idea in: プロジェクト管理を効率化する Web アプリケーション その 2 - タスクチャート手法の提案

浜田真成[†1]　安藤大地[†1]　笠原信一[†1]

本研究は，専門知識や難しい操作を必要とせずに扱えるユーザーフレンドリーなプロジェクト管理支援 Web アプリケーションの開発と，そのシステムの実現を目指すものである．今期研究では，より幅広いプロジェクトに普及させるための導入手法，およびユーザーインターフェースについての検討と，プロジェクトにおけるタスクをベースにバージョン管理の手法を応用した「タスクチャート手法」の提案を行う．

Idea In : A Web Application For Project Management Part2 – Task Chart

MASANARI HAMADA[†1]　DAICHI ANDO[†1]　SINICHI KASAHARA[†1]

This study proposes user-friendly project management system without specialized knowledge and difficult operation, researches the way of efficiency of collective work in the process of project management, and aims to build this system by developing the web application software. In this report, we consider the introduction method to disseminate a wider range of project managements, then suggest the "Task Chart" method applying the distributed revision control system into a variety of files in the projects.

1. はじめに

前回の報告[1]では，分業における情報共有が困難となり易い点，Web上におけるチーム作業が増加している背景[2][3]から，こうしたユーザーレベルのコミュニティにおいて，幅広く導入されるWebアプリケーションの提案と制作を行った．
アプリケーションでは，問題点や現状を踏まえ，プロジェクトにおけるファイル管理をチャート形式で俯瞰的に表示できるファイル管理システムを提案し，実際にサービスサンプルを展示運用したうえでフィードバックを行った．

その成果として，異なる分野間でのファイルのやり取りを一望できる管理システムによって，管理の効率化に一定の有用性を提示することができた．特に，画像ファイルを扱うことの多いプロジェクトにおいては，チームの作業内容が可視化された形で管理することが可能となり，効率的にプロジェクトを進行することに一定の成果があった．

しかしながら，こうしたファイルをベースにした管理手法だけでは，様々なプロジェクト形態での有効性を示すことは難しく，より現実に沿った提案・機能拡張を行ったうえで改善を図る必要がある．

ファイルをベースにした管理で特に問題となるのは，様々なタイプのプロジェクトに応用できないことである．プロジェクトには，話し合いや各チームの進行をベースに行うチームも存在するため，ファイル管理だけでは対象の利用者を狭めてしまっている．また，簡易的な時系列による管理を行っているが，ファイル数の多いチームのみが突出する場合において，全体の進歩状況が見えにくくなることがわかった．

一方で，操作方法の分かり易さについてもより踏み込んだ検討が必要である．ファイルを自由に配置できることが本アプリケーションのメリットである反面，多人数でのファイル整理が困難になるといったケースも見られたためである．

また，こうした新規のシステムや専門手法の導入をどのような形でユーザーに提案するのかは，分かり易いシステムを構築する上で重要な検討事項である．

今期における研究では，これらを踏まえ，プロジェクトにおける「タスク」をベースにバージョン管理の手法を取り入れた管理ソフトウェアの提案と，より幅広いユーザーが利用できるようにするための導入手法・分かり易いユーザーインターフェースについての検討を行う．

2. 現状の制作フローと問題点

2.1 現状の制作フロー

現在，プロジェクトの管理はタイプによって，様々な制作フローが用いられているが，ユーザーレベルの制作コミュニティで多く行われるソフトウェア開発，プロジェクト，作品制作の分野での管理方法について検討を行った．

(1) ソフトウェア開発における管理方法

Git[4]（ギット），Mercurial[5]（マーキュリアル），Subversion[6]（サブバージョン）などのシステムに代表される分散バージョン管理の手法は，プログラムのソースコードなどを修正差分として管理することで，制作しているプログラムの経過管

†1 首都大学東京システムデザイン学部
Tokyo Metropolitan University

理や多人数での製作を容易にする管理システムである．そのため，オープンソースのソフトウェア開発にも有効に用いられている．

また，フォークやマージの概念によって，コードの構築や管理を並列的に行ったり，他の改変を自由に取り込むといった手法によって管理することができる．

(2) プロジェクトの管理方法

管理が重要になるプロジェクトにおいては，プロジェクトの進行状況を客観的指標によって管理する手法が一般的に用いられる．プロジェクトで行うべき課題をタスクごとに分割し，目標とするべき指標（マイルストーン）などを定めることによって，計画的なプロジェクト進行を行う．

また，プロジェクトの到達度を，価格などの数値基準（EV）によって定量化することでスケジュール管理を行うアーンドバリュー評価や，プロジェクトがどの程度計画通りに進んでいるのかを可視化するためのバーンダウンチャートなどを用いることが多い．これらの指標を用いながら，計画を見直すなどを繰り返し，最終的な目標に到達させてゆく手法である．

(3) 作品制作における管理方法

作品の原案から完成までを納期までに仕上げるために，メンバー同士でスケジュール管理しながら作品制作を行っていく方法が一般的である．また一連の作業では，作業途中の様子や，途中まで制作したファイルを随時，他のメンバーや依頼主とやり取りを行い，その都度，修正箇所のアドバイスや，今後の方針を判断していく方法が取られている．

2.2 既往ソフトウェア

(1) 専門的なソフトウェア

主に大規模または専門的なプロジェクトにおいて導入が行われるが，一般の小規模プロジェクトでは，ほとんど導入が検討されない．しかし，タスクベースの管理など，長期の視点として見た場合には，どのプロジェクトにおいても便利に利用できる手法がある．

(2) 簡易的なソフトウェア

小規模プロジェクトにおいて導入される傾向がある．利用例としては，メーリングリストによる手法や，カレンダーによるスケジュール管理，エクセルやワードをベースにしたログ管理，ストレージサービスとメールによる連絡を併用するなどが一般的で，普段から利用しているアプリケーションの+αとして用いられる場合が多い．

2.3 問題点

Web 上における創作活動は大幅に増加しているにもかかわらず，ユーザーレベルの制作プロジェクトにおいては，現在管理ソフトウェアの導入が余りなされないのが現状である．

その原因として，以下がある．

(1) 学習コストによる導入障壁

ライトユーザーの多くは，こうしたソフトウェアで取り入れられている専門的手法に馴染みが無いため，導入に際しては新規に手順や用語を覚えるための「学習コスト」が発生することになる．学習コストは事前知識となるものであり，共同作業となる場合は，全員が管理ソフトウェアの手順や用語を前提として持っていなければ，プロジェクトを進行していくことはできない．そのため多人数の場合は，各自が学習を行うことになるが，プロジェクト参加者の増加に伴って，結果としてプロジェクトあたりにかかる学習コストは増大する．

また，PC 操作に不得手な参加者も少なからず存在するが，その場合，導入障壁はより高いハードルに見えている場合がある．こうした管理ソフトウェアの特徴や印象から，プロジェクト進行のオーバーヘッドとなると判断された場合は，結果として管理ソフトウェアは導入されないことになる．

(2) プロジェクトタイプの違い，担当分野の違いによる管理手法の乖離

プロジェクトはタイプによって制作フローが大きく異なり，作品制作の完成を目的にするもの，話し合いがベースとなるもの，ドキュメントベースで決定が行われるものなど，それぞれのタイプで管理手法も様々である．コスト管理を伴うプロジェクトでは，リスク管理やテストといったマネジメントが必要になるケースもある．

担当分野間の違いでも制作フローは異なり，例えばデザイナーとコーダーでは制作物の管理方法が一般的に異なるものである．ある程度の規模になると，コーダーは主にバージョン管理システムや，それに伴うプロジェクトマネジメントの機能取り入れた管理を行う場合が多いが，デザイナーの場合は画像・映像といったバイナリファイルの制作を主に行うため，こうした管理手法を用いず，主に納期を重視した管理を行う場合が多い．

このような管理手法違いから，同じプロジェクトであっても異なる管理ソフトウェアを用いる場合や，こうした管理ソフトウェアを導入しない場合が多いのが現状である．そういった場合，全体のプロジェクト像を他分野から俯瞰的に把握することは困難となり易い．

2.4 改善検討

(1) 専門手法の印象の改善

上記に挙げた専門的手法を，なじみのないユーザーにとっても分かりやすく導入させるためには，直感的に扱えるユーザーインターフェースが非常に重要になる．

そのために考えられる手法としては，

i) 専門機能を分かり易い表現で実現する

ii) 専門機能をそのまま載せるのではなく，類似した別機能に置き換える

の 2 点が考えられる．

i では，専門用語や難しい操作を簡単な表現に置き換えることで，わかりやすさと専門機能を両立させる．ii では，簡素化など別のアプローチをとることで，専門機能の概念のみを実装する方法である．

(2) **閉鎖的にしない**

また，メールなどの簡易的なソフトウェアを利用する者層の使用シーンを考えると，閉鎖的にしないことが重要である．既存のサービスとの連携を可能にすることで，利用者の間口を広げることが可能になるためである．

3. システムの提案

3.1 タスクチャート手法

2.1 で述べた制作手法を踏まえると，作品制作におけるフローも，ソフトウェアの制作手法と同様に，バージョン管理を経て作業が進行していることがわかる．制作物の進行状況を随時，閲覧，コメントすることが出来れば，他チームが作業内容の把握や修正を効率的に行うことが出来る．そのため，こうした作品制作におけるフローにおいても，進行状況を管理するバージョン管理の手法が有用であると考えた．

そこで著者らは，これまで行ってきた手法に，Git などに代表されるバージョン管理の手法を，様々な管理に応用した「タスクチャート手法」を提案する．

タスクチャート手法の概要は以下のとおりである．

i) プロジェクト全体のタスクをチームごとに俯瞰的に見渡せる．（図 1）

ii) タスク(図 2)を選択することで，そのタスクに関連するファイルや議論の過程を即座に閲覧することが出来る．また，それらの議論に参加したり，ファイルを追加したりすることが出来る．

iii) バージョン管理におけるフォークやマージといった概念を取り入れ，コードだけでなく画像や議事録といった，プロジェクトの制作ファイル，思考の過程を，簡易的にバージョン管理できるようにする．

iv) タスクに簡易的な重要度と，期限を設定し，達成度を視覚化して表示する．

また，これまでは「ファイル」をベースに管理を行う手法であったが，様々なプロジェクト内容に対応するため，プロジェクトにて行うべき課題内容である「タスク」を単位に管理する手法に変更を行った．

3.2 機能概要

また，これまでと同様，プロジェクトの全体像，今どのチームで何のファイルが作られているのか，何が話し合われているのかといった，チーム作業において分散しやすい情報を一元的に管理することが出来るチャート形式のユーザーインターフェースを採用する．

これ以外に，実用的に利用できるようにするため，プロジェクトマネジメントの手法を用い，管理ソフトウェアに以下の機能を実装する．

(1) **プロジェクトの進行状況の管理**

タスクに簡易的な重要度と期限を設定したことによって，達成度の算出が可能となるため，これを基にした進行状況の管

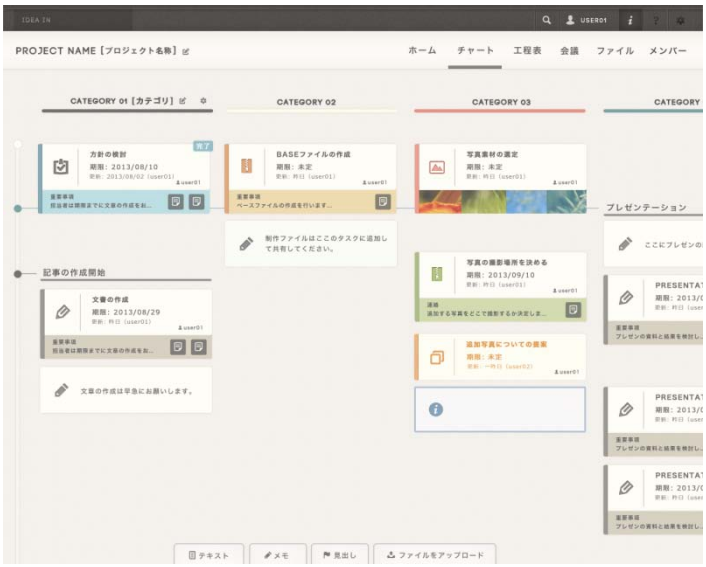

図 1 ideain – タスクチャート手法によるタスクの俯瞰表示

図 3 ideain – 各オブジェクトはプロジェクトのタスクを表す

図 2 ideain – 全体図（参考）

理を行う.

(2) **工程表**

進行状況は，一般に浸透している，簡易的なガントチャートによる日程管理機能によっても管理を行い，ファイル管理，タスク管理との連携を図る.

(3) **メンバー/チーム管理**

プロジェクトにおける人材の管理を行う機能．前回の報告において，作業の混乱が見られた点を踏まえ，メンバーごとに権限の設定を行えるようにすることで，編集の混乱を防ぐ.

(4) **連絡・会議**

メールによる情報交換では，情報が分散してしまうため，こうした情報を扱うことが出来るプラットフォームによって情報の統合を図る.

以上により，様々な分野の集まるプロジェクトに必要となる一連の作業を，一つのプラットフォームで行えるようにする.

4. システムの実装と技術的課題

管理ソフトウェアでは，多人数が同時に操作した際にも円滑に利用できることが重要となる．そのため，常に同期的に作業内容を表示させることが望ましい.

こうした同期的システムの構築のために，前回の報告では，Webアプリケーションの構築の際, Ajaxを用いた更新システムを実装した．しかしながら，サーバーへの負荷と同期するまでのタイムラグが発生することになるため，稼働時に適切な処理が行われない場合が発生した.

これを改善するため，今期研究では，Ajax通信の代替としてWebSocket[a]を利用したシステムの構築を行い，よりシームレスなアクセスを可能とした．また，それに伴いWebSocketシステムの構築のしやすさから，開発言語をPHPからNode.jsへと移行を行った.

5. おわりに

以上の機能により，タスクチャート手法を利用したプロジェクト管理アプリケーションについて述べた.

今後の展望として，どの程度機能性や効率性が向上するのかを図るため，具体的なプロジェクトでテスト運用を行い，本システムの妥当性について検討しようと考えている．そのため，早期の実現と，オープンベータテストの稼働を行い，フィードバックを踏まえた改善を図っていく予定である.

a) WebSocket は，ブラウザとサーバー間で双方向通信を可能にする通信プロトコル技術の一つで，応答のためにサーバーへリクエストを送ることなく、イベントドリブンなレスポンスを受信できる

参考文献

1) 浜田真成(2012.11) idea in: プロジェクト管理を効率化する Web アプリケーション提案と制作

2) ニールセン株式会社 イラスト特化型 SNS の「pixiv」の利用者数が 390 万人を突破（2011.8） : http://www.netratings.co.jp/news_release/2011/08/31/Newsrelease083120 11_J.pdf

3) ブログ・ＳＮＳの経済効果の推計 総務省情報通信政策研究所調査研究部 (2009.7) : http://www.soumu.go.jp/main_content/000030547.pdf

4) Git : http://git-scm.com/

5) Mercurial : http://mercurial.selenic.com/

6) Apache Subvirsion : http://subversion.apache.org/

7) Google Code : http://code.google.com/

8) CodePlex : http://www.codeplex.com/

9) Microsoft Project : http://www.microsoft.com/japan/project/default.aspx

10) dotProject : http://www.dotproject.net/

11) 5pm : http://www.5pmweb.com/

12) Backlog : http://www.backlog.jp/

13) Brabio! : http://brabio.jp/

14) Zoho Project : http://www.zoho.jp/projects/

15) Google Groupe : https://groups.google.com/

16) Facebook : http://facebook.com

17) ChatWork : http://www.chatwork.com/

18) Fluent : http://fluent.io/

19) サイボウズ Live! : https://live.cybozu.co.jp/

20) 西村克己(2000.7)「よくわかるプロジェクトマネジメント: 入門マネジメント ＆ ストラテジー」，日本実業出版社

21) 内館町子(2008.4) ひと目でわかる Microsoft Office Project 2007，日経 BP ソフトプレス

22) Project Management Institute, inc.(2004) A Guide to the Project Management Body of Knowledge (PMBOK Guide)

23) Stoyan Stefanov(2010.9) “JavaScript Patterns Build Better Applications with Coding and Design Patterns” O'Reilly Media

24) WebSockets | MDN : https://developer.mozilla.org/ja/docs/WebSockets