(担当者)の定義、各帳票に一意にふられる管理番号の管理、である。障害票コンテナ・プロセスのフォームで、役割と担当者の対応、管理番号の管理機能等を提供する。これらの情報は、障害票回答プロセスが新規作成される際に受け継がれる。このように帳票全体を管理する手段に品質管理作業手順のプロジェクト毎の規則を実現することで、規則の着実な実行が保証される。

### 2.2.2 品質情報収集プロセス

障害情報を収集する障害票回答プロセステンプレートを図4に示す。楕円形は一つの作業ステップを表し、ステージと呼ぶ。ステージの上部には担当するロール、下部にはステージの名前が定義されている。ロールとはこのプロセスの参加者の仮の名前であって、実際にどの人物がそのロールを担うかは、プロセスの運用開始時点、あるいは運用中に決定する。図4で定義されるプロセスには次のようなロールが参加する。

**Submitter:** 障害票の起票者である。プロセス管理ツール運用部署の内部での起票の他、外部部署での障害報告を障害票として起票する。

**Project Leader:** プロジェクトの責任者。

**Tester:** テストの実施担当者。

**Programmer:** 障害対処の担当者。

**Checker:** 障害対処結果の障害票を障害報告部署へ返送する担当者。

実際にはこれらのロールのうちのいくつかを同一人物が担うことはありえる。

ステージの緑の小さい円は、各ステージのオプションである。オプションとは、各ステージの担当者のなんらかの判断を表す。オプションはそのステージを不活性(インアクティブ)化するもの(○)としないもの(●)がある。不活性化するオプションは通常は、そのステージの終了の宣言に相当する。例えば、障害票起票ステージで「S」と書かれたオプションは、「Submit」、障害票を提出したことを担当者が宣言することを意味する。

まず、障害対処部署外部からの障害票を受け付ける担当者、または、内部で障害を発見した者が、障害票コ



通知されるようにしておけば、レビューと次の作業を並列に進めることができる。また、一つの作業ステップが完全に終了する以前に、次の作業ステップの一部が開始可能となる場合には、そのことを次の作業ステップの担当者に通知することで、一部分を並列に進めることができる。さらに、いくつかの作業ステップがすべて終了していることを前提として行なう作業ステップが存在する(結合テストなど)場合に、すべての先行作業ステップが終了する以前に、すでにいくつかが終わっている段階で一部分を先行して実行できる可能性がある。

以降は、豊富なプロセス記述能力と動的なプロセス変更機能を持つプロセス管理ツールを利用した、並列度の高い開発プロセス運用方法の検討結果を報告する。

### 3.1 進捗管理ツールによるスケジュールの策定と管理

図6は進捗管理ツールによって策定される、典型的な開発スケジュール(の一部)である。

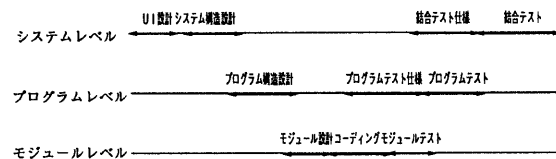


図6: 進捗管理ツールによる作業スケジュール

ここでは開発されるシステムは複数のプログラムからなり、各プログラムは複数のモジュールから構成されることを想定している。図6の三つのレベルは、プロダクト構成の三つのレベルに従っている。図6では、複数あるプログラムレベル、モジュールレベルの作業スケジュールを一本で集約して表現している。この場合、実績は各作業ステップごとの進捗率(予定開発量に対する実開発量の比率、予定工数に対する実工数の比率など)によって表示される。したがって全体の進捗把握には向いているが、遅れが出ている場合に具体的にどの作業が問題になっているのか、どの程度の手戻りが発生しているのか、先行して開始できる後工程は何か、を読みとることはできない。プロセス管理ツールを進捗管理ツールと併用し連携させれば、これらの問題点を解決できる。

### 3.2 連携システム概要

プロジェクト管理ツール PMVIEW/P は、システム開発標準 SDEM90 に基づく工程管理を支援するツールである。スケジュール編集機能と進捗の入力と表示機能を提供する。スケジュールは、提供されるスケジュール・テンプレートをカスタマイズすること編集が容易となる。また、自分たちが作成したスケジュールをテンプレートとして登録できる。

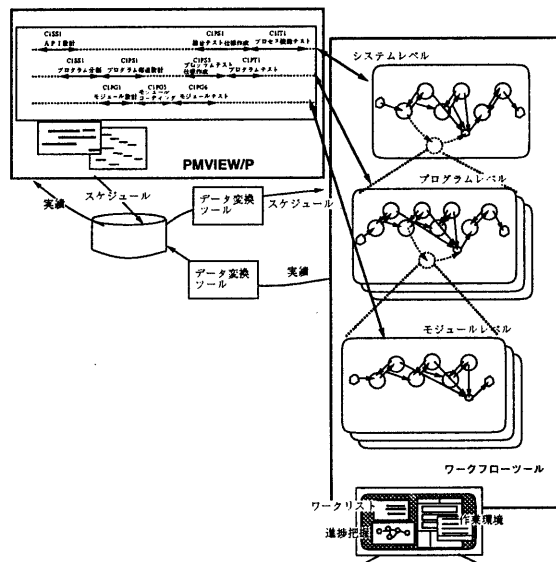


図7: PMVIEW/P との連携

まず管理者が PMVIEW/P でプロジェクト管理データとスケジュールを作成する。次にスケジュールデータをプロセスデータに変換し、必要なカスタマイズを行なう。プロセスデータを基にプロセス管理ツールのワークリストにより各個人に作業指示がされ、さらにワークリストを選択してその作業を支援する環境起動することで個人作業支援が行なわれる。作業実績が自動収集される。管理者はプロセス管理ツールが収集する実績を PMVIEW/P が提供するインターフェースで確認する。(図7)

### 3.3 開発作業プロセスの運用

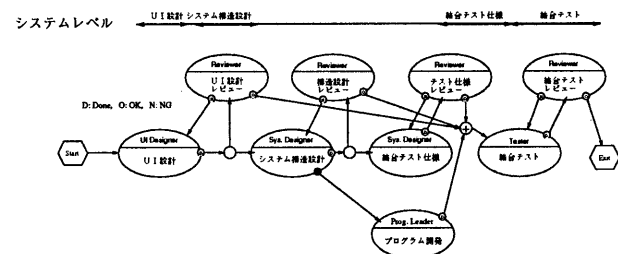


図8: システムレベルプロセス

システムレベルプロセスを例にとって、その運用方法を検討する。図8はシステムレベルのプロセスのテンプレートである。これはトップレベルの作業プロセスであり、ここでは一プロジェクトについて、このプロセスが一つだけ運用されることを想定する。中央の四つのステージが、スケジュール線に現れる作業ステップと対応している。また、ここではそれぞれの作業に対して必ずレビューが行なわれるように、手順が詳細化されてい

る。図8で定義されるプロセスには次のようなロールが参加する。

**Owner:** このプロセスを起動し、他のロールの具体的な担当を決定し、場合によってはプロセスを動的に修正して問題に対処する。

**UI Designer:** ユーザインタフェース設計の担当者。

**System Designer:** システム構造設計及び結合テスト仕様設計の担当者。

**Tester:** 結合テストの実施担当者。

**Reviewer:** 各作業ステップのレビュー担当者。

**Program Leader:** プログラム開発の責任者。

システムレベルプロセスは典型的にはプロジェクトマネージャによって一つだけ起動される。プロジェクトマネージャは、このプロセスの Owner として、まず各ロールの実際の担当者を決定する。すると、最初のステージである UI 設計ステージの担当者に作業開始の指示が出され、作業プロセスの運用が開始される。プロジェクトマネージャは、このプロセスをモニタし、問題が起きた時にどのように対処するかを決定する責任を持つ。プログラムレベルプロセスは、このシステムレベルプロセスの下位プロセスとして、プログラム開発ステージにおいて起動する。したがって、プログラムレベルの各プロセスの監督責任は、Program Leader に委譲される。Program Leader はその権限に基づいて、複数のプログラム開発の間で各開発担当者の負担のバランスを調整することができる。

図8では Program Leader の作業着手時期を判断するのは System Designer になっている。ただし、構成要素であるどのプログラムについて作業に着手して良いかは明確ではない。これを明示するためには、プログラムごとにオプションとそれによって活性化されるステージを独立させ、System Designer が個別に作業着手 OK の指示を出せるようにすることが考えられる。

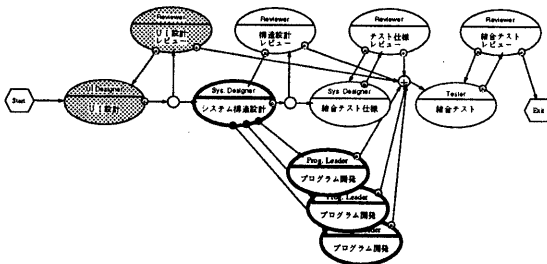


図9: プログラムごとに作業着手を指示するプロセス

図9は、開発システムが三つのプログラムから構成される時に、System Designer がそれぞれのプログラム

の外部仕様が確定した時点で個別に開発開始の合図を Program Leader に指示できるように、拡張したシステムレベルプロセスである。こうすることによって、Program Leader は最初の指示の際にどのプログラムに着手できるか明確に分かるだけでなく、第二、第三の指示を受けとることができ、二つ目以降のプログラムに着手可能となったにも関わらずにそれに気づかないで無駄な待機時間を過ごすことが無くなる。

ただし、開発システムがいくつのプログラムから構成されるかは、システム構造設計によって決定するので、このようなテンプレートをあらかじめ用意しておくことはできない。プロセス管理ツールにこのように個別のプログラムの外部仕様が決定されるごとにプロセスを拡張していく方法の実現手段は、現状では今後の課題となる。

プロセス管理ツールは、タイムリな作業指示を自動化できるので、開発作業全体の並列化を促進し、工期の短縮を計ることが可能である。ここで提案したプロセステンプレート及びその運用方法によって、次のような工期短縮効果を期待できる。プログラム、モジュールの開発作業を個別に把握でき、それらの間で開発者の負担のバランス調整が可能である。各ステージの担当者にタイムリに作業開始指示が通知されるので、無駄な待機時間が短縮される。レビュー作業を並列に実行できる。構造設計の終了前に、個別のプログラム / モジュールの開発着手を指示できる。

#### 4 まとめ

本稿では、統合 CASE 環境におけるプロジェクト管理ツールとプロセス管理ツールの連携を踏まえた、品質管理作業と開発作業を示した。今回はプロジェクト情報等をデータ変換によるデータ共有により実現したが、本来はリポジトリでの一元管理を理想とする。今後は、ソフトウェアプロダクトの管理を含めた開発作業全般に渡るプロセス支援に関して既存ツールと連携を図り、CDIF[4]等のような CASE ツール間のソフトウェアの仕様や構造のデータ変換の標準化動向を踏まえながら検討を進めていく。

#### 参考文献

- [1] 山本里枝子, 近藤竜生, 吉田裕之. ソフトウェアプロセスに基づくソフトウェア開発環境の検討. 情報処理学会 ソフトウェアプロセス・シンポジウム, pp. 1-10, June 1994.
- [2] 富士通. プロジェクト管理支援ツール 利用者マニュアル 品質管理システム V2, September 1993.
- [3] 富士通. プロジェクト管理支援ツール 利用者マニュアル 工程管理システム, December 1993.
- [4] 篠木裕二, 西尾高典, 吉川彰弘. Cdif - case データ交換形式. コンピュータソフトウェア, March 1993.