

ソフトウェア開発PBLにおける プロジェクトの可視化に基づく適応的な指導の実現

福安 直樹^{†1} 佐伯 幸郎^{†2} 水谷 泰治^{†3} 井垣 宏^{†4} 眞鍋 雄貴^{†4}

^{†1} 和歌山大学 ^{†2} 高知工科大学 ^{†3} 大阪工業大学 ^{†4} 大阪大学

An Adaptive Teaching on Software Development PBL with Visualization of Projects

Naoki Fukuyasu^{†1} Sachio Saiki^{†2} Yasuharu Mizutani^{†3}

Hiroshi Igaki^{†4} Yuki Manabe^{†4}

^{†1}Wakayama University ^{†2}Kochi University of Technology

^{†3}Osaka Institute of Technology ^{†4}Osaka University

PBL (Project Based Learning) 形式による実践的なソフトウェア開発演習 (ソフトウェア開発 PBL) は受講生の主体的な活動による学習方法であり, 指導者は全体の状況を把握し受講生の活動に対して適切な助言を行うことが重要である. 本稿では, 我々がこれまでに提案しているソフトウェア開発 PBL における各チームの状況把握を支援するツールを利用した, 受講生に合わせた適応的な指導の実現について報告する.

キーワード: PBL, 適応的な指導, プロジェクトの可視化, ソフトウェア工学教育

1. はじめに

IT を活用して高い付加価値を創造できる高度な情報通信人材を産学官連携により育成することの必要性が提言され⁽¹⁰⁾, 在学中にチームによるソフトウェア開発を実際に行うことを通じ, システムの実装技術だけでなく, プロジェクト運営に必要なマネジメントやコミュニケーションのスキルを身に付けることが産業界より求められている. そのための方法の一つとして PBL (Project Based Learning) 形式による実践的なソフトウェア開発演習 (ソフトウェア開発 PBL, 以降 SDPBL) が多くの教育機関で実施されている⁽⁷⁾. PBL 形式による教育は, 通常の座学による講義・演習だけでは習得の難しいプロジェクトの運用に必要なスキルを, 実際のプロジェクト運営を受講生が主体的に行うことを通じて獲得することを目的として実施されるものであり⁽²⁾, 指導者はチームの活動状況を把握しながら適切な助言を行うことが求められる⁽⁹⁾. 本稿ではタスク管理システムのデータに基づいてプロジェクトを可視化するツール⁽⁴⁾を受講生の活動状況の把握に用いた, SDPBL における適応的な指導手法を提案する. 本手法を用いて指導を実施した SDPBL の事例について説明し, 本手法の有効性について考察する.

2. ソフトウェア開発 PBL

2.1 概要

SDPBL には様々な目的があるが, 本稿が対象とする SDPBL は, 数名の受講生で構成されるチームによる活動を通じて, 座学により習得したソフトウェア工学やプロジェクトマネジメントの知識を確認し, 応用可能なスキルとして身に付けるために実施されるものとする.

2.1.1 指導方法と演習のフェーズ

PBL において指導者は先生ではなくファシリテータ (補助する人) と呼ばれ⁽⁹⁾, その指導方法は, 問題点を指摘しその解決方法を指示するような直接的な指導よりむしろ, チームの活動に対して自発的な問題点の発見と解決を促す適切な助言を行うといった間接的な指導が重視される. SDPBL は, 受講生の主体性の度合いに応じて 3 つのフェーズに分けることができ, それぞれのフェーズにおける指導方法は以下の通りである.

フェーズ 1 本格的な PBL に入る前の準備期間であり, 主にツールの使い方に関して直接的な指導を行うフェーズである. チームによるソフトウェア開発では一般に, チーム内のコミュニケーションを円滑にするために複数のツールが組み合わせて利用されるが, 開発を進めるにあたりこれらのツール

の使い方を受講生が覚える必要がある。また、演習においてツールの使い方に制約がある場合には、この期間に指導する。

フェーズ2 準備期間の後、受講生の活動に対して指導者が間接的な指導を行うことのできる期間である。受講生は間接的な指導により与えられた助言に基づいて活動の工夫・改善を行い、受講生による活動と指導者による助言を繰り返しながら演習が進められる。

フェーズ3 演習における最後の指導の機会から演習終了までの期間である。受講生による主体的な活動が中心であり、指導者による助言の機会はない。

2.1.2 演習形態

SDPBLにおけるチームによる開発作業は、設定された演習日に受講生が演習室等に集まって行われる場合と、演習日以外に受講生が各自のペースで行われる場合があり、演習によってこれらが組み合わせて実施される。本稿では前者を対面での開発と呼び、後者を遠隔での開発と呼ぶ。

2.1.3 指導方針

SDPBLはプロジェクトマネジメントの学習方法の1つであり、その指導はプロジェクトマネジメントの知識体系に基づいて実施される。プロジェクトマネジメントの知識体系の1つであるPMBOKでは、プロジェクトマネジメントにおいて実行されるべきプロセスを9つの知識エリアに分類している⁽¹¹⁾。

プロジェクト・スコープ・マネジメントは、プロジェクトの範囲を明確にし、管理するためのプロセスである。プロジェクト・タイム・マネジメントは、プロジェクトの納期や時間的な計画を管理するためのプロセスである。プロジェクト・コスト・マネジメントは、コストの見積もりや実績との差を管理するためのプロセスである。プロジェクト品質マネジメントは、プロジェクトの成果物が要求と一致していることを保証するためのプロセスである。プロジェクト人的資源マネジメントは、プロジェクトの要員の役割分担などを管理するプロセスである。プロジェクト・コミュニケーション・マネジメントは、プロジェクト内の情報伝達に関するプロセスである。プロジェクト・リスク・マネジメントでは、発生した場合プロジェクトの遂行に対して影

響を与える不確実な事象や状態であるリスクの管理と、リスクが発生した際の対応について扱う。プロジェクト調達マネジメントは、プロジェクトの遂行に必要な物品やサービスを取得するために必要なプロセスである。プロジェクト統合マネジメントは、これら全体のプロセスを調整する。

2.2 指導における課題

ソフトウェア開発をチームにより行う演習では、チームの成果物として容易に観測可能な開発プロダクトのみでは、納期に対するチームの進捗状況しか分からず、進捗遅れの原因を推し量ることは難しい。また、プロジェクト運営の計画性を重視した演習では、適切に作業が分担されているかどうかや、チーム内で意思の疎通がはかれているかなど、チームのメンバー個人に着目して状況を把握する必要がある⁽³⁾。さらにPBLにおいては、チームごと、個人ごとに状況が大きく異なるため、自己が対象かどうか不明瞭な全体に対する指導に対して、受講生が自身に対する注意喚起として意識しにくく個別の状況に合わせた指導が必要である。

チームのメンバー個人に着目して状況を把握することにより、チーム全体の開発状況を認識し、チームに対してメンバーの状況に合わせた指導を行うことを、本研究ではSDPBLにおける適応的な指導と呼ぶ。演習指導において学習者に合わせて適応的に行うことは当然のように思えるが、一方でPBLにおいてはチームとしての活動の成果が評価の対象となるため、メンバー個人の活動が指導の対象となりにくい⁽⁸⁾。

3. プロジェクトの可視化に基づく適応的な指導

前章において述べたように、SDPBLにおいては受講生各個人に対する状況に応じた適応的な指導が必要不可欠である。本章ではこれまでにSDPBLにおいて行われた適応的な指導実現に向けた取り組みを紹介し、本論文で提案する新しい指導の枠組みとその実現方法について述べる。

3.1 ソフトウェア開発 PBL における適応的な指導

適応的な指導を行うためには、常に開発チーム全体およびメンバー個人の状況把握も同時に行いながら状況に応じた適切な指導を行える態勢が必要である。企業からの講師をプロジェクトマネージャー（PM）として各チームに配し、受講生だけでなく社員の研修としても活用する事例⁽⁹⁾は、このような要求に対する一つの解であり、エキスパートを各チームに配置するという方法である。この方法は、演習の実施期間中、エキスパートである指導者が常にチームに関わる必要があり、対面だけでなく遠隔でも行われる SDPBL では指導者の負荷が高い。また、企業からの協力が常に得られるとは限らず、すべての SDPBL に適用できるわけではない、という問題がある。

これらの問題に対し、人的要因を軽減しながら適応的な指導を行う実践例として、専用の開発環境を通じてツールにより受講生の進捗把握を支援する iPBL⁽³⁾ や EtUDE⁽⁵⁾ などがある。iPBL では、WBS (Work Break-down Structure) を作成する専用の環境を通じて、受講生の活動の進捗把握を支援する。EtUDE では、専用の開発環境を用意しており、この環境により得られた開発ログの分析結果を受講生の評価に利用することができる。しかしながら専用の開発環境は受講生の活動を制限するため、受講生による工夫・改善の自由度を制限することにもつながる可能性がある。

一方、我々はこれまでに SDPBL において汎用的な開発ツール群を対象として開発状況の可視化を行うツール⁽⁴⁾を提案している。そこで、本稿ではこの可視化ツールを用いて受講生の活動の進捗把握を支援し、適応的な指導に活用する手法を提案する。

3.2 プロジェクトの可視化

本稿で想定するプロジェクトの可視化ツールは、チームのメンバー同士で情報共有を行うための環境としてソフトウェア開発に用いられる一般的なタスク管理システムの利用を想定した SDPBL を対象に、指導者によるプロジェクトの状況把握を支援するものである。

一般的なタスク管理システムとして想定している機能は、プロジェクトに対する納期の設定されたいくつかの開発単位（マイルストーン）への分割、マイルストーンの達成に必要なタスク（実装、レビュー、試験な

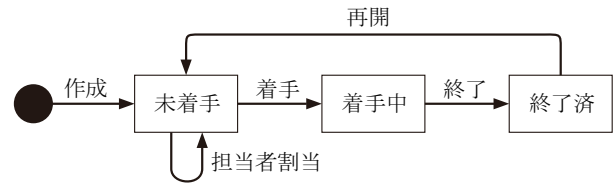


図 1: タスク管理システムにおけるタスクの状態遷移

ど)のタスク管理システムへの登録である。また、タスクは図 1 に示す状態遷移モデルに基づいて管理される。タスクの属性には作業の内容を表すタイトルのほか、見積時間や実測時間、種類、対象プロダクト、終了方法などを設定することができる。

タスク管理システムとして Trac を用いた可視化ツールの出力例を図 2 に示す。Trac に登録されたチケット (Trac において図 1 に示したタスクの状態遷移を管理する単位) の状況とその履歴を取得し、ガントチャート形式で視覚的に一覧表示するとともに、チームのサマリーとして各メンバーの状態や役割の分担状況を表示する。また、チケットに対するデータの入力不備などを指摘する。

3.3 対象とする知識エリアと目標

SDPBL では、PMBOK で定義されている 9 つすべての知識エリアを網羅することが望ましいが、実際の活動においては、人的、時間的要因により制約を受ける。そのため、一般にはあらかじめ知識エリアのいくつかを対象に絞った指導が実施される。同様に、本提案で行う適応的な指導が対象とする知識エリアはシステム開発において重要視される、QCD、すなわち品質 (Quality)、コスト (Cost)、納期 (Delivery) をふまえ、PMBOK における、プロジェクト品質マネジメント、プロジェクト人的資源マネジメント、プロジェクトタイムマネジメント、とし以下の目標を設定している。

- (D1) 品質を保证するためのレビューや試験といったタスクも十分に実施する
- (D2) タスクを適切に分担し特定のメンバーだけに作業が偏らないようにする
- (D3) 与えられた仕様を満たしたプロダクトを期間内に実装する

(D1) は品質に対する目標である。品質の保証にはさまざまな方法が考えられるが、ここではレビューや単体

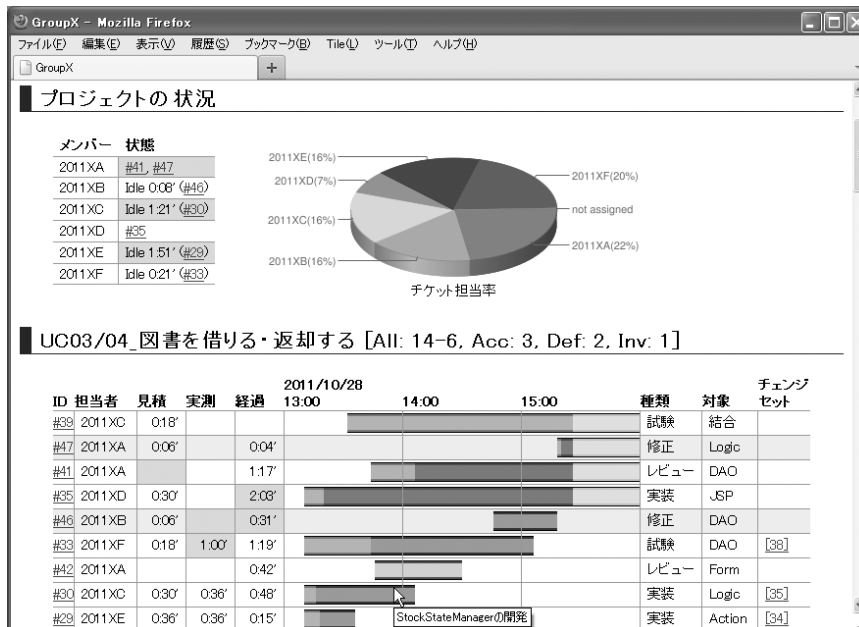


図 2: 開発状況可視化ツールの表示画面

試験, 結合試験を十分に実施することとしている (D2) はコストに対する目標であるが, 実際の開発で重視される金銭的な面を演習においてそのまま目標とすることは難しい. 実際のユーザとの契約を結ぶことによって, より現実的な開発を経験する PBL⁽⁶⁾ もあるが, 企業との連携が不可欠であり容易ではない. そこで本演習では適切な役割分担を目標として設定している. 一部のメンバーが徹夜で作業をすることによって実装されたプロダクトは, チームに割り当てられた人的資源が十分に活用できておらず, たとえ納期や品質を満たしていたとしてもコストに見合う開発が行われたとは言えない. また, 教育の観点から一部の受講生の学習機会を奪う可能性は避けるべきであり (D2) を意識したプロジェクトマネジメントによってチームのメンバー全員がさまざまな経験の機会を得られるようにする (D3) は納期に対する目標である. これらの目標を意識して計画的に開発を実施することにより, 受講生は座学で学習したソフトウェア工学やプロジェクトマネジメントに関する知識を確認する.

3.4 開発履歴の収集と適応的な指導

先に挙げた D1 ~ D3 までの教育目標に対し適応的な指導を実現するためには, 様々な観点から個人の開発履歴を収集し把握する必要がある. 本節では目標に対し必要となる開発履歴情報を示し, それらを元にどのような適応的な指導が行えるかを述べる.

D1: 品質 品質の観点においては, 演習において開発されるべきプロダクトを対象として, 行われるべき実装, レビュー, テストが適正に行われているかを判断する. 実際に開発されたプロダクトとそのプロダクトを対象として行われるべきであったタスクが存在するかを調べることで実現する. ここで, レビューはすべてのプロダクトにおいて, 実装や実装の修正が行われることに行われているかどうか, テストは単体テストが実行可能であるモジュールについては行われているか, と言う点に着目する.

D2: 分担 分担の観点においては, 各メンバーが担当したタスク種別, 作業時間の偏り, タスク割り当て計画の適正さについての分析を行う. どのような分担が適正であるかは PBL の方針により異なるが, 上記のデータを取得することにより作業時間が同等となる量的均等性, 行ったタスクに偏りがない質的均等性などの検証が可能である.

D3: 納期 納期の観点については, マイルストーン単位での納期目標に間に合ったか遅れたかを確認する. 目標に間に合わなかった場合や計画性に問題が見られた場合は, ガントチャートにおいて, 見積もりと実績にどの程度の乖離があるかや, 当初想定されていなかったタスクが発生しているかを把握し, その原因となる開発履歴をチームに提示

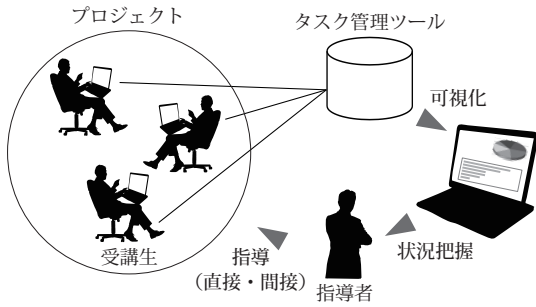


図 3: ソフトウェア開発 PBL における適応的な指導

することで振り返りを行わせる。

これらの情報をツールにより取得し、問題が起きている又は起きそうな要因を把握する。その後チームにおける PM などのリーダー的役割を担うメンバーに対し、指導者側で把握している状況を確認する質問を發することで、学生の主体性を失うことなく状況に合わせた指導を試みる。以上をふまえ、本提案で想定される SDPBL の概念図を図 3 に示す。

4. 指導事例

我々は、高度なソフトウェア技術者の育成を目的とした教育プログラム IT Spiral⁽¹⁾ において開講されている SDPBL 「開発実践演習」において、プロジェクトの可視化に基づく適応的な指導を試みた。本章では、開発実践演習の概要を説明するとともに、各チームの状況把握に利用したプロジェクトの可視化ツールについて述べる。

4.1 開発実践演習

IT Spiral では、関西圏 2 府 5 県の 9 つの大学院に在籍する、主に修士 1 年次の学生を対象として、各大学で行われる通常の座学の他に、受講生が一堂に会して行われる計 16 回（1 回 4 コマ、全 6 単位）の講義・演習を開講している。開発実践演習は、その最後の 4 回で実施している SDPBL である。

開発実践演習では、5~6 名で構成されるチームに分かれて、与えられた設計仕様書に基づき開発を行う。この演習で受講生が行うべきアクティビティは、コーディング、レビュー、単体試験、結合試験の各タスクおよびプロジェクトマネジメントである。2011 年度は 34 名の受講生が 6 つのチームに分かれて演習を行った。

表 1: 2011 年度の演習日程

演習日	形態	フェーズ	マイルストーン
10/21	対面	1	MS1
	遠隔	2	MS2
10/28	対面	2	MS3
	遠隔	2	MS4
11/11	対面	2	MS5
	遠隔	3	MS6 ~ MS8
12/02		成果報告会	

開発対象は Java EE に基づくウェブアプリケーションであり、受講生には 12 のユースケースに関する詳細設計仕様書と、実装の雛形となるソースコード群（2 つのユースケースは実装済み）が与えられる。本アプリケーションの実装には、プロジェクトマネジメントに関する知識の他に Java EE やフレームワークといった実装技術に関する知識が必要となるが、それらは座学等により習得済みである。

演習には、受講生が一堂に会して対面で開発を行う計 3 日の演習日と、最後に各チームが演習での工夫・改善点を報告する成果報告会が設定されている。2011 年度における演習日、成果報告会の日程、演習のフェーズおよび指導者によりあらかじめ設定されたマイルストーンを表 1 に示す。それぞれのマイルストーンには 1~2 つのユースケースが割り当てられ、決められた納期までに計画的に開発を実施することとしている。各演習日には指導者として 5 名の教員と 3 名の TA が演習を担当し、フェーズ 1 では直接的な指導を、フェーズ 2 に含まれる 2 日目および 3 日目の演習日の 1 コマでは中間講評としてそれまでの演習内容に基づく間接的な指導を行った。一方で遠隔での開発期間に対応できたのは主に 3 名の教員のみであり、それらの教員も常に対応できたわけではないので、各チームに 1 人の指導者を割り当てて状況を把握することは現実的ではなかった。

以降では、2011 年度の開発実践演習におけるフェーズ 1 およびフェーズ 2 の指導事例を示す。

4.2 フェーズ 1 における指導事例

フェーズ 1 では、3 名の TA が各チームを巡回する形で演習指導を行った。教員は、プロジェクトの可視化ツールを用いて各チームの状況を把握し、指導内容を

TA を通じて各チームに伝達した。TA は、教員からの指導内容を伝達する他、各チームからの質問を随時受け付け、場合によっては教員に確認を取りながら各チームに回答を行った。

今回の事例では、開発環境の使い方などの説明を除いた実質的な開発作業が行われたのは約 1.5 時間であった。その間に TA による巡回によって受け付けられた質問が 24 件、教員による可視化ツールを用いた状況把握に基づいて、チームに伝達された指導内容が 9 件であった。9 件の指導内容は、いずれも Trac の使い方の間違い（主に入力不備）に関するものであり、受講生はその間違いに気づいておらず、放置されればその後の演習にも影響を与えた可能性が高い。およそ 2.7 分に 1 件の割合で質問に対する回答や指導が行われ、回答時間を考慮すると今回の指導体制では常にいずれかのチームに対応している状況であり、可視化ツールによる支援がなければ 9 件の指導は難しかったと言える。

4.3 フェーズ 2 における指導事例

このフェーズでは中間講評という形でマイルストーン 2 つ分進捗が進むごとに、3.4 節で述べた方針に従って各チームを対象に指導者が品質、分担、納期の各観点に沿った指導を行った。ここでは、実際に行った指導内容を例に挙げる。

4.3.1 品質

表 2 に開発実践演習のあるマイルストーンで開発されたプロダクトとそのプロダクトを対象として行われるべきであったタスク、さらに実際に複数のチームでこれらのタスクが行われたかどうかの一覧を示す。例えば、view01.jsp は Web アプリケーションの表示に関するプロダクトであるため、テストは行わず、レビューを必須とする。

この表 2 では、○ は該当するタスクが行われたことを、× は全く行われなかったことをそれぞれ示す。また、△ は開発終了後にレビューでバグが見つかったようなケースで、一度目は行われたが、二度目以降レビューやテストが行われなかったことを示す。この例では、TeamA において view01.jsp、view02.jsp について、そもそもレビューが計画されておらず、DAO01.java および Manager01.java についてはバグ修正後のテストあるいはレビューが行われていなかったことが我々の可視化

表 2: チームごと実行済みタスク種別分析

プロダクト名	実行すべきタスク種別	TeamA	TeamB
Action01.java	レビュー		
DAO01.java	レビュー テスト		
Form01.java	レビュー テスト		
Manager01.java	レビュー テスト		
view01.jsp	レビュー	×	
view02.jsp	レビュー	×	
struts-config.xml	レビュー		

ツールによって確認された。TeamB については、一般的に必要なタスクが実行されていたが、struts-config.xml のみ、バグ修正後のレビューが行われていなかった。これらの指導を中間講評として各チームに行うことで、次回以降の中間講評では必要なタスクの実行漏れが全チームにわたって若干ではあるが減少したことが確認できた。

4.3.2 分担

表 3 は 4.3.1 節と同様 TeamA の各メンバーが担当したあるマイルストーン 2 つ分のタスク種別の数を示している。括弧はバグの修正等、開発を進めていった結果必要になったタスクの数を表している。例えば、メンバー 002 は 3 種類のプロダクトを実装したが、レビューでバグが検出された結果 7 回の追加実装を行っている。このチームでは、PM(005) のタスクを減らし、それ以外を均一化することを目的として、タスクの割り当てを計画した。

表からも分かるように、基本的に実装の分担はある程度均等になっていた。しかしながら、タスクの分担や割り当て計画について以下のような問題点が検出された。

- レビューを 001,006 しか行っていない
- テストを 003,006 しか行っていない
- 004 が複数のタスクを同時に実行しており、かつ、それらのタスクの遂行に 3 時間以上かかっていた
- 002 が複数のタスクを同時に実行していた
- PM の負荷を減らすはずが、ほとんどのテストを PM が行うことになっていた

指導者はこれらの問題点を分担に関する課題としてチームに提示を行い、振り返りを促した。結果、そのときの TeamA の振り返りとして、PM に負荷が集中す

るのを防ぐために、より積極的にレビュー以降のタスクを手が空いている人が行うよう心掛けるといったものや、他の人の進捗をより積極的に意識するといった項目が受講生らから得られた。

表 3: TeamA のタスク分担

メンバー	実装	レビュー	テスト
001	1	4(3)	0
002	3(7)	0	0
003	1(3)	0	2
004	2(1)	0	0
005	1	0	0
006	0	1	4(1)

4.3.3 納期

分析の過程では表 3 で示したメンバーごとのタスク種別数に基づいて、当初想定していなかったタスクがなぜ発生しているのか、また、図 2 のガントチャートにおいて、見積もりと実績にどの程度の乖離があるかに着目した。指導者が指摘した納期の観点における指導内容例を下記に記す。

- レビューで発見されたバグが、別のメンバーに 1 度修正された後、またレビューで指摘されていた (バグ報告の伝達ミス)
- 実装終了後、レビューが開始されるまでに 1 時間以上開いているケースが複数回存在した (ただし、この分析は分散開発を行っている場合は対象としていない)
- 1 度目のレビューで発見されたバグ修正後の 2nd レビューで、1st レビューで発見されるべきだったバグが新たに発見された
- タスクの見積もりが行われていない (同様のものとして、実績時間が入力されていない)
- レビュー終了後にレビューと無関係にバグが発見されている (タスクが行われた記録がない)
- 特定のプロダクトを対象に何度もバグが発見されている

タスクに対するデータ入力ミスといったケアレスに相当するものから、個人の開発スキルの問題で開発時間やバグ検出回数が想定よりも大きくなっているケース、コミュニケーションの不整合や系統だったレビューが行えていないために無駄なレビュー、バグ修正が行

われているケースなどが具体的に検出された。

5. 考察

5.1 受講生個人の技能が適応的な指導に与える影響

SDPBL において、チーム内のメンバーが持つスキルやモチベーションは均一でも一様でも無いことが多い⁽⁹⁾。そのため同じタスクであってもタスクの遂行に要する時間は変わってくる。既存の SDPBL の多くでは、各メンバーの得意なスキルに応じたタスク割り振りによって、納期や品質を確保することを目的としている⁽⁵⁾。結果として、SDPBL 終了後に各メンバーが得る経験の内容に偏りが生じてしまい、教育効果にメンバーごとの差が発生してしまう可能性がある。

IT Spiral においては、基本的には受講を希望した学生のみが参加しているため、開発実践演習に対する受講生のモチベーションは一様に高い傾向にある。しかしながら各メンバーがそれまでに学習したスキルが異なるため、開発スキルについては、同種のタスクであってもメンバーごとの遂行時間に倍以上の差があるケースが散見された。

そのため開発実践演習では、納期および分担の観点における適応的な指導として、見積もりを超過しているケースやタスク割り当てが過度に少なくなっているケースをメンバーに示し、振り返りを促した。結果として、タスク割り当てによって得意なタスクのみを各メンバーが担当するのではなく、極め細かく進捗を確認し、問題が発生していないかを継続的に確認するといった対応やタスク遂行に問題が発生した場合に周りがフォローする体制づくりを目指すといった対応をとるチームが現れた。

スキルやモチベーションの偏りに関する対応はチームを組ませる形式の教育手法にとって等しく困難な課題である。今後、我々の指導手法によって個人のスキル差を早期に確認できるというメリットを活かし、より良い指導手法の確立を目指していきたい。

5.2 教育目標を設定しなかった知識エリアに対する適応的指導の適用可能性

今回は対象にしなかった PMBOK の知識エリアのうち、プロジェクト・コミュニケーション・マネジメントについては、チームメンバー同士の開発に関する打ち合わせ、チーム内での質問といった文字ベースでの情報のやり取りを時系列に追って取得する枠組みを用意することで、チーム内でどのような方針を立てているか、どのようなスケジュール管理を予定しているかなどを把握することができる。プロジェクト・リスク・マネジメントは、チーム内で行われたリスクの表面化の発生を判断するための指標が個人の活動に影響を受けるため、ツールによってこれが可視化されれば適応的な指導を行うことができる。一方、プロジェクト統合マネジメント、プロジェクト・スコープ・マネジメント、プロジェクト調達マネジメントは、個人の活動で影響を受ける領域ではないため提案する適応的な指導を行うには向かないと考える。

6. おわりに

本稿では、SDPBL においてチームのメンバーそれぞれの活動状況からチームの状態を把握し、適応的な指導を実現する手法を提案した。プロジェクトの可視化ツールを用いることで、チームとしての開発プロダクトだけでなく、チームのメンバーそれぞれの活動プロセスからチームの開発状況を品質、分担、納期の3つの観点に基づいて把握し、指導する手法である。実際に本手法を用いて SDPBL のフェーズ1における直接的な指導およびフェーズ2における間接的な指導を実施し、その有効性を確認した。

参考文献

- (1) Barker, M. and Inoue, K.: “IT SPIRAL: A Case Study in Scalable Software Engineering Education”, In *Proc. 22nd IEEE-CS Conference on Software Engineering Education and Training*, pp. 53-60 (2009)
- (2) Batatia, H., Ayache, A. and Markkanen, H.: “Netpro: an Innovative Approach to Net-

work Project Based Learning”, *Proc. International Conference on Computers in Education (ICCE'02)*, pp. 382-386 (2002)

- (3) 中鉢欣秀, 土屋陽介, 長尾雄行, 加藤由花, 酒森潔, 戸沢義夫: “グループウェア導入による PBL の見える化”, *日本 e-Learning 学会論文誌*, 第 9 巻, pp. 129-135 (2009)
- (4) 福安直樹, 佐伯幸郎, 水谷泰治: “プロジェクトの可視化に基づく PBL 指導支援”, *ソフトウェア工学の基礎 XVIII, レクチャーノート/ソフトウェア学* 37, pp. 255-256 (2011).
- (5) Hashiura, H., Yamashita, K., Ishikawa, T. et al.: “A Software Development Group Exercise Support Environment, EtUDE: The System Overview and the System Evaluation Through Applying to Classes”, *Proc. of the 7th WSEAS International Conference on Software Engineering, Parallel and Distributed Systems (SEPADS'08)*, pp. 124-131 (2008)
- (6) 井上明, 金田重郎: “実システム開発を通じた社会連携型 PBL の提案と評価”, *情報処理学会論文誌*, 第 49 巻, 第 2 号, pp. 930-943 (2008)
- (7) 独立行政法人 情報処理推進機構: “IT 人材白書 2011”, (2011)
- (8) Matsuura, S.: “AN EVOLUTION METHOD of PROJECT BASED LEARNING on SOFTWARE DEVELOPMENT EXPERIMENT”, *Proc. of the 37th ACM Technical Symposium on Computer Science Education*, pp. 163-167 (2006)
- (9) 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*, 第 48 巻, 第 8 号, pp. 2767-2780 (2007)
- (10) 社団法人 日本経済団体連合会: “産学官連携による高度情報通信人材の育成強化に向けて”, (2005)
- (11) Project Management Institute: “A Guide to the Project Management Body of Knowledge (PMBOK GUIDE) — Forth Edition”, (2008)