Vol.2019-CE-152 No.14
Vol.2019-CLE-29 No.14
2019/11/16

PBLにおけるコミュニケーションツールと開発ツールの可視化情報を利用したプロセス評価方法の提案

日戸 直紘[1]　伊藤 恵[2]　大場 みち子[2]

概要：多くの情報系大学において，Project-Based Learning(PBL) が広く取り入れられ，有効性や実践例が多数報告されている．一般講義と同様に PBL においても，講義設計や学び・質の状況把握から講義改善や学びの支援を行うことが望ましい．プロジェクト途中段階での成果や達成度の把握，受講初期と終盤の学びの状況などのプロセスを評価する必要性があるが，それらの評価実施は難しい．我々はプロセスを評価し改善するモデルである能力成熟度モデル統合 (CMMI) に基づいた PBL のためのプロセス評価手法を開発し，本手法を用いた PBL の評価に関する研究を行っている．先の研究では，開発した評価手法を用いて，実際に本学で実施されている PBL に適用して有効性を示した．しかし，この手法では学生自らが自己評価を行うことから評価結果の客観性や妥当性に課題が残る．本論文では，これらの課題解決を目指し，PBL で利用されるコミュニケーションツールと開発ツールに着目し，ツール群の可視化情報を利用した PBL のプロセス評価方法を提案する．先の研究と同様のプロジェクトを実験対象として，従来手法と本論文で提案する手法を用いた場合の実験結果の比較分析を行った結果と考察について報告する．

キーワード：PBL，学習評価，プロセス評価，可視化

Proposal of Process Evaluation Method Using Visualization of Communication Tool and Development Tool in PBL

Naohiro Hinoto[1]　Kei Ito[2]　Michiko Oba[2]

Abstract: Project-Based Learning (PBL) has been widely adopted in many information universities, and its effectiveness and practical examples have been reported. In PBL as well as other general lectures, it is desirable to carry out the improvement from the grasp of situation, quality of lecture design and learning. It is necessary to evaluate the process, such as grasping results and achievements during PBL and the status of learning at the beginning and end of the course. However, it is difficult to evaluate them. We have developed a process evaluation method for PBL based on Capability Maturity Model Integration (CMMI), which is a model to evaluate and improve the process, and are studying the evaluation of PBL using this method. In our previous study, the developed evaluation method was applied to PBL conducted in our university and its effectiveness was shown. However, the previous proposed method leaves problems in the objectivity and validity of the evaluation result, because each student oneself carries out the self-evaluation. In this paper, we focus on the communication tools and development tools used in PBL to solve these problems, and propose a process evaluation method for PBL using log visualization results. We apply the proposed method to the experimental PBL project in the previous study and report the evaluation results.

Keywords: PBL, Learning Assessment, Process Evaluation, Visualization

[1] 公立はこだて未来大学 システム情報科学研究科
Graduate School of Systems Information Science,
Future University Hakodate
[2] 公立はこだて未来大学
Future University Hakodate

© 2019 Information Processing Society of Japan

1. はじめに

文部科学省は，IT 人材の不足が今後一層深刻化する可能性が高いなどから，高度 IT 人材の育成を日本の極めて重要な課題としている [1]．高等教育機関において高度 IT 人材の育成が求められる背景のなかで，Project-Based Learning(PBL) が注目されている．PBL とは課題解決型学習を指し，学習者が課題解決に向け主体的 (あるいは能動的) に取り組む実践的な授業法である．座学等で学んだ体系的知識の定着と利活用への効果も期待されており，より良い教授法を目指し試行錯誤が繰り返されている．

実践的な教育を推進する情報技術人材育成のための実践教育ネットワーク形成事業 (通称：enPiT)[2] では，PBL に関する活動が活発的に行われている．また，enPiT では教員や enPiT に関係する企業人が集い，教育内容・方法等をはじめ PBL の教授法や体制など様々な議題に関して議論を行う FD(Faculty Development) 活動が盛んに行われている [3]．enPiT で得られた知見の共有や議論，課題解決などを行うことにより，各大学の PBL および enPiT 全体の改善に寄与している．

PBL の学習評価，特にプロセス (活動そのもの) の評価を適切に実施することは困難であるという報告がされている [4][5]．PBL の学習評価は PBL 運営の一つの課題となっており，FD 活動でも PBL の評価に関する話題が頻繁に取りあげられている．PBL のフィールドや条件，講義目的が異なる場合が多いことや，PBL の担当教員が専任ではなく通常の教員業務との兼任の場合，学生の講義時間外活動が頻繁に発生するため，学習者を観測しづらいことなどが考えられる．

PBL などの演習科目は，一般の講義と同様に質を高めるための改善活動を行うことが望ましく，PBL 実施途中での成果や達成度の把握，受講初期と終盤の学びの状況などのプロセス評価の実施が期待される．前述した PBL の学習評価の課題なども踏まえると，実施時の学習者や教育者の負担が大きいなどの課題にくわえ，プロセス評価の実施は難しい場合が多い．

これらの背景を踏まえ，我々は，Capability Maturity Model Integration (CMMI)[6][7] に着目し，PBL のプロセス評価手法を開発・提案してきた [8][9][10]．先の研究では，開発した評価手法を用いて，学生が自己評価によってプロセスを評価する方法を提案し，実際に本学で実施されている PBL に適用して有効性を示した．しかし，この手法では学生自らが自己評価を行うことから客観性，妥当性に課題が残る．本論文では，これらの課題解決を目指し，PBL で利用されるコミュニケーションツールと開発ツールに着目し，可視化情報を利用した PBL のプロセス評価方法を提案する．我々が過去プロセス評価実験を行ったプロジェクトと同様のプロジェクトを実験対象として，従来手法と本論文で提案する手法を用いた場合の実験結果の比較分析を行った結果と考察について報告する．

本稿では以降，2 章で我々が提案してきた従来のプロセス評価手法について述べ，3 章では従来の評価手法の課題とその解決アプローチについて述べる．4 章では今回提案する手法を用いた実験結果について述べ，5 章で実験結果について考察する．6 章では本研究の結論と今後の課題について述べ，まとめる．

2. 従来の提案手法

2 章では, 従来の研究で我々が着目した CMMI(Capability Maturity Model Integration)[6][7] に基づいたプロセス評価手法とルーブリックの開発過程について概説する．その後，従来手法を用いた実験結果について，実験対象のプロジェクトと合わせて紹介する．

2.1 Capability Maturity Model Integration

CMMI は，米国国防総省とカーネギメロン大学 (CMU) により設立されたソフトウェア工学研究所 (SEI) によって誕生した．プロセス評価・改善活動 (組織やプロジェクトの有効性を評価，様々な情報を総合的な分析，プロセス改善に有効な策の展開) を行うための評価モデルであり，世界中の多くの企業で採用されている．

2.2 CMMI に基づいたプロセス評価手法とプロセス領域

従来の研究で我々が開発したプロセス評価手法は，プロセスを評価改善するモデルである CMMI を応用することができれば PBL のプロセスを評価できると考え開発に至った．

CMMI-V1.3 では，評価観点別に 22 のプロセス領域が定義されており，プロジェクトの管理活動から実装，テスト，トレーニング (教育) など幅広い活動をカバーしている．そのため，学生が主体的に学ぶ PBL では，適当でない領域が複数あると考えられる．本提案手法では CMMI の 5 領域と独自に開発した 1 領域の計 6 領域を選定・開発し，評価を行う．選定した 5 領域は，プロジェクトに必要なデータ収集・分析，定期的なスケジュール等の計画・確認，プロセス実施の評価などの領域を中心に選定している．

表 1 は，これら 6 領域の CMMI における区分，プロセス領域名，選定理由についてまとめたものである．以降，各領域を表 1 に記載された領域 I ～ 領域 VI で表記する．

独自に開発したプロセス領域 VI の “Scrum 遵守” 領域は，Scrum Guide[11] に記載されたロール・イベント等の遵守に関して評価を行うための領域である．Scrum Guide にくわえ，CMMI Institute が発行する CMMI によるアジャイルパフォーマンス改善のガイド [12] も参考にプロセス領域の開発を行った．

表 1　対象領域と独自領域の名称と選定理由

領域	区分	プロセス領域名	選定理由(概要)
I	プロジェクト管理	プロジェクト計画策定	学生が課題とするスケジュールなどを含む**プロジェクト計画**を扱うため
II		プロジェクトの監視と制御	**プロジェクト状況**を適切な頻度で**把握・是正処置するため**の領域であるため
III	支援	測定と分析	あらゆるデータの**収集・分析・測定**を扱う領域であるため
IV		プロセスと成果物の品質保証	**プロセスや作業成果物などを客観的に評価**あるいは見通しを立てる領域であるため
V	エンジニアリング	検証	**成果物に対するレビュー準備・遂行・是正**スクラムイベントとの親和性が高いため
VI	-	Scrum遵守(独自領域)	**Scrum Guide**に記載されたイベントやロール等の遵守に関して検査するため

2.3　ルーブリックの開発

本来の CMMI は専門のアプレイザーが評価を行うのに対し，本手法では学生が主体的に学ぶという PBL の観点から，学生自身が評価を行うことを想定している．CMMI を PBL のプロセス評価手法に応用し，学生自身が評価を行うにあたり，学生が CMMI に付随する知識を持っていないことが考えられる．

これらの課題から，CMMI の知識がない学生でも評価できるよう，CMMI の各プロセス領域に存在する項目に準拠しつつ，専門的な単語や表現をできる限り排除したルーブリックを開発した．

2.3.1　開発過程における検討

表 1 で示した “プロジェクト計画策定” “プロジェクトの監視と制御” “測定と分析” “プロセスと成果物の品質保証” “検証” “Scrum 遵守” の 6 領域について Lv.0 ～ Lv.3 の 4 段階の基準を設定したルーブリックを開発した．CMMI は，成熟度 Lv と呼ばれる組織を 5 段階 (Lv.1～Lv.5) で評価する Lv を持っており，成熟度の最初の段階である Lv.2(Lv.1 はプロセスが混沌としている組織を指し該当領域はない) には 7 領域が存在する．初期は，この 7 領域を選定し開発したルーブリックを用いて，PBL 経験者の学生に自分たちのプロセスを評価してもらった．その評価結果やヒアリング，進行形の PBL に適用した結果などを元に PBL に適切な領域の選定やルーブリックの基準などを見直し・設定することで，学生自らがプロセスを評価可能なルーブリックの開発を試みた．

完成したルーブリックの一部を表 2 に示す．

2.4　従来手法を用いた実験

2.4.1　実験対象のプロジェクト

本学では，通年必修 PBL であるシステム情報科学実習 [13] のほかに，高度 ICT 演習と呼ばれる学部生・大学院生を交えた学年・コースの壁がない学年混成型の自由参加型で単位の出ない PBL を開講している [14]．学部 1 年生から修士 2 年生まで様々な学年・コースの学生がプロジェクトを組んで活動している．高度 ICT 演習は通常，毎週月曜日の 4～5 限 (14:50～18:00) を活動日としており，基本的に全プロジェクトこの時間内のみ各教室に集まり開発等のプロジェクト活動を行っている．

実験は，高度 ICT 演習のプロジェクトの 1 つである FinTech プロジェクトを対象とした．

FinTech プロジェクトは，高度 ICT 演習にて 2017 度発足された，学内向け仮想通貨を企画・開発するプロジェクトである．メンバーは，学部 1 年生 2 名，学部 2 年生 5 名，学部 3 年生 6 名，学部 4 年生 2 名，修士 1 年生 4 名の計 19 名で構成されている．プロジェクトは，R&D チームと応用開発チームの 2 チームで構成されている．R&D チームはブロックチェーンを含めた仮想通貨そのものを開発，応用開発チームは仮想通貨を利用する際の取引所の様な役割を果たす仮想通貨アプリを開発している．

本実験では，Scrum を取り入れている応用開発チームを対象とした．応用開発チームは全 6 チームに細分化されており，学部 1 年生 2 名，学部 2 年生 5 名，学部 3 年生 5 名の計 12 名構成の多学年 PBL となっている．

実験は，プロジェクト活動期間のうち開発期間である 2018 年 10 月～12 月の 3 ヶ月間，計 6 スプリント分実施した．

2.4.2　従来手法を用いた実験結果

実験対象である 6 チームを Team A ～ F として，従来手法を用いた実験結果を図 1～7 に示す．図の縦軸は評価結果 (0%～100%)，横軸はスプリント (1st～6th) を表している．

図 1 は，全プロセス領域の評価結果を相加平均として割り出したグラフであり，図 2～7 は，プロセス領域ごとの評価結果をまとめたものである．

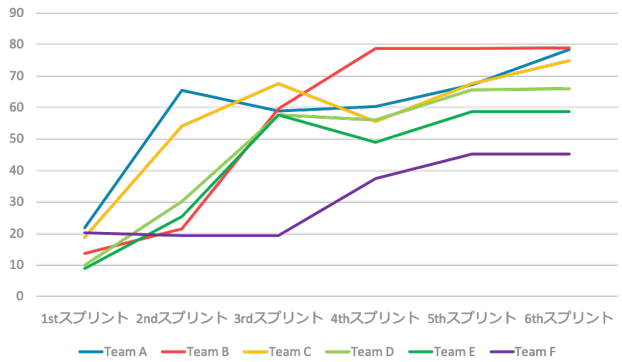

図 1　全プロセス領域の評価結果の相加平均値の推移

3.　従来の提案手法の課題と解決アプローチ

3.1　従来の提案手法の課題

従来の提案手法を用いた実験は，各スプリントのふりかえり活動時に実施しており，以下の手順で行った．

(1) ルーブリックが記載された評価シート

表 2　ルーブリック (領域 III: 測定と分析)

No.	項目名	項目概要	Level. 0	Level. 1	Level. 2	Level. 3
SP1.1	プロセスを客観的に評価する	自分たちの活動や状況に対するプロジェクトの遵守状況を評価する	実施していない	自分たちの**活動や状況**(プロセス)を客観的に評価する(してもらう)プロセスがある	Lv.1 + 評価中に見つかった不遵守項目を特定している	Lv.2 + プロセスを改善できる教訓/課題/問題を見つけている
SP1.2	作業成果物を客観的に評価する	成果物を客観的に評価する	実施していない	自分たちの**成果物**を客観的に評価する(してもらう)プロセスがある	Lv.1 + 評価中に見つかった不遵守項目(バグ含め)を特定する	Lv.2 + それらを(文書等に)記録している
SP2.1	非遵守課題を伝達し解決する	評価で特定された問題を適切な管理層やメンバーに共有し課題を解決する仕組みをつくる	実施していない	特定された問題をメンバー内で解決または共有し，問題として取り上げている	Lv.1 + 解決できな場合は上位の管理層や他チームと協力し，解決できるかを確認，解決していくよう努力している	Lv.2 + 特定された問題を追跡し，未解決の問題に対して定期的にレビューや解決を行う、解決まで追跡を続ける
SP2.1	記録を確立する	品質保証活動の記録を管理する活動	実施していない	評価した結果や，特定された問題の状況及び結果を報告できるよう記録している	Lv.1 + 文書化するなど十分な詳細さで報告できるよう記録している	Lv.2 + 必要に応じて評価や問題解決の状況及び履歴を変更している

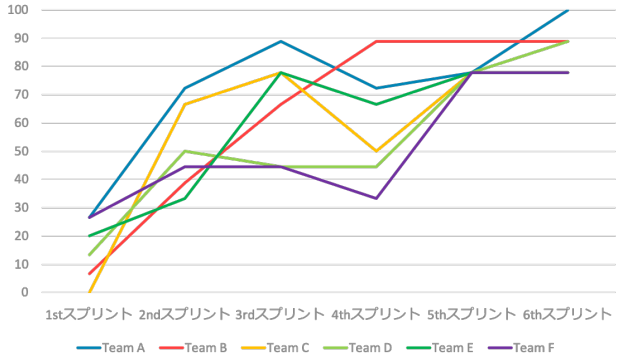

図 2　“Scrum 遵守”領域の評価結果

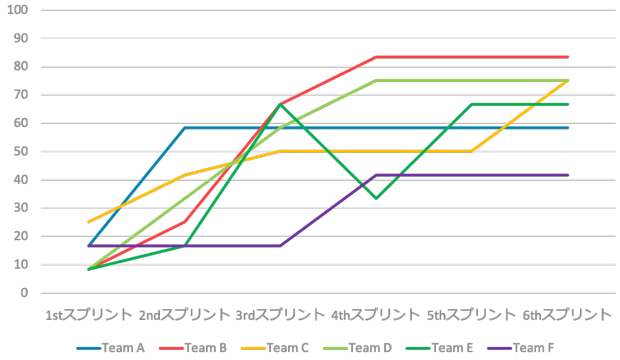

図 4　“プロセスと成果物の品質保証”領域の評価結果

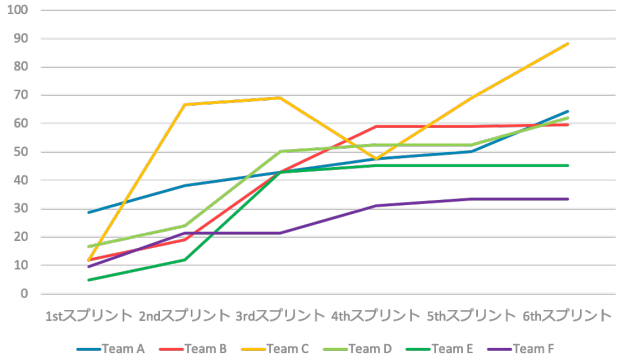

図 3　“プロジェクト計画策定”領域の評価結果

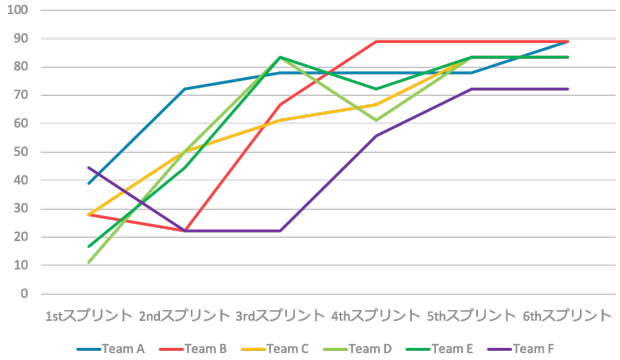

図 5　“検証”領域の評価結果

(Google Spreadsheet) を利用

(2) 各評価項目に対し，自分たちがどのレベルに当てはまるのかルーブリックを基に議論/評価

(3) 当てはまるレベルを Lv.0〜Lv.3 から選択

(4) 手順（2）〜（3）を全プロセス領域 (評価項目) 分実施

従来の提案手法では，ルーブリックを用いることで客観性と妥当性の実現を目指した．しかし，評価根拠となるドキュメントなどのエビデンスがない (要求していない) ことや，意図的に評価結果を高くする，評価を適当にやっているなどの事象を否定できないことなどから客観性や妥当性に課題が残った．

3.2　課題解決アプローチ

3.1 節で述べた，従来の提案手法に関する課題に対して，2 つの解決アプローチを提案する．

1). コミュニケーションツールの可視化によるプロセスの評価

2). 開発ツールの可視化によるプロセスの評価

PBL で利用されているコミュニーケーションツールと開発ツールのログを可視化することによって，プロセスをより客観的かつ詳細に分析できると考えた．以降，可視化によるプロセス評価の可能性に至った理由や，可視化する方法に関して詳しく述べる．

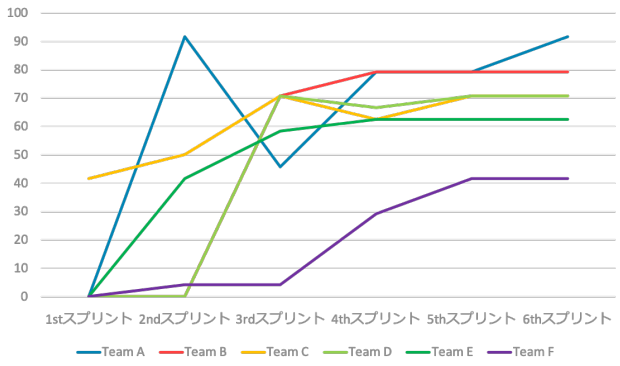

図 6 “測定と分析”領域の評価結果

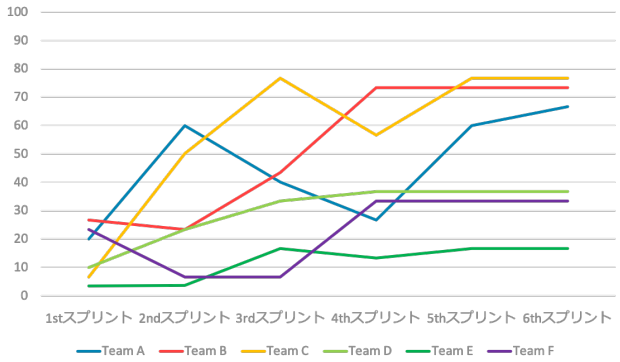

図 7 “プロジェクトの監視と制御”領域の評価結果

3.3 コミュニケーションプロセスの可視化アプローチ

3.3.1 コミュニケーションツールから取得可能な情報

ほとんどの PBL では，コミュニケーションツールが導入されており，学生間だけでなく学生と教員，TA 間のやりとりにも利用されている．講義内だけでなく，講義時間外のやり取りや報告，指示などにも利用されており，利用頻度はメールよりも非常に高い．

メールのような定型的なやりとりがない場合が多く，心理的ハードルが低い．そのため，効率化や円滑なコミュニケーションの支援となり，活発な利用に結びついている．また，コミュニケーションツールは，チャット機能だけでなく，スタンプなどのリアクション機能や音声通話機能が搭載され，多種多様な情報がやり取りされている．

コミュニケーションツールについて，それらを利用するチームやプロジェクトの PBL 中の利用状況を長期的に分析することにより，プロセスを可視化・評価できると考えた．プロジェクト初期と終盤の投稿メッセージ数を比較することで，増加傾向にあれば議論やコミュニケーションが活発化したプロセスであることなどが推測できる．

3.3.2 Slack の情報可視化システム

実験対象のプロジェクトでは，コミュニケーションツールとして，Slack[15] が最も活発に利用されていたため，Slack のログ情報を可視化の対象とした．Slack では，1 つのワークスペースと呼ばれるグループに参加し，その中にチャンネルと呼ばれる小さいグループを目的別に複数つくることで円滑なコミュニケーションを行えるビジネスチャットツールである．

Slack は Web API を公開しており，リアルタイムなデータを取得することが可能である．ユーザ別だけでなくチャンネルごとの投稿数や共有ファイル数，投稿日時などを取得することができる．その取得した情報を動的に可視化するため，BI(Business Intelligence) ツールを利用する．BI ツールとは，企業や集団などに蓄積された大量のデータを分析し可視化し，迅速な意思決定などを手助けするツールである．本研究では接続可能なデータベースの種類の多様性などから Microsoft 社の Power BI[16] を使用した．可視化システムの構成は図 8 の通りである．

図 8 Slack の情報可視化システムの構成

上記のシステムを用いることで，多種多様な情報からコミュニケーションに関するプロセスを分析・可視化できると考えた．

3.4 開発プロセスの可視化アプローチ

3.4.1 開発ツールから取得可能な情報

ソフトウェアを開発する場合，開発を支援するシステムやサービスを利用することが一般的である．その中でもバージョン管理システムである Git は，非常に多くのソフトウェア開発 PBL で利用されている．

Git には，開発者が行ったコーディングやコミットなどの操作，コメントなどの活動履歴を保存されており，Git を用いたサービスの API から情報を取得できる．それらを取得し分析することにより，PBL の開発に関する特徴やプロセスを可視化できると考えた．

3.4.2 Git の情報可視化ツール

実験対象のプロジェクトでは，開発ツールとして，GitHub[17] を利用していたため，Git のログ情報を可視化の対象とした．

Git の可視化には，Git リポジトリのコミット履歴の可視化のためのツールとして利用可能である，GitStats[18] を利用する．GitStats から読み取れない情報に関しては，GitHub API を利用して取得することで保管していく．GitStats は，Git リポジトリに含まれるコミット数などの情報を可視化できるため，チームの活動過程を把握できる可能性があるとして，PBL の活動 (プロセス) 分析の試みに利用されている [19]．ただし，GitStats は GitHub ユー

ザ名が日本語である場合文字化けすることや，可視化結果が視覚的に分析しづらいことから，取得した情報を我々で再編集したものを掲載する．

本提案では，開発過程で行われるコミット履歴に着目する．コミットとは，Git で利用されるソースコードをリポジトリへ登録するイベントのことである．コミットで登録される情報は，ソースコードの内容以外にコミットを行ったユーザ名，コミット日時，コミット内容を示す文章が含まれている．コミットが行われた曜日・時間，コミット数などを集計することで，開発プロセスの傾向が分析できると考えた．

4. 可視化実験

4.1 実験対象プロジェクト

3 章で提案した評価手法と従来の評価手法で行った実験結果を比較するため，可視化実験でも従来手法の実験と同様のプロジェクトを対象とする．

FinTech プロジェクトは，コミュニケーションツールに Slack，コード管理システム/サービスに Git/GitHub を利用している．FinTech プロジェクトの Slack ワークスペースは，実験対象チーム以外のチームも利用しているほか，学内イベント等のやりとりも行っているため，実験対象チームのチャンネルを対象に分析する．

また，実験の対象とした応用開発チームでは，6 チームで 1 つのアプリケーションを開発しており，6 チームで 1 リポジトリを共有している．そこで，各ユーザの情報を個別に取得し，その情報を帰属するチームの情報として統合することでチーム別の可視化を行っていく．

4.2 Slack の可視化システムを利用した実験結果

図 9 はスプリントごとの投稿メッセージ数，図 10 はスプリントごとの累計投稿メッセージ数を示したグラフである．

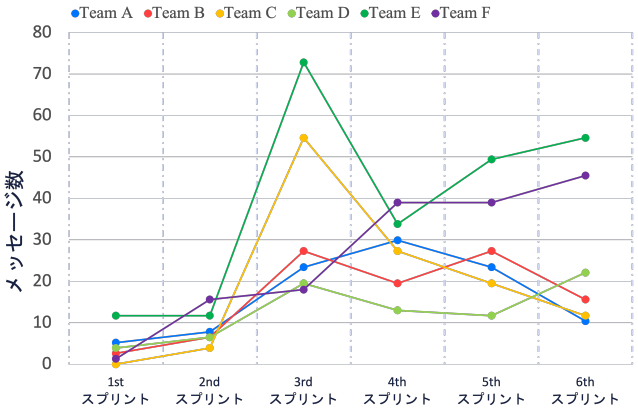

図 9 スプリントごとの投稿メッセージ数

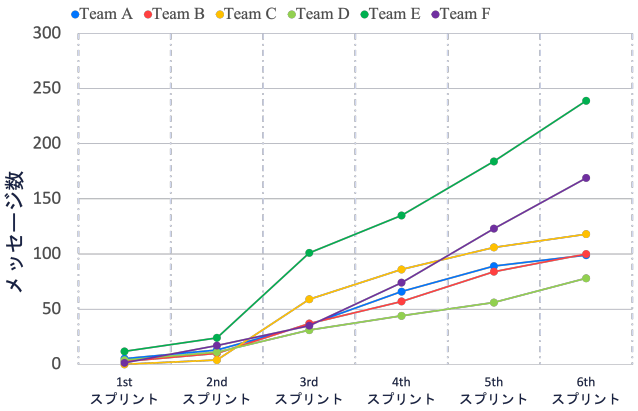

図 10 スプリントごとの累計投稿メッセージ数

4.3 Git のログ情報の可視化ツールを利用した実験結果

図 11 は曜日/時間別コミット数の推移を表しており，マス目の数字は，対応する曜日と時間帯に登録されたコミット数である．色が濃くなるにつれコミット数が多くなり，空欄のマス目はコミットがなかったことを示している．

Week day	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Mon							1					3	1		2	11	9	6	1	4	2	3	2	1
Tue		3		1								1	1	3		1	5	1				1	1	1
Wed		1	1						4		1		1		1	1	3	3			2			1
Thu		1										1	3		1	2	6	4	2	2	2	4	4	1
Fri	1	2	1	1					2		2	3	1	3	1	4	1	1	1					1
Sat	2													1		3	1	3	3		2			
Sun			2	1												1	1	2		1	1	2	2	

図 11 曜日/時間別コミット数

図 12 は，スプリントごとのコミット数，図 13 はスプリントごとの累計コミット数の変化を示している．

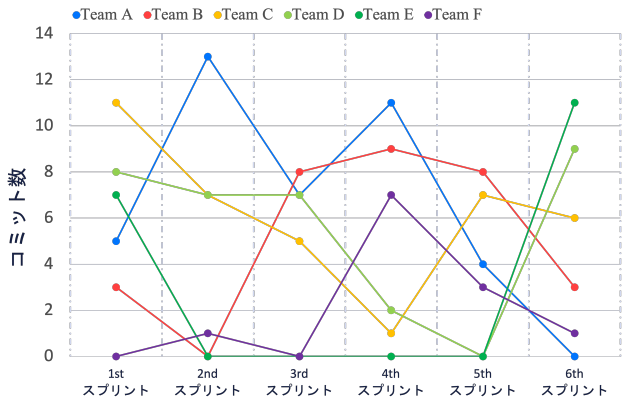

図 12 スプリントごとのコミット数の推移

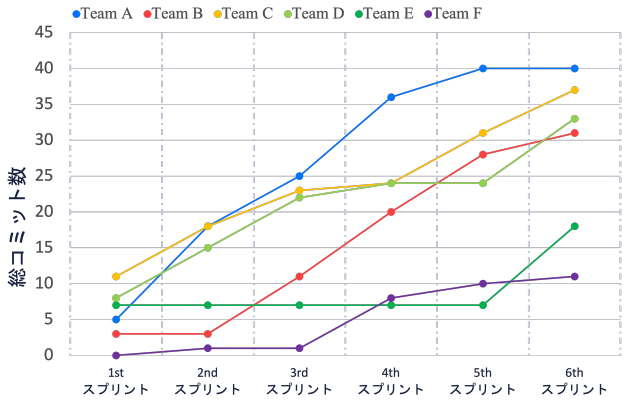

図 13　スプリントごとの累計コミット数の推移

5.　分析と考察

実験結果を分析・考察していくにあたり，評価結果を裏付ける情報として，投稿メッセージ・コミットメッセージや，メンバーへのヒアリングから情報を収集した．

以降，4 章に記載した実験結果，従来手法で行った評価結果と本実験結果であるコミュニケーションツール・開発ツールの可視化情報について，収集した情報と合わせ分析・考察していく．

5.1　Slack-ログの可視化実験に関する評価と考察

図 9 から投稿メッセージ数の変動と活動内容を分析すると，開発が活発化したタイミング，遅延や仕様変更が行われたタイミングで投稿メッセージ数が増加している傾向にある．

各チーム 3rd スプリントでメッセージ数が増加しており，特に Team C と Team E が最も明らかに増加している．これは，2nd スプリントから 3rd スプリントにかけて Team C と Team E のメンバーが仕様変更や UI/UX の設計について活発に議論し，それらを他チームに展開していたことなどから投稿メッセージ数が増加していると推測される．Team C はこのスプリント期間内にアプリケーションのフロント部分の土台を作成していた．そのため，他チームへの共有・連絡事項を頻繁に行っていたことがわかり，このタイミングにおける開発の中心が Team C，Team E であったことがわかる．

3rd〜4th スプリントにかけて，全体的に投稿メッセージ数が減少傾向であることがわかる．4th スプリントの期間は，学内イベント (中間試験や研究発表等) が行われており，全体的に開発が進まなかったためと考えられる．一方で，Team A と Team F のみ投稿メッセージ数が増加しており，図 12 から，これらのチームはコミット数が伸びていることから開発が進んでいたことがわかる．

図 1 をみると 4 チームが 4th スプリントで一度評価結果が下がっていることや，図 2 では，5 チームが 4th スプリントで評価結果が下がっていることがわかる．Slack の投稿メッセージ数の推移と従来手法でのプロセス評価結果が一致しており，コミュニケーションツールの可視化から従来手法のプロセス評価結果の分析を行うことで客観性や妥当性を検証できる可能性が高い．

Team E は，アプリケーション開発のリーダーが所属していたことから，全チームのまとめ役を行うことが多かった．投稿メッセージ数を見ると，全スプリントを通して，Team E が最も投稿メッセージ数が多いことがわかる．これらから全スプリントを通して，Team E が開発を牽引したことがわかる．後半スプリントでは，Team F の投稿メッセージ数が伸びていることがわかる．Team F は，上級生の割合が高かったことなどから，技術フォローや開発イベントのファシリテートなどを行っており，後半は Team E とともに，Team F もプロジェクトの牽引の手助けを行えていた可能性が高い．

減少傾向でみると，Team A/B/C は 4th スプリント以降，他チームと比較して投稿メッセージ数が減少している．これは，4th スプリントで所有していた開発タスクを完了し，残りの期間はリファクタリングや追加の簡単な機能の実装，他チームの支援などを行っていたことがわかった．

コミュニケーションツールの情報から，メッセージ数の増加・減少のタイミングや，チームごとの傾向が顕著に現れているタイミングを可視化することができた．該当するスプリント期間中について，メンバーへのヒアリングや投稿メッセージの内容，振り返りの際の議事録などから情報を得ることでそれらの理由を分析可能であり，どのようなプロセスかを把握・評価できる見通しが得られた．

5.2　Git-ログの可視化実験に関する評価と考察

図 11 の結果から，全体的にコミット時間が分布しているものの，月曜日の 15〜17 時台が最も多い時間帯になっていることから，プロジェクト時間内を中心に開発が進められていることが推測できる．また，木曜日もコミットが多い傾向にあることから，プロジェクト時間外にも木曜日を中心に開発を行っていたことがわかった．

スプリントごとのコミット数の推移をあらわす図 12 からは，開発が計画的かつ順調に行われているチームとそうでないチームを評価することができる．

増減はあるものの後半までに開発を進めている (コミット数が多い) チームは，後半に進むにつれ 0 に近づいていることがわかる．計画性の無いチームや，途中でタスクが増加したチームなどは後半に向けて開発が佳境を迎え，コミット数が多くなっていると推測できる．

図 13 から Team E/F は他チームと比較して，コミット数が少ない．これは，全体の技術フォローやファシリテート，UI/UX の設計を担当していたことから総じて少なくなっていることが理由である．メインで開発 (コーディン

グ) を行っていたチーム A〜D は，スプリントを追うごとにコミット数を増加させていることがわかる．

5.3 従来手法との比較と考察

スプリントごとの累計投稿メッセージ数 (図 10) と従来手法の評価結果について，Spearman の順位相関係数は 0.583(p<.001) と正の相関が見られた．スプリントごとの累計コミット数 (図 13) と従来手法の評価結果について，同じく Spearman の順位相関係数を計算したところ，0.691(p<.001) とやや強い正の相関が見られた．コミュニケーションツールと開発ツールの可視化情報と従来手法を用いたプロセス評価結果には，相関があることがわかった．従来手法では，プロセス評価結果に対して，客観性・妥当性に課題が残っていたが，これらの活動プロセスに関わる客観的なデータをコミュニケーションツールと開発ツールから可視化することにより，客観性や妥当性を補完できる可能性が高い．

コミュニケーションツールと開発ツールの可視化により特徴的なタイミング (スプリント) を可視化することができた．そのタイミングのメッセージ内容やヒアリング内容から考察を行うことにより，PBL の各チームがどのようなコミュニケーション・開発プロセスであったかを評価することもできた．しかし，コミュニケーションツールと開発ツールの可視化情報のみでは，どのような事柄に関するプロセスの変化やイベントが起きたのかを読み取るのは難しい．可視化情報とヒアリングや投稿メッセージ内容にくわえて，従来手法のプロセス評価結果とを合わせることで，より詳細なプロセスを分析することができた．このことから，従来手法と組み合わせて分析・評価することで，客観性や妥当性を実現しながら，PBL のプロセス評価を行える可能性が高いことを示唆できた．

6. おわりに

本研究では，PBL で利用されるコミュニケーションツールと開発ツールに着目し，可視化情報を利用した PBL におけるプロセス評価方法を提案した．また，我々が過去プロセス評価実験を行った結果とツールの可視化情報を比較するため，プロセス評価実験を行った PBL を対象に本提案手法を用いた実験を行った．結果として，コミュニケーションツールを可視化することにより，メッセージ数の推移などからプロセスを評価することや，開発ツールの可視化より，開発プロセスを分析することができた．過去行ったプロセス評価データをコミュニケーションツールと開発ツールの可視化情報と合わせて分析することで，PBL のプロセスを評価できる見通しを得た．

今後は，PBL で利用されるツールを継続的に可視化することで，学生がプロセスを改善する際，教育者が評価する際に活用可能な動的なシステムの実現を目指していく．

参考文献

[1] 文部科学省:「成長分野を支える情報技術人材の育成拠点の形成（enPiT）」公募要領, pp. 2, 2017.
[2] 文部科学省: enPiT 成長分野を支える情報技術人材の育成拠点の形成, http://www.enpit.jp/, <cited:2019.07.28 >.
[3] CHISHIRO Hiroyuki , CHUBACHI Yoshihide:Faculty Development Camp for Project Based Learning in enPiT, コンピュータソフトウェア，36 巻, 1 号, pp.14–23, 2019.
[4] 本庄加代子: PBL の課題克服に向けたプロジェクトマネジメント理論の有効性, 東洋学園大学紀要, 25 号, 145-164, 2017.
[5] 櫻井浩子, 山本雅基, 海上智昭, 小林隆志, 宮地充子, 奥野拓, 粂野文洋, 春名修介, 井上克郎: 情報系大学院生に対する実践教育の効果測定, 工学教育, 65 巻, 1 号, pp.52–57, 2017.
[6] CMU/SEI: CMMI for Development-Version 1.3, Vol.1.3, 2010.
[7] CMMI Institute : Introducing CMMI V2.0,2018.
[8] 日戸直紘, 伊藤恵, 大場みち子: 能力成熟度モデル統合に基づいた PBL における定量的学習評価手法の提案, 日本ソフトウェア科学会第 34 回大会講演論文集, 日本ソフトウェア科学会, 2017.
[9] 日戸直紘, 伊藤恵, 大場みち子: アジャイルソフトウェア開発 PBL のための CMMI に基づいた定量的学習評価手法とその効果, 第 5 回実践的 IT 教育シンポジウム (rePiT2019) 論文集, 日本ソフトウェア科学会, 2019.
[10] 日戸直紘, 伊藤恵, 大場みち子: アジャイルソフトウェア開発 PBL における形成的アセスメントの効果測定, 日本ソフトウェア科学会第 36 回大会講演論文集, 日本ソフトウェア科学会, 2019.
[11] Ken Schwaber, Jeff Sutherland: Scrum Guide, 2017, https://www.scrumguides.org/docs/scrumguide/v2017/, <cited:2019.07.28 >.
[12] CMMI Institute : A Guide to Scrum and CMMI: Improving Agile Performance with CMMI, 2016.
[13] 美馬のゆり: 未来を創る「プロジェクト学習」のデザイン, 公立はこだて未来大学出版, 2018.
[14] 公立はこだて未来大学 HP:高度 ICT 演習について, https://www.fun.ac.jp/department/course/ict/ict_about/, <cited:2019.10.10 >.
[15] Slack: Slack, https://slack.com/, <cited:2019.08.02 >.
[16] Microsoft: Power BI, https://powerbi.microsoft.com/, <cited:2019.08.02 >.
[17] GitHub: GitHub, https://github.co.jp/, <cited:2019.10.10 >.
[18] SOURCE FORGE: GitStats, http://gitstats.sourceforge.net/,<cited:2019.10.10 >.
[19] 松原克弥, 伊藤恵, 木塚あゆみ:コード更新履歴の可視化による分散 PBL でのチーム活動評価の試み, 第 4 回実践的 IT 教育シンポジウム論文集, 日本ソフトウェア科学会,pp.49–55, 2018.

© 2019 Information Processing Society of Japan