

過去の PBL の開発履歴を活用した PBL 運用支援

伊藤 恵 木塚 あゆみ 奥野 拓 大場 みち子

近年、情報教育の分野において PBL(Project Based Learning) を用いた実践的な教育が着目されてきている。PBL の有効性については一定の評価があるものの、PBL をどのように運用するのが効果的・効率的であるか、そのノウハウの蓄積は不十分である。一方、システム開発を中心とした PBL 実施の際にはプロジェクト管理やバージョン管理のための様々なツールが用いられることが多く、それらのツールには多くの情報が蓄積される。本研究では、過去の PBL によって蓄積されたプロジェクト管理ツール Redmine のチケット情報と、バージョン管理ツール Subversion のコミット情報を分析し、そこから PBL 運用に有用な情報の抽出・提示を行うことを目指す。

In recent years, practical education using PBL(Project Based Learning) is focused on information education area. Effectiveness of PBLs achieve a measure of legitimacy. But, it is insufficient to accumulate know-how on efficient and effective operation of PBLs. For system development PBLs, several tools are used such as project management tool, version control tools and so on. In such tools, a lot of information about the development are accumulated. This study aims to provide useful information for PBL operation by collecting and analyzing both tickets of Redmine which is a project management tool and commits of Subversion which is a version control tool.

1 背景と目的

近年、情報教育の分野において PBL を用いた実践的な教育が着目されてきている [1][4]。PBL の有効性については一定の評価があるものの、PBL をどのように運用するのが効果的・効率的であるか、そのノウハウの蓄積は不十分である。産学連携により企業から実務経験のあるプロジェクトマネージャが大学の PBL に参加することによって、その経験や勘を踏まえてプロジェクト運用をする事例も多いが、永続的に企業からの協力が得られるわけではないことのほか、開発プロジェクトとしての成否と教育としての PBL の成否は一致しないことから、大学教員によるプロジェクト運用が本来であることは間違いない。

一方、ソフトウェア開発プロジェクトの残す様々な情報を可視化して、それによってプロジェクト管理に

役立てる取り組みは様々ある [3][6]。しかし、こういった可視化情報でさえも、システム開発やプロジェクトマネジメントの実務経験のない大学教員にとっては、そのプロジェクトの進捗状況がどの程度のもので、どのくらい深刻か否か判断するのは困難であることが多い。一定数以上のグループが同時並行に同じシステムを開発するグループ並行型 PBL において、複数グループのプロジェクト情報を同時並行に可視化し、その比較を行うことで、実践経験の少ない教員でもプロジェクトへの指導が可能な程度に情報を得られる試みは行われている [2][5][7]。また、これらの試みではよく作りこまれた PBL 教材を使うなど、教員が PBL 全体を制御し易い状態で PBL を実施している。しかし、PBL の実践的教育としての効果を十分生かすためには多くのチームが同じ題材を扱うことが良いとは限らないし、必ずしも制御し易くないような実践的な題材や環境で行うことで、より効果的な実践的教育が行えると考えられる。また、そもそも十分なグループ数を構成するほど学生数が多くはない実施状況も多くあり得る。

A Support for PBL Operation Using Development History of Past PBLs

Kei Ito, Ayumi Kizuka, Taku Okuno, Michiko Oba,
公立はこだて未来大学, Future University Hakodate.

そこで本研究では、同時並行に同じシステムを開発するグループを比較するのではなく、過去に実施された類似の PBL の情報と、現在進行中の PBL の情報を比較可能な支援をすることで、実務経験の少ない大学教員が、グループ数や開発題材に依らず、有効なプロジェクト運用を行えるようになることを目指す。

2 節では関連する先行研究や先行事例を紹介する。3 節では PBL 運用支援のための具体的なアプローチを述べ、このアプローチに基づく運用支援を実施するためにデータ収集した過去の PBL について 4 節で述べ、進行中の PBL との対比可視化方法について 5 節で、運用支援の試みについて 6 節で述べる。

2 関連研究

2.1 開発履歴の利用

ソフトウェア開発プロジェクトの残す様々な情報を可視化して、プロジェクトのモニタリングを行い、それによってプロジェクト管理に役立てる取り組みは行われてきた。EPM[3][6]ではソフトウェア開発プロジェクトにおいて構成管理、メーリングリスト管理、障害管理などの各システムからリアルタイムに情報を収集して、プロジェクトの状態を可視化し、それによってプロジェクトのリアルタイム管理を目指した。現在では Trac や Redmine などの有用なプロジェクト管理ツールにより、EPM と類似のプロジェクト状態可視化が可能となっている。しかし、システム開発やプロジェクトマネジメントの実務経験のない大学教員にとっては、これらから得られる情報だけでは、その進捗状況がどの程度のものか、どのくらい深刻か判断するのは困難であることが多い。

2.2 PBL 支援システム

井垣, 柿元ら[2][5]は、システム開発やプロジェクトマネジメントの実務経験のない教員でもシステム開発 PBL における指導を容易にすることを目指して、PBL 環境を構築している。しかし、彼らの PBL 実践では、複数の学生グループが同じ仕様書から同じシステムを開発するグループ並行型 PBL の特徴を活かしたものであり、一定数以上の学生グループが同時並行に同じシステムを開発するのをモニタリングする

ことにより、進行の遅い(あるいは早い)グループや特異な活動をしているグループを見つけるのを容易にしていることや、教材としてよく作りこまれた開発題材を複数年に渡って使用することで、教員側も開発題材についてよく把握しているという状況での実践である。十分な数のグループ数が構成できないような学生数の場合には対応が難しいほか、システム開発がより実践的であるために開発題材をより新しいものに変えていく場合やグループごとに異なる題材を扱う場合への対応も困難である。

福安ら[7]は同じようなグループ並行型 PBL を同様なグループ数に対し、より少ない教員数で実践しており、教員による成果物検査等を一部省略しながらも、教員数以外の条件は同様である。

3 アプローチ

本研究では、過去に実施した類似 PBL の開発履歴を活用し、それらと現在進行中の PBL のプロジェクトデータと対比させて可視化することで、そのプロジェクトの進捗の程度などを実務経験のない大学教員でも容易に確認できるようにすることを目指す。

過去の PBL の開発履歴として、昨今の PBL にてよく利用されていると考えられるチケット管理システムおよびバージョン管理システムからのデータ収集を行い、蓄積する。そのデータと現在進行中の PBL のチケット管理システムおよびバージョン管理システムのデータとを対応付けて、運用支援となる情報の可視化提示を行う(図 1)。

3.1 チケット管理システムからのデータ収集

チケット管理(または Issue 管理)システムからのデータ収集は、trac のチケット情報、Redmin のチケット情報、および、GitHub の Issue 情報からのデータ収集を想定しているが、4 節で述べるように今回実際に開発履歴データの収集をした PBL はいずれも Redmine を使用していたため、今回は Redmine のチケット情報収集のみについて述べる。

Redmine はその設定によってチケットに保存できる情報も異なってくるが、今回収集対象とするのは以下の情報である。

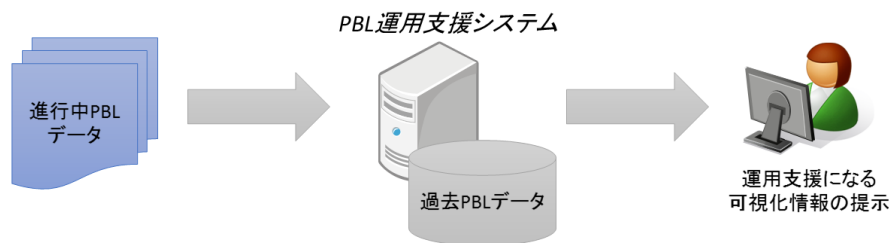


図 1 PBL 運用支援システム

- チケットの作成者
- チケットの担当者
- チケットの作成日時
- チケットの作業が着手された日時
- チケットが close された日時
- チケットの種別

作成者および担当者については、個人名を特定する必要はなく、チーム内の誰であったかを収集することで、作成者および担当者のチーム内での偏りを抽出するために使用する。チケットの種別はチケット管理システムやその設定によって様々だが、収集できる範囲でその偏りや使い分けの程度を判別するために使用する。

PBL ではチケット管理システムを使うよう指示しても、必ずしも十分に使いこなせるわけではない。チケットを作成し忘れたまま作業をしてしまい、作業が終わった時点で気が付いて作成する場合や、実際は担当者が決まってもチケット上の担当者を設定しないまま進んでしまう場合などもある。もちろん、チケットが適切に作成され、適切に更新されるのが望ましいが、PBL のチケット情報にはそういうデータの欠落もあり得るという前提で収集する。

また、多くのチケット管理システムではそのチケットの作業の工数見積もりや作業実績なども入力可能であり、これらの項目が十分に活用されていれば、PBL の運用支援のための極めて有効な情報となるが、実際に PBL からのデータ収集を行ったところ、入力し忘れが非常に多い項目でもあったため、今回は敢えて収集対象から除外した。

3.2 バージョン管理システムからのデータ収集

データ収集対象のバージョン管理システムとして CVS, Subversion, Git を想定しているが、4 節で述べるように今回実際に開発履歴データの収集をした PBL はいずれも Subversion を使用していたため、今回は Subversion のリポジトリ情報収集のみについて述べる。

Subversion のリポジトリに保存されている情報のうち、今回収集対象とするのは以下の情報である。

- コミットした人
- コミットした日時
- コミットによって追加/変更/削除されたファイルとその行数
- コミットメッセージの有無
- コミットメッセージ等からチケット管理システムのチケットとの関連が分かれば関連するチケットの番号

コミットによる具体的なファイルの変更内容までは考慮せず、行数の変化までを対象とする。リポジトリには開発対象のソースコード以外にドキュメント類も格納する可能性があるが、ドキュメント類は PBL で使用する開発プロセスによって大きく異なるため、今回は主にソースコードに着目する。

また、バージョン管理システムではリポジトリで管理されているファイルを更新するためのコミット等の操作だけでなく、開発作業のためにリポジトリからファイルを取り出すアップデート等の操作の記録も、そのプロジェクトの活動の様子を把握するために有用である可能性があるが、今回はそこまで考慮していない。

4 過去のPBL開発履歴

著者ら所属大学で2013年度に実施された4種類11チームのPBLについて蓄積された情報を収集した。情報収集した4種類のPBLは表1の通りである。プロジェクト学習は学部3年生を対象とした必修科目のPBLであり、2013年度に学生15名が配属されて行われたプロジェクトの中の計5つのサブプロジェクトに関してデータ収集を行った。また、enPiTは文部科学省の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を受けて全国15大学が中心となって開始した「分野・地域を超えた実践的情報教育協働ネットワーク」[1]により、著者ら所属大学で他大学の学生を含む大学院修士1,2年の学生を対象として実施したものであり、2013年度の夏季の集中講義形式で行われたミニPBL4プロジェクトと後期に行われた分散PBL2プロジェクトのデータ収集を行った。プロジェクト学習は対象者が学部3年生であるため、参加メンバーのほとんどはPBL経験が少ないか初経験であるのに対し、enPiTの各PBLは参加メンバーの半分以上は2回目以降のPBL参加であり、一部の学生は大学内でのPBLに多数参加しており、PBL経験が豊富であった。

これら11チームのプロジェクトは、いずれもWebアプリケーション開発を行っており、開発に際してWebアプリケーション開発フレームワークCakePHPを使用していた。また、プロジェクト管理ツールとしてRedmineを、バージョン管理ツールとしてSubversionを用いた。従って、どのPBLについてもRedmineおよびSubversionに蓄積された情報を収集した。

なお、どのチームも共通してCakePHPフレームワークを用いたWebアプリケーション開発を行っており、CakePHPフレームワークに元々含まれるソースファイルが1400超あるため、どのチームもソースファイルとしてバージョン管理していたファイル数が1400~1800程度となっている。

4.1 チケット数の推移

チケット管理システムから得られるデータをプロジェクトの状況を示すために可視化する方法の一つに、チケット数の推移をグラフ化する方法がある。

図2はenPiT分散PBL/Bチームにおいて、そのプロジェクト期間中にチケット数、着手済みチケット数、作業終了/完了済みチケット数がどのように推移したかを示すグラフである。プロジェクト期間を通じた平均チケット着手率は78.7%で最終着手率は93.7%、平均チケット完了率は49.7%で最終完了率は89.8%であった。チケットの記録上は未着手あるいは未完了の作業が残っていることになるが、多くのチケットを作成したチームの中では着手率/完了率ともかなり高い方である。

図3はenPiTミニPBL/Aチームにおいて、そのプロジェクト期間中にチケット数、着手済みチケット数、作業終了/完了済みチケット数がどのように推移したかを示すグラフである。enPiTミニPBLは5日間の短期集中で行うPBLであり、週1~2回集まって数か月掛けて行うPBLとは日程的に大きく異なるが、チケット数の推移は通常のPBLと大きな違いは見受けられない。プロジェクト期間を通じた平均チケット着手率は85.7%で最終着手率は92.4%、平均チケット完了率は58.7%で最終完了率は74.7%であった。

これら2チームを含めいくつかのチームはPBL期間が実質的に終わった後に、未着手だったチケットを着手に、未完了だったチケットを完了に変更していた。

図4はプロジェクト学習本開発/Yagaigekiチームの種類別のチケット数推移を示すグラフである。使用していたRedmineサイトの設定により、チケットの種類(トラッカー)は「タスク」「サポート」「機能」「バグ」の4種類が利用可能であり、行うべき作業の内容や目的によってチケットを分類して登録することが期待されていたが、このチームはプロジェクト期間の初期および末期に作成したいくつかのチケット以外はすべて「サポート」になっており、チケットの種類を適切に設定しなかったようである。これらの種類間のチケット数のバランスは開発対象や主に開発すべき内容、あるいは、プロジェクトのその時点の状況によって異なると考えられるが、このような極端な偏り

表 1 情報収集した 4 種類の PBL

PBL 種類	期間	学年	チーム数/学生数
プロジェクト学習 模擬開発	約 2 か月	学部 3 年	3 チーム 5 名
プロジェクト学習 本開発	約半年	学部 3 年	2 チーム 5~10 名
enPiT ミニ PBL	5 日間	修士 1,2 年	4 チーム 4~5 名
enPiT 分散 PBL	約 4 か月	修士 1,2 年	2 チーム 6~8 名

表 2 11 チームから収集したデータの概要

PBL 種類/チーム名	期間	学年/学生数	チケット数	コミット数	ファイル数
プロジェクト学習模擬開発/A チーム	約 2 か月	学部 3 年/5 名	25	208	1,645
プロジェクト学習模擬開発/B チーム	約 2 か月	学部 3 年/5 名	17	108	1,596
プロジェクト学習模擬開発/C チーム	約 2 か月	学部 3 年/5 名	25	106	1,642
プロジェクト学習本開発/Event チーム	約半年	学部 3 年/10 名	240	526	1,578
プロジェクト学習本開発/Yagaigeki チーム	約半年	学部 3 年/5 名	152	121	1,813
enPiT ミニ PBL/A チーム	5 日間	修士 1,2 年/5 名	79	23	1,473
enPiT ミニ PBL/B チーム	5 日間	修士 1,2 年/5 名	47	45	1,479
enPiT ミニ PBL/C チーム	5 日間	修士 1,2 年/5 名	38	327	1,487
enPiT ミニ PBL/D チーム	5 日間	修士 1,2 年/4 名	59	109	1,498
enPiT 分散 PBL/A チーム	約 4 か月	修士 1,2 年/6 名	13	47	1,552
enPiT 分散 PBL/B チーム	約 5 か月	修士 1,2 年/8 名	127	196	1,827

はチケットの使用方法が不適切であるか、もしくはプロジェクトが特殊な事態になっている可能性が高い。

4.2 チケット作成者/担当者の偏り

図 5 は enPiT ミニ PBL/D チームの Redmine チケットの作成者別および担当者別のチケット数をグラフ化したものである。凡例の 126 や 130 などの数値はこのチームにメンバとして所属していた学生個々を区別するための ID であり、凡例の最後にある NULL はどの学生にも割り当てられていないことを示す。5 日間の短期集中 PBL 中にこのチームが作成したチケット数は 59 であり、作成者別で見ると上位 2 名の学生がそれぞれ 30 個、21 個のチケットを作成している。PBL においてチケット作成者が偏るのはよくありがちであり、極端な場合、ほぼすべてのチケットを 1 人の学生が作成する場合もあるが、それ自体が PBL の進行上問題であるとは限らない。担当者別に

見ると、担当者未割当のチケットが 59 チケット中 16 も残っているのは多いが、それ以外のチケットは 5 名のメンバ学生にほぼよくばらついている。このチームの 5 名のメンバ学生の担当チケット数の標準偏差は 2.5 であり、担当チケットに関するばらつきはかなり少ないと言える。

4.3 コミット回数推移

それぞれのプロジェクトの PBL 期間中の述べコミット回数の推移を調べたところ、enPiT 分散 PBL/B チームの推移は図 6 の通りであった。要求工程やヒアリングを主に行っていた 10,11 月はコミットは少なく、開発が本格化した 12 月に急増、1/24 に行われた発表会に向けて 1 月に段階的に増加している。2 月末にも学外での発表会に出展したため、2 月後半にも更にコミットしている。

一般的な PBL では週に 1~2 回メンバが集まって

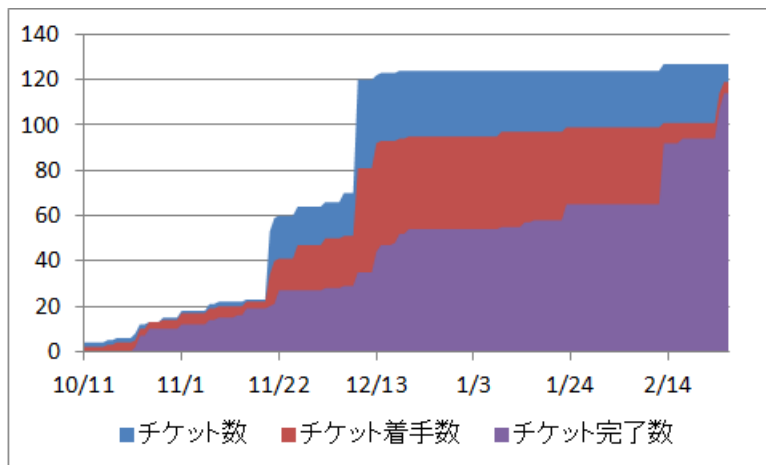


図 2 分散 PBL B チームのチケット数推移

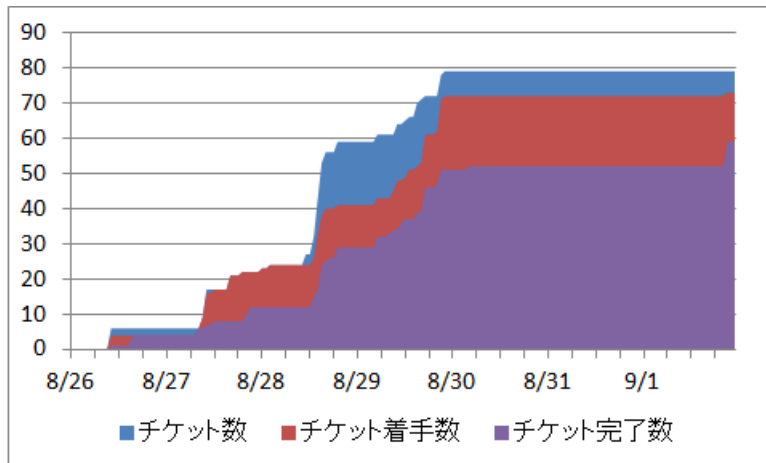


図 3 ミニ PBL A チームのチケット数推移

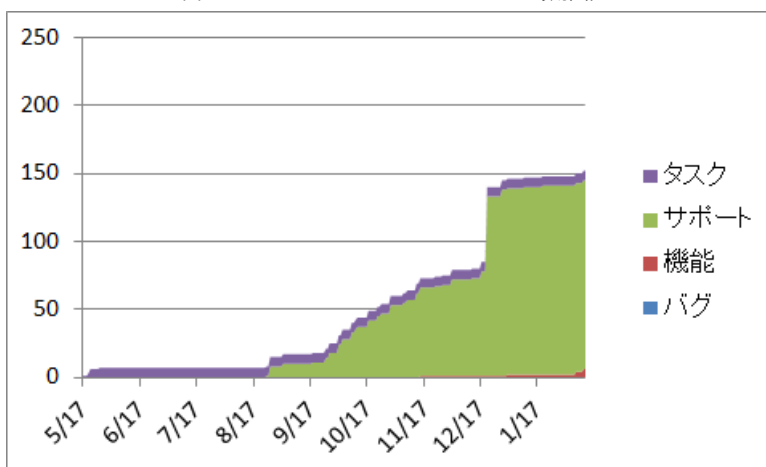


図 4 プロジェクト学習本開発 Yagaigeki チームの種類別チケット数推移

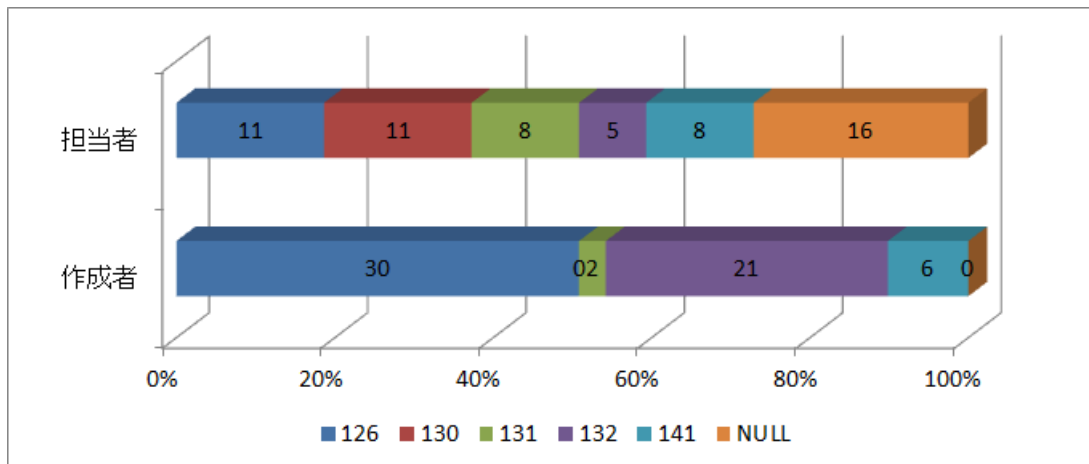


図 5 enPiT ミニ PBL D チームの作成者/担当者別チケット数

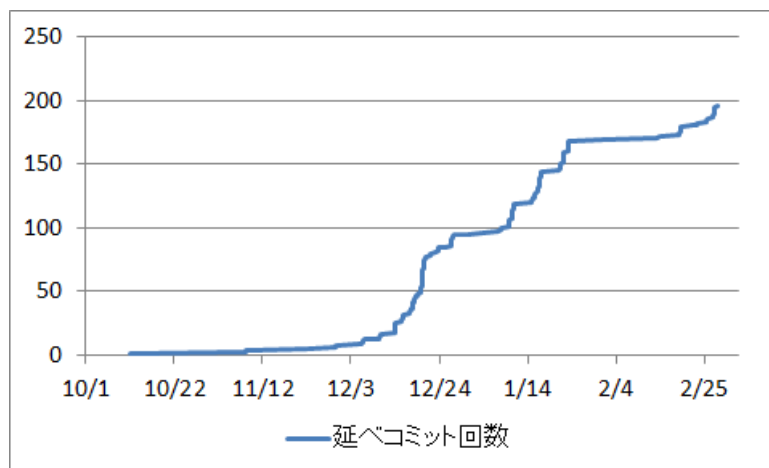


図 6 enPiT 分散 PBL B チームのコミット回数推移

ミーティングや作業を行うという形態が多いが、短期集中形式で行われ、期間中ほとんどメンバが顔を合わせていた PBL ではコミット回数の推移傾向は変わるだろうか。5 日間の短期集中で行われた enPiT ミニ PBL/C チームの述ベコミット回数推移は図 7 の通りである。5 か月近く掛けて行われた enPiT 分散 PBL のグラフ (図 6) と比べて、特段傾向が異なっているわけではない。つまり、PBL 期間が長くて短くても述ベコミット回数推移の大まかな傾向はあまり変わらないと考えられる。もちろん、総コミット数が極端に少ないなどのバージョン管理の利用が特殊なプロジェクトの場合はこの限りではない。

4.4 ファイル数/ソースコード行数の推移

各プロジェクトでバージョン管理していたソースファイル数およびソースコード行数の推移を調べたところ、enPiT ミニ PBL/A,B,C,D チームのチーム別のコード行数推移は図 8 の通りであった。4 チームとも CakePHP フレームワークを用いているため、元々フレームワークに含まれるソースコード行数が相応にあること、また PBL 開始前に PBL 担当教員が各チーム用の Subversion リポジトリを用意し、そこに CakePHP を丸ごと入れておいたことなどからこのようなグラフになっている。なお、1 チームだけ不自然に行数が少ないのは担当教員の準備ミスと思われる。

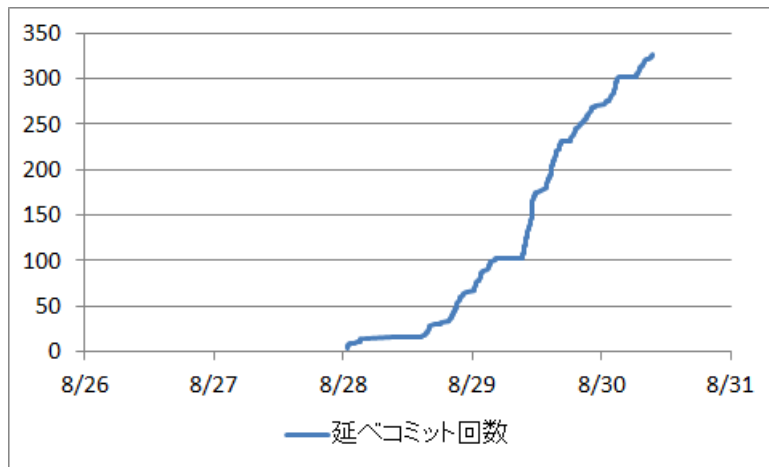


図7 enPiT ミニ PBL C チームのコミット回数推移

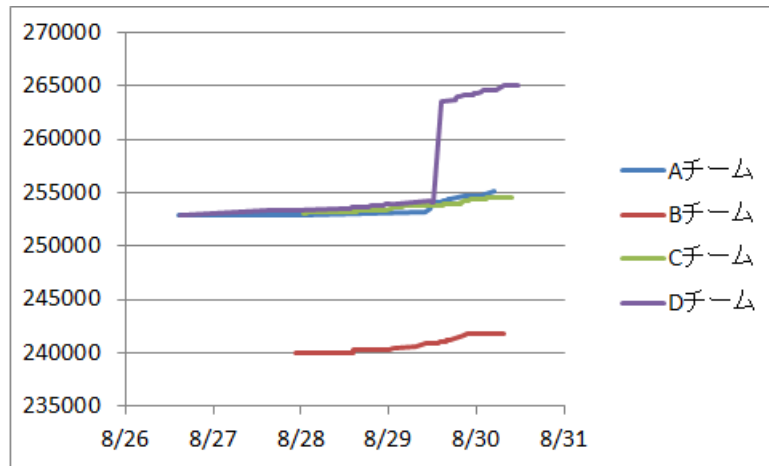


図8 enPiT ミニ PBL 4 チームのコード行数推移

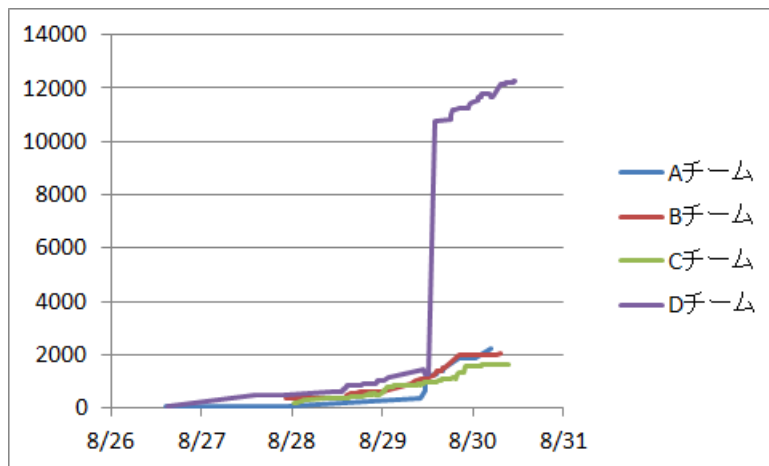


図9 enPiT ミニ PBL 4 チームのコード行数推移 (開始時からの相対)

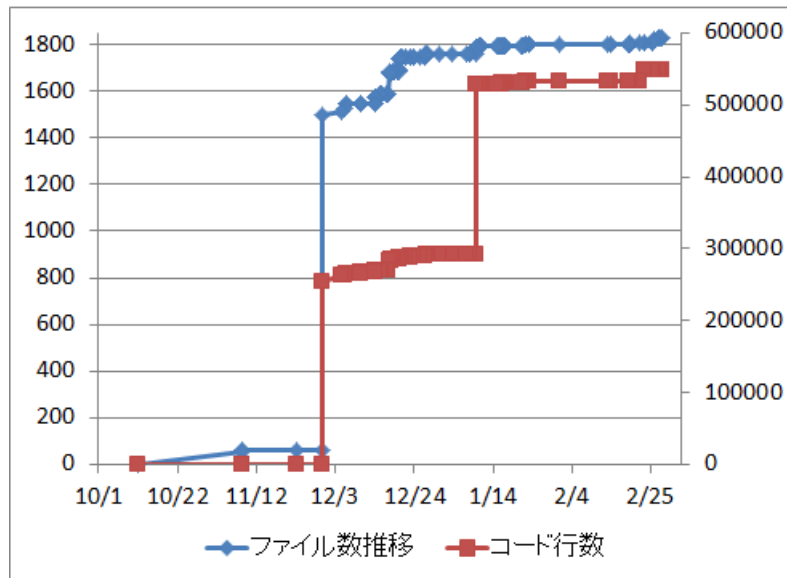


図 10 enPiT 分散 PBL B チームの成果物ファイル数とコード行数推移

フレームワークを使うか使わないか、使うとしてどのフレームワークを使うかなどは、個々の PBL ごとに異なり、ファイル数や行数の絶対値比較はあまり意味をなさないと考えられる。図 8 のグラフを PBL 開始時点からの相対的な行数でグラフ化したものが図 9 である。開始時点からの相対行数変化でみると、4 チーム中 3 チームはほぼ同様の傾向であり、1 チームだけが途中で急激に増えていることが分かる。何が起こったのか実際にデータを確認したところ、行数が急激に増えたのはフレームワークに対して追加のアドオンを導入したからであった。PBL の運用としてアドオンの導入に特に注意する必要があるというわけではないが、ファイル数や行数の急激な変化が見られれば、そのチームで何が起きているか気を払う必要はあると考えられる。

図 10 は enPiT 分散 PBL/B チームがバージョン管理していたファイル数と行数の推移グラフである。グラフを見ると 11 月下旬にファイル数と行数が急激に増え、また 1 月にファイル数は若干、行数が急激に増えている。11 月下旬は開発を本格化させるために CakePHP フレームワークそのものをバージョン管理下においたタイミングであり、1 月はソースコード以外のドキュメント類をバージョン管理下に含めたタイ

ミングであった。

4.5 コミット者の偏り

バージョン管理システムへのコミット者が同じチーム内でどの程度偏っているかを調査した。図 11 は enPiT 分散 PBL/B チームのコミット者別のコミット回数グラフである。このチームはメンバが 8 名とやや多く、役割分担や作業負荷が偏る可能性が高いと推測された。この結果を見ると、主要 4 メンバの間ではかなりバランスが良く、その 4 メンバと残りの 4 メンバの間ではかなり偏っていたということが分かる。教員からのコントロールが少なく、学生側のスキル向上意識が不十分な場合、システム開発 PBL において実装を担当する学生が極端に偏ることがしばしばある。実装担当者が偏るとその学生だけがコミット回数が極端に多く、それ以外の学生がコミット回数が極めて少なくなるのが通例である。実装を担当する学生が偏ることを教師側が容認する場合には、即座に注意すべき事態とはならないが、その場合でも全体的な作業負荷が偏り過ぎて、実装担当学生の PBL 以外の活動にまで影響を及ぼす場合もあるため、相応の注意は必要となる。

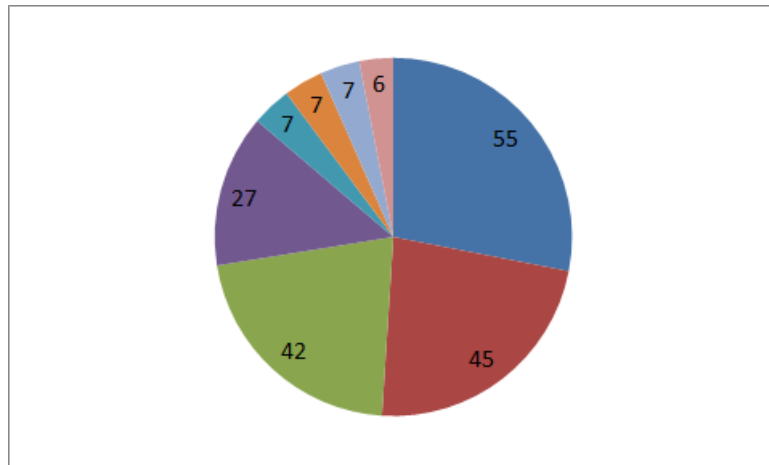


図 11 enPiT 分散 PBL B チームのコミット者別コミット回数

5 過去と進行中の PBL のデータ可視化方法

本節では収集した過去の PBL データと現在進行中の PBL データを、現在進行中の PBL 運用支援のためにどのように可視化するかを述べる。

本研究では本稿冒頭で述べたように題材や環境が異なる PBL に対しても運用支援することを目指している。4 節で述べてきたように今回データ収集した過去の PBL は共通点もあるが実施形態など異なる面も多い。実施形態が異なっても同じ傾向のデータが得られることも分かってきた。

プロジェクト期間は PBL によってかなり異なるが、プロジェクト期間に依らずチケット数の推移やコミット回数の推移は大きく変わらなかった。チケット数が最終的に何個、コミット回数が最終的に何回程度になるのが妥当かは判断が難しいが、プロジェクト期間の時間的な進行割合に対してチケット数やコミット回数の平均的な増加傾向は分かるため、それを現在進行中の PBL の開始してから終了予定時期までの割合と照らし合わせることで、それぞれの増加が少なめであるか多めであるかを提示することが可能である。

PBL を受講する学生は年々入れ替わっていくため、参加するメンバのスキルも変わるが、収集した過去の PBL のデータからはチーム単位でのスキルの差は明らかではなかった。チーム内でのスキル差や意識の差はチケットやコミット回数の偏りとして表れており、

それらは PBL の題材や実施形態にはほとんど依存しないため、過去の PBL と現在進行中の PBL との偏りの程度を比較できるように可視化することで、どの程度の偏りまで許容できるかを検討可能になると考えられる。

データ収集した過去の PBL の中でチームが崩壊状態になり、PBL 期間の後半はほぼ 1 人で活動していたというチームがあった。チームが崩壊したかどうかはデータに直接現れないが、PBL 期間の長さの割にチケット数もコミット回数も少ないというデータ傾向を示していた。同様のことが起こる場合、チケットやコミットが少いか、もしくは、一部のメンバ以外は担当チケットやコミットが極めて少ないなどの傾向が表れると想定される。

今回データ収集した過去の PBL はいずれも同じ Web アプリケーション開発フレームワークを使っており、各チームの開発規模は大きく変わらなかったと考えられるため、このデータを開発規模の異なる PBL と比較して十分な運用支援になるかどうかは未実証である。しかし、開発規模が大きく変わらなかったはずの各チームであるにも関わらずチケット数やコミット回数は大きく異なっていることのほか、PBL として教員が運用する性質上、開発規模が極端に変わることは多くないと考えられるため、運用支援上は PBL の開発規模を厳に捉えなくても良いと考えられる。

6 PBL 運用支援の試み

本稿執筆時点ではまだ開始していないが、収集した 2013 年度実施の 11 チーム分の PBL データを基にして、2014 年度に実施するいくつかの PBL において、運用支援を想定したデータ可視化を行い、実際に PBL 運用の助けとなるか試みる予定である。実施対象となる PBL は、チケット管理やバージョン管理に関してはデータ収集した 2013 年度の PBL と同様に Redmine と Subversion を使用予定であるが、2013 年度の 11 チームすべてで使用していた開発フレームワーク CakePHP を使用しないチームがほとんどになる予定である。使用する開発フレームワークや開発内容が大きく異なった場合に、支援できる度合いがどう変わるかが今回の試みの注目点となる。

7 考察

今回、Redmine や Subversion から収集可能であったが収集対象としなかったデータがある。例えば Redmine では各チケットに見積り工数や作業実績が入力できるため、このデータに着目することで PBL 各チームの見積りの精度やその向上度合いを見ることができるとは考えられるが、現時点では考慮していない。また、Redmine のチケット作成者とチケット担当者について、作成者についてはチーム内で偏ることを容認し、担当者の偏りには注意するという方針で考えているが、関連する別の情報としてチケットの変更者が誰だったかという情報もある。実際にデータ収集した PBL の中のあるチームでは、チケット担当者はチーム内でそれなりにばらついているが、チケットを作成するのも変更するのもほぼ一人が行っているという状態であった。つまり、チケット担当者はチケットで示される作業を実際にやっているがチケット管理システムには全く触らず、システム上のチケット操作はすべて特定の学生に任せているという状態だった。プロジェクトとしてはそれでも進行するが、チケット駆動開発としてはかなり特殊な状態であるため、チケット駆動であることを重視する場合には考慮した方がよい。

今回はチケット管理システムおよびバージョン管

理システムに関して結果的に Redmine と Subversion という組み合わせだけに対応することになった。チケット管理システムやバージョン管理システムには流り廃りがあるため、それぞれ複数のシステムに対応することを目指したい。現時点では GitHub の Issue 機能や Git への対応が有望であるほか、過去の PBL データをより多く収集・活用するために trac や CVS への対応も検討したい。

また、チケット管理やバージョン管理以外の PBL に関するデータも運用支援につながるものがあり得る。議事録や週報などは有用な情報が含まれている可能性があるが、その形式がプロジェクトごとに全く異なるなど、過去の PBL のものと現在進行中の PBL のものを対比して可視化するのは難しいと考えられる。PBL に関するメールのやり取りや Skype での会話ログなども考えられる。従来であればチケット管理やバージョン管理以外のやり取りはプロジェクト用のメーリングリストなどを用いることが多かったが、昨今の PBL において教員側が特に何かを強要しない場合は学生間で自然と Skype 等でのやり取りが行われている。参加学生承諾の下で Skype の会話を記録し、過去の PBL のメーリングリストでのやり取りと対比して可視化することがプロジェクト運用支援になるかどうかとも検討していきたい。

8 まとめと今後の課題

過去の PBL の開発履歴を活用し、それらと現在進行中の PBL のプロジェクトデータと対比させて可視化することで、そのプロジェクトの進捗の程度などを実務経験のない大学教員でも容易に確認できるようにすることを目指し、2013 年度に著者ら所属大学で実施した PBL のチケット管理情報とバージョン管理情報を 11 チーム分収集し、分析した。そして、それらを運用の支援となるように可視化する方法について検討した。本年度の PBL において実際に過去のデータと進行中の PBL のデータを対比可視化して、PBL 運用支援を試みる予定である。また、データ収集範囲の再検討、他の PBL 関連システムへの対応、より多くの PBL 開発履歴の収集なども並行して行っていく予定である。

参考文献

- [1] enPiT 事務局: 分野・地域を越えた実践的情報教育協働ネットワーク (通称 enPiT), <http://enpit.jp>.
- [2] 井垣宏, 柿元健, 佐伯幸郎, 福安直樹, 川口真司, 早瀬康裕, 崎山直洋, 井上克郎: 実践的ソフトウェア開発演習支援のためのグループ間比較にもとづくプロセスモニタリング環境 (教育実践研究論文, < 特集 > 学習・教育支援のための技術開発), 日本教育工学会論文誌, Vol. 34, No. 3(2010), pp. 289–298.
- [3] 大平雅雄, 横森励士, 阪井誠, 岩村聡, 小野英治, 新海平, 横川智教: ソフトウェア開発プロジェクトのリアルタイム管理を目的とした支援システム (ソフトウェア開発環境・開発支援システム, < 特集 > システム開発論文), 電子情報通信学会論文誌. *D-I*, 情報・システム, *I*-情報処理, Vol. 88, No. 2(2005), pp. 228–239.
- [4] 福田晃, 鷗林尚靖, 荒木啓二郎, 峯恒憲, 日下部茂, 金子邦彦, 亀井靖高, 廣重法道: 情報工学系大学教員のための *PBL* 実践ガイド, 九州大学大学院システム情報科学府 情報知能工学専攻社会情報システム工学コース, 2012.
- [5] 柿元健, 井垣宏, 川口真司, 福安直樹, 佐伯幸郎, 早瀬康裕, 崎山直洋: 開発プロセスの観察とグループ間比較による実践的ソフトウェア開発演習の支援環境, 電子情報通信学会技術研究報告. *SS*, ソフトウェアサイエンス, Vol. 109, No. 456(2010), pp. 13–18.
- [6] 横森励士, 市井誠, 新海平, 井上克郎: 開発履歴データのリアルタイム収集・分析システム EPM の拡張について: SRGM を用いた予測グラフの実現および既存解析システムとの連携, 電子情報通信学会技術研究報告. *SS*, ソフトウェアサイエンス, Vol. 104, No. 722(2005), pp. 1–6.
- [7] 福安直樹, 佐伯幸郎, 水谷泰治: リポジトリのリアルタイムな可視化にもとづく *PBL* の支援環境: 継続的な実施を目的として, 電子情報通信学会技術研究報告. *SS*, ソフトウェアサイエンス, Vol. 110, No. 458(2011), pp. 121–126.