

ソフトウェア開発 PBL への チケット駆動開発の適用による共同作業の改善

木崎 悟* 丸山 英通* 土屋 陽介* 中鉢 欣秀*

Improvement of collaborative work by application
of ticket-driven development with software development PBL

Satoru Kizaki* Hidemichi Maruyama* Yosuke Tsuchiya* Yoshihide Chubachi*

本論文では、はじめに2010年度の慶應義塾大学の大学学部生を対象とした産学協同プロジェクト「コラボレイティブ・マネジメント型情報教育」で筆者（木崎）が参加したプロジェクトについて述べる。このプロジェクトでは、タスク管理やチーム内のコミュニケーションに問題があった。その経験を踏まえ、チケット駆動型によるリアルタイム性を重視した共同作業をPBLに適用することによりこれらの問題の解決を図る。

In this thesis, we speak about a project that was the university-industry collaboration project "Collaborative Management Laboratory", intended for undergraduate students of the Keio University in fiscal year 2010. This project had problems of the task management and communications within team was a problem in this project. We proposed to focus on the real-time attribute of collaborative work under the ticket-driven development.

KeyWords&Phrases : アジャイル開発, チケット駆動開発, PBL, タスク管理

Agile software development, Ticket driven development, PBL, Task management

1. はじめに

本論文では、2010年10月～2011年2月までの約4カ月間で実施された慶應義塾大学SFCのPBL (Project-Based Learning/Problem-Based Learning) 型教育の講義科目「協創型ソフトウェア開発」[1]でのアジャイル開発を採用したプロジェクトについて述べる。2010年度で起きた問題の改善を提案する。

この講義は、2005年度秋学期に開講され、2010年度で5回目である。毎年、複数の開発チームが結成され、プロジェクトのメンバは3～5名(学部1～4年生)で構成される。また、プロジェクトマネージャ(以下、PM)は若手の企業人や情報系を専攻する大学院生が担当する。プロジェクトには、実際にシステムを必要とする企業や個人の顧客が設定され、ユーザに利用されるソフトウェアの開発を目標としている。また、教育面ではPBL実施により、情報システム技術者の育成を目的としている[2][3]。

プロジェクトの進行はPMに一任される。プロジェクトが成功すれば、どのような開発手法を選

択してもよい。しかし、開発期間が短期間(約4カ月間)であるため、効率的に成果を出すことが求められる。開発手法の選択肢として、迅速かつ適応的にソフトウェア開発を行うアジャイル開発は有効的に考えられた。アジャイル開発手法はエクストリーム・プログラミング(以下、XP: Extreme Programming) [4]やスクラム (Scrum) [5]など数多く考案されている。

日本国内のアジャイル開発では、アジャイル開発手法全体を適用するよりは、個別事情に応じて手法を形成するプラクティスを選択的に活用している事例が多く見られる[6]。本プロジェクトにおいても使いやすいプラクティスを採用して実践した。しかし、PMによる進捗管理やタスク管理、チーム内でのコミュニケーションなどに問題があった。そして、顧客の評価を得られなかったことや、チームの目標達成とならなかったことから、プロジェクトは失敗した。

これらの問題を解決するためにアジャイル開発との親和性が高いチケット駆動開発 (TiDD: Ticket Driven Development) [7]を適用とコミュニケーションツールをしたリアルタイム性重視の共同作業を提案する。

本論文の構成は以下の通りである。まず、第2章でプロジェクトの概要を説明する。次に、3章でプロジェクトの成果について、4章でプロジェ

*産業技術大学院大学 (AIIT: Advanced Institute of Industrial Technology)

クトの問題点について、5章で共同作業の改善提案について述べる。

2. 本プロジェクトの概要

本章では、2010年度のプロジェクトの概要について述べる。チームの結成は、2010年10月に行われ、筆者（木崎）はプロジェクトのPMとなった。参加したメンバは3年生の学生A、2年生の学生B、1年生の学生Cである。3名ともプログラミング経験があり、Java文法やオブジェクト指向を理解しているなどのスキルがある。

本プロジェクトでは、現状のシステムの改修をすることにより、既存のユーザ数を延ばすことを目的とする。

カタゾウ[8]は、プリクラに似たカメラアプリである。例えば、京都にいかずとも、京都のカタを使って写真を撮るなどが可能であり、そのカタはユーザ投稿型である。

この現状のシステムを分析し、問題点の抽出を行ってから、追加機能の実装をすることで改修する。また、開発対象はAndroid端末、Webサイト（Ruby on Rails）とした。

顧客とのミーティングはPMだけが行い、PMが学生に作業を指示することによりメンバは開発に集中できる環境を整えた。

本システムの目標はユーザ数を増やすことにあり、Android Market[9]やアンドロナビ[10]での公開を目指した。

開発プロセスはXPを基本として、ユースケース駆動開発と組み合わせ、各プラクティスを採用した。開発ツールはEclipseを使用し、Subversionによるバージョン管理を行った。

プロジェクト管理ツールはExcelを使用し、WBS（Work Breakdown Structure）とEVM（Earned Value Management）による進捗管理を行った。

3. 本プロジェクトの成果

本章では、プロジェクトの成果について述べる。「カタゾウ」のAndroidアプリとWebサイトに関して改修を行った。

3.1 ソフトウェアの規模

本プロジェクトにおけるソフトウェアの改修規模を表1に示す。改修範囲はAndroidアプリとカタゾウのWebサイトである。

Androidアプリの追加機能は、Twitter投稿機能である。アプリケーションの撮影画面には「Twitterでつぶやく」ボタンが追加されている。本機能は、

学生Bが担当した。

次に、Webページの改修である。本改修では、トップページのレイアウト変更と画像検索機能の追加、Facebook連携機能の追加を行った。なお、レイアウト変更と画像検索機能は学生Cが担当し、Facebook連携は学生Aが担当した。

表1 ソフトウェアの改修規模

カテゴリ	言語	行数 ※ () は改修前
Android	Java	7570 (6902)
Web	PHP	206202 (206044)
追加機能		内容
Twitter連携		カタゾウのAndroidアプリからTwitter投稿を可能にする
レイアウト変更		Webサイトに表示する画像を増やす
画像検索		キーワードを入力して画像の検索を行う
Facebook連携		WebサイトとFacebookを連携させて、WebサイトからFacebookへの投稿を可能にする

3.2 活動時間

本プロジェクトにおける活動時間を表2に示す。PMの実働時間が148時間と大幅に大きくなっており、学生Aが比較的時間をかけているが学生Bと学生Cは、本講義にかける時間が少ない。

本講義は、全部で4チーム実施されていたがメンバが途中で離脱してしまったプロジェクトも存在する。その理由は、次章で述べる。

表2 マネジメントデータ（単位：時間）

対象	PV	AC	EV
PM	127.0	148.0	120.5
学生A	127.5	123.5	99.0
学生B	74.5	40.0	34.5
学生C	72.5	55.5	57.0

3.3 アジャイルプロセスの適用

本プロジェクトでは、アジャイルプロセスの一種であるXPを適用した。XPでは12~14個のプラクティスを提唱している。

XPのプラクティスの内、適用可能であったプラクティスと適用することが困難であったプラクティスがある。以下は適用可能であった。

- ・小さなリリース
- ・コードの共同所有
- ・計画ゲーム
- ・シンプルな設計

・メタファ

開発期間の4カ月の内、本プロジェクトは反復計画を立て機能毎の2回のイテレーションによるリリースを行った。

また、Subversionによるソースコードのバージョン管理を行いチーム内で共有した。

顧客とは、週1回の対面ミーティングをPMが行い要件定義において改修提案内容のストーリーを作成し、顧客に優先順位を付けていただいた。これはXPの計画ゲームにあたる。

XPでは最低限のシンプルな設計をプラクティスとするが、基本設計書、画面設計書のみ仕様書として作成した。

本プロジェクトではユースケース駆動開発を採用したため、基本設計書の内容はシナリオ及びユースケース記述、ユースケース図である。また、画面のイメージを顧客と共有するために画面設計書を作成した。詳細レベルの設計は行わずに実装者の判断に任せた。

また、場所、時間、人的な制約により適用するのが難しいと判断するプラクティスは次のようになる。

- ・オンサイト顧客
- ・ペアプログラミング
- ・オープンワークスペース
- ・最適なペース
- ・リファクタリング
- ・テストング
- ・コーディング規約
- ・継続的インテグレーション
- ・日ごとのスキーマ移行

適用することが難しいプラクティスについて主要なものにつて、以下に理由を述べる。

(1) 場所による制約

2人でレビューしながらコーディング作業を行うペアプログラミングは、空いている時間帯はそれぞれ異なる。そのため、共同作業することが困難であると考ええる。

次に、オープンワークスペースは設計に関するディスカッションがすぐに行える場所を必要とする。作業場所はメンバ宅、または、メンバが所属する研究室であるため適用することが難しかった。

(2) 時間による制約

フルタイムで労働する社会人ではなく、他の講義、研究会、サークルなどがあり、学生が本プロジェクトに割く時間は週平均5~10時間である。

(3) 人的な制約

プロジェクトに参加したメンバは、開発で用いられていた言語 (Java, Ruby) に関して経験が乏しかった。そのような状態でリファクタリングやテ

ストコードの作成を依頼すると時間的な制約により厳しいと判断した。

(4) その他

既存システムの改修であったが、元の開発元にコーディング規約が存在しなかった。既存のコードと合わせる形でコーディングを行った。

また、顧客とのミーティングは、顧客の都合により、ミーティングを行えない週もあった。

4. 本プロジェクトにおける問題点

本章では、プロジェクトの問題点をPMとメンバの評価から述べる。

4.1 PMの評価

PMの観点から、致命的であった品質とマネジメントに関して評価する。

(1) 品質に関する問題

本プロジェクトにおいて、Androidアプリ・Webサイト双方にソフトウェア品質に問題があり、リリース延期やリリースができない自体を招いた。テストングのケース不足とメンバの経験不足を原因と考える。

(2) プロジェクトマネジメントに関する問題

顧客と打合せをして、計画を立ててもそれを実施するメンバが作業をしなければ意味がない。本プロジェクトでは、メンバの稼働状況の不足をPMが補ってしまいPMに対して、二重に負荷が掛る結果となった。

また、アジャイル開発において、Excelシートによる進捗管理の難しさがあった。アジャイル開発の場合、要件の変更は常に発生する。また、経験が浅いPMにとって、正確な見積りを出すことが難しく、WBSの書きかえとリスケジュールが頻繁に生じた。

次に、割り当てたメンバの進捗の遅れがあった。学生は、本プロジェクトの他にも講義やサークル活動などのタスクを有している。また、本プロジェクトは企業での勤務のように拘束力がない。そのため、作業時間は学生のモチベーションが大きく影響した。

さらに、情報共有に問題があった。毎日集まって進捗状況の確認をすることが不可能だったためコミュニケーションが取り難かった。オンラインミーティング (Skype) を行うことにしていたが、全員が揃うことがほとんどなかった。

そのようなことから、意思伝達に遅れが生じて、リリース直前での不具合の発見に対して、即座に対応することができなかった。そのため、開発期間のメンバの行動が見えなかった。作業の「見え

る化」は必須事項であると考える。

4.2 メンバからの評価

このプロジェクトを実施するにあたり問題となったことが何点かある。プロジェクトにおいて学生Bが担当した機能についてリリースすることができなかった。

リリースできないということは、顧客も評価できずに失敗ということである。また、学生Cも、本プロジェクトに割いた時間が短く、PMが期待した結果を得ることができなかった。

学生B・C共にソフトウェア開発を行う上でのプログラミングの知識や経験を有していた。しかし、十分な成果を出すことができなかった。その理由について、学生B及び、学生Cは次のように述べている。

学生Bは、第1の問題点として、グループワークにおける人との連携が苦手であり、最大の原因はPMとのコミュニケーションの齟齬や、報告・連絡・相談の怠慢であったとしている。

第2の問題点として、顧客と企業のサービスの完成度の違いを把握しきれなかったことを上げている。エラーチェックやテストが完璧でないために企業が提供するソフトウェアとしての基準を満たすことができなかった。

また、学生Cによるプロジェクトの評価は、プロジェクトが失敗した問題点として興味深い。

プロジェクトを成功させるには、個人のモチベーションが高くなければならない。そのためには、プロジェクトが面白いものである必要がある。内容が面白いほどチームのモチベーションは上がり、結果として良い方向に進んでいくと述べた。

そして、プロジェクトにおいてPM主導でプロジェクトを行っていたが、口出しを出せる体制にしてほしかったとしている。

以上のプロジェクト実施後の評価より、プロジェクト内で、コミュニケーションを円滑に取ることと、メンバのモチベーションを維持させることがプロジェクト成功への鍵である。

4.3 問題点

4.1と4.2の評価の結果、以下の項目に問題があることがわかった。

- ・ソフトウェア品質
- ・プロジェクトマネジメント
- ・コミュニケーション
- ・モチベーション

以上の問題を改善することで、プロジェクトを成功に導くことができる。

5. 共同作業の改善提案

本章では、プロジェクトにおける問題点である。アジャイル開発を適用する上でのソフトウェア品質の問題、プロジェクトマネジメントの問題、メンバとのコミュニケーションやチームのモチベーションの問題を解決するために、チケット駆動開発を適用した情報共有の方法について提案する。

5.1 チケット駆動開発の適用

チケット駆動開発 (TiDD : Ticket Driven Development) とは、2007年にまちゅ氏が、バグトラッキングシステム (以下、BTS : Bug Tracking System) のチケットを用いて開発プロセスを改善したことに始まる。チケット駆動開発を採用することで、関係者で情報を共有でき、担当者ごとの集計も可能で、誰がどのような状況であるかが分かり、作業負荷を平準化できる。

チケット駆動開発の先駆者である阪井誠氏は、ソフトウェア開発は時間との闘いであり、ソフトウェアに対する多様な要望に答えるにはアジャイル開発やチケット駆動開発といった、さまざまな開発法が必要であると述べている[11]。

5.2 実務における導入事例

チケット駆動開発を実際に適用して開発を行った事例は増えつつある[12][13]。チケット駆動開発を実施しているプロジェクトのマネージャーである筆者(丸山)のプロジェクトについて述べる。

このプロジェクトでは、障害管理にTrac[14]を用いてウォーターフォール型の開発を行っていたが、Redmine[15]を利用したチケット駆動開発に切り替えた。次にチケット駆動開発を適用することにより得られた効果、改善された点の他、問題点について述べる。プロジェクトの概要については、表3に示す。

表3 プロジェクトの概要

プロジェクトの規模	
要員	3~6名 PM : 1名, PL : 1名, メンバ
期間	1プロジェクト半年前後
工数	20~50人月程度
利用したツール	
現在	Redmine + Subversion + ReviewBoard
過去	Trac + CVS + (コードレビューはTracのプラグインを使用)

- ・業務で役に立った (改善された) 点
TiDD 導入前と比較して改善された点について

挙げる。まず、TiDDを導入することにより、タスクの「見える化」に効果があった。「見える化」とは、ソフトウェア開発プロジェクトの可視化により、情報共有化、コミュニケーション増加、モチベーション向上、進捗の可視化などに効果をもたらすことを指す[16][17]。

このプロジェクトでは、進捗管理はExcelを使用していたが、Redmineで一元管理することにした。プロジェクト管理ツールであるMicrosoft ProjectやExcelであれば、各クライアントに対し、ツールを導入しなければならなかった。しかし、RedmineはWebベースの為、各クライアントにツールをインストールするコストと時間を軽減した。

また、各タスクの参照が容易であるため、プロジェクト全体で使われやすくなった。先の作業まで把握できるのでプロジェクトのゴールが見えやすくなり、ペースが掴みやすくなった。TiDDの考え方では、すべての作業はチケットを発行するルールが前提であるため、担当者の勝手な判断による作業が発生しにくくなった。

次に、バージョン管理ツールとの連携した。チケットの無いタスクのコミットが削減できた。チケットなしにコミットができないため、こっそりバグを修正できなくなり、的確な管理をできるようになった。

・問題点

チケット駆動開発を利用することにより、導入時の負荷を問題としている。2点あり、まず、リーダーに掛るチケット管理の作業負荷である。次に、ルール作りとチームへの定着化である。ただし、慣れればそれほど負荷ではない。また、ツールの使い勝手についてレビューツールとの連携が良くない(Redmine, ReviewBoard)。Redmineのワークフローとは連携できず、チケットのレビュー通知とレビューツールのレビューワークフローが二重管理となってしまう(チケット番号は連携可能)。ただし、ReviewBoardは使う価値はある。その他、チケットの親子関係が綺麗に表示されないなどの問題点がある。

・問題点への対応策

事前にメンバへ導入の意図を伝える。そのことにより、改善に前向きなチームとなり定着するのが早くなる。また、事前にチケットのワークフローなどの資料を用意する。このプロジェクト場合は、wikiを活用している。そして、フルスペックでの導入を考えず、メンバに対し負荷がかかるので習熟度をみながら、必要の度合いが高いものを導入した。

5.3 2011年度でのプロジェクトへの適用案

5.2の業務での活用事例を参考に、チケット駆動開発を2011年度のプロジェクトに適用する。Redmineをプロジェクト管理に利用する。メンバ全員が使うため、チーム結成時にRedmineの使い方をレクチャーする予定である。

プロジェクトでは、アジャイル開発の全てのプラクティスを適用することが難しいことを3.3で述べた。

2010年度では、メンバの作業が見えていなく、進捗状況を把握することが困難であった。そのため、TiDDを導入することでタスクの「見える化」を図る。そして、ルールを徹底しすべての作業はチケットにより管理する。

ペアプログラミングは、ソフトウェア品質を向上させるためにアジャイル開発では必要であり、ReviewBoardを用いることで補う。

また、経験や技術を必要とするリファクタリングやテストはPMまたは、開発経験者によるサポートが必要になる。そのため、PMが開発を担当する場合を考え、マネジメントの負荷を軽減することを次節にて述べる。

5.4 PMの負担軽減

本プロジェクトでは、PMに負荷がかかり作業時間が予想より増えてしまった。その問題をRedmineの活用により軽減する。

ExcelシートによるWBSでの管理は変化が速いアジャイル開発では向いていない。そのため、Redmineのガントチャート機能により管理する。

また、進捗状況の可視化する目的でEVMを用いていた(表2参照)が、進捗状況の把握が難しかったため、PMに負担をかける原因になっていた。

小川明彦氏、阪井誠氏の著書『Redmineによるタスクマネジメント実践技法』[18]によるとチケット駆動開発では表4のように置き換えることができる。

表4 EVMの置き換え

EVM	TiDD
PV	測定日時点の全チケットの総予定工数
AC	測定日時点の全チケットの総実績工数
EV	測定日時点の終了チケットの総予定工数

Redmineを利用することで表4の値は、データベースに格納されているため、今後のコストやスケジュールを簡単に計算できシミュレーションすることができる。2010年度の活動では、PMたちが進捗報告時にEVMで進捗状況を報告するため

に多くの時間（コスト）を割いていた。Redmineを利用することにより、EVM作成の手間を軽減できると考える。

5.5 評価方法

本研究の評価方法であるが、2011年度の慶應義塾大学における産学連携プロジェクト「協創型ソフトウェア開発」において筆者（木崎）がPMとなり、アジャイル開発型プロジェクトを行う。

そして、問題として挙げたソフトウェア品質、プロジェクトマネジメント、コミュニケーション、モチベーションが改善されたか評価する。

具体的な評価方法であるが、この講義では、ソフトウェア開発プロジェクトの進捗やコストを計測する手法として用いられるEVMを使用する。

EVMのデータは2010年度と比較する。それにより、学生たちのプロジェクトに対するモチベーションを定量的に判断する。

また、プロジェクトマネジメントに割いた時間を2010年度と比較し、チケット駆動開発によるプロジェクト管理がマネジメントの負荷（時間）を減らすか評価する。

ソフトウェア品質については、納品できる品質であることを評価基準とする。厳密なメトリクス測定は検討中である。

講義の最後に、PMとのコミュニケーションの齟齬やメンバー間でのコミュニケーションに問題がなかったかアンケートにより評価する。

6. おわりに

本論文では、ソフトウェア開発PBLにおいて、経験の少ないPMと学生をサポートするマネジメント手法にチケット駆動開発を採用して、2010年度に実施したプロジェクトで発生した問題（ソフトウェア品質、プロジェクトマネジメント、コミュニケーション、モチベーション）を改善させることを計画した。

アジャイル開発をPBLにそのまま適用することは難しい。Redmineを用いたチケット駆動開発は、分散環境のような対面で作業できない環境においても、アジャイル開発を成功させるための手助けになると考える。

謝辞

本プロジェクトは、慶應義塾大学SFCの大学学部生と共同で行いました。また、慶應義塾大学名誉教授の大岩元先生、静岡大学助教の松澤芳昭先生、慶應義塾大学講師の岡田健先生、BPS株式会社の渡辺正毅さんに多大なご協力を頂きここに謝

意を表します。

参考文献

- [1] コラマネ, <http://collam.crew.sfc.keio.ac.jp/>, 2010.
- [2] 松澤 芳昭, 大岩 元: 産学協同の Project-based Learning によるソフトウェア技術者教育の試みと成果, 情報処理学会論文誌 Vol.48 No.8 pp.2767-2780, 2007.
- [3] 松澤 芳昭, 杉浦 学, 大岩 元: 産学協同の PBLにおける顧客と開発者の協創環境の構築と人材育成効果, 情報処理学会論文誌 Vol.49 No.2 pp.944-957, 2008.
- [4] Kent Beck, B. Boehm: Agility through discipline A debate, IEEE Computer, pp.44-46, 2003.
- [5] Rising, Janoff: The Scrum Software Development Process for Small Teams, IEEE Software, pp.26-32, 2000.
- [6] 川端 光義, 阪井 誠, 小林 修: 効果的なXPの導入を目的としたプラクティス間の相互作用の分析, ソフトウェアシンポジウム2004論文集, 2005.
- [7] 小川 明彦, 阪井 誠: チケット駆動開発 -BTSでExtreme Programmingを改善する-, SQiPシンポジウム2009, 2009.
- [8] カタぞう: <http://katazou.jp/>, 2010.
- [9] Android Market: <https://market.android.com/>, 2011.
- [10] アンドロナビ: <http://andronavi.com/>, 2011.
- [11] 阪井 誠: 時間とコンピュータ(ソフトウェア開発と時間), 会誌「情報処理」Vol.52, No.6, pp.618-661, 2011.
- [12] 木崎 悟, 成田 亮, 丸山 英通, 中鉢 欣秀: グローバルなソフトウェア開発におけるマネジメント手法, 研究報告ソフトウェア工学(SE), 情報処理学会, 2011.
- [13] Kazuhiro Matsuo, Shota Anzawa: Work in Progress Project Practices of Agile Software Development for Undergraduate Students, Frontiers in Education Conference (FIE), IEEE, 2010.
- [14] Trac, <http://trac.edgewall.org/>, 2011.
- [15] Redmine, <http://redmine.jp/>, 2011.
- [16] 福田 進一: 「見える化」から見えてきた「見える化」の効果, プロジェクトマネジメント学会研究発表大会予稿集 2008(春季), pp.297-301, 2008.
- [17] 神谷 芳樹: ソフトウェア開発プロジェクトの可視化(「見える化」)に関する実証的な取り組みについて, 平成22年度情報処理学会関西支部大会講演論文集, 情報処理学会, 2010.
- [18] 小川 明彦, 阪井 誠: Redmineによるタスクマネジメント実践技法, 翔泳社, 2010.