

ソフトウェア開発PBL授業における進捗管理手法の評価

山本 美幸¹ 福岡 宏一¹ 清原 良三² 寺島 美昭¹

概要: 近年, 大規模ソフトウェア開発の分散開発を学ぶ, PBL 授業が重要視されている. そこで, 本研究は分散開発を想定した Project-based Learning(PBL) 型の授業を対象とし, 分散開発の独立性を妨げない非同期型の情報共有を用いて, 開発の遅延を解決するための進捗管理手法を提案する. PBL 型の授業では, Teaching Assistant(TA) が学生のサポートを行うが, 分散開発を想定する場合, TA と学生が地理的に離れた状況になる. そのため, TA にとって, 複数のプロジェクトや多数の学生の状況を監視する作業に時間がかかり, 学生への対応が遅れる可能性がある. そこで, 提案手法では非同期型の情報共有を開発に取り入れ, TA の開発進捗監視と分析, 助言の役割を自動化することで学生の遅延を早期に解決する. TA の役割を自動化するため, 開発進捗を測る指標として学生の遅延がプロジェクトに与える影響を数値化した深刻度を定義し, 深刻度に対応した助言内容を定める. 実験では, プロジェクトの開発進捗が学生の能力や作業効率に左右されるため再現することが難しく, 実験方法自体を考慮する必要があることから, 提案手法の効果を確認することに加え, 有効性評価方法の検討を行った.

Evaluation of Progress Management Method for Software Development Project-based Learning

MIYUKI YAMAMOTO¹ KOICHI FUKUOKA¹ RYOZO KIYOHARA² YOSHIAKI TERASHIMA¹

1. はじめに

本研究では, 分散開発を想定したソフトウェア開発型の Project-based Learning(PBL) 授業において, プロジェクトの遅延を抑えることを目的とした進捗管理手法を提案する. 対面の PBL 型授業では, 学生の能力, 作業効率, コミュニケーション力等の要因によって作業遅延が発生することがあるため, プロジェクトの進行状況を監視し, 助言を行う支援者として, Teaching Assistant(TA) が存在する. TA の役割として, 学生の開発中の問題に対応し, プロジェクトの遅延を早期解決に導き, プロジェクトを円滑に進めることが挙げられる. しかし, TA は, 複数の学生の状況を監視しなければならない, 対応が遅れたプロジェクトで開発遅延が拡大する恐れがある, 提案手法は, TA の役割に着目し, 開発の独立性を妨げないために学生間のやりとりで非同期型の情報共有を取り入れ, その情報からプロジェクト状況を監視, 分析, 助言内容を判断する. 提案

手法を授業に取り入れることで, TA が行う役割を自動化することができ, 学生の開発遅延の問題を早期解決に導くことが期待できる. 本稿では, 2, 3 章で対象授業の背景と提案手法について述べ, 4 章で有効性評価の方法を検討するための実験について述べる.

2. 研究背景

文部科学省は, 情報技術人材育成のための事業で, プロジェクト型の教育を推奨している [1]. そのような事業の中でも, 分散 PBL が取り入れられていることから, 大規模ソフトウェア開発の分散開発を想定した PBL 型の教育が重要視されている [2]. 対象とする授業では, 分散開発を想定するために, 学生が独立して開発を行い, 対面で開発をせず定期的に進捗報告を提出することで, プロジェクトを進度を把握し, 開発を進める. 授業の課題として, 分散開発において発生する課題に加え, PBL 型の授業特有の課題も発生する.

2.1 想定する授業の流れ

授業の前提を次の通りに設定する. 授業は半年間のプロ

¹ 創価大学大学院 工学研究科

² 神奈川工科大学 情報学部 情報工学科

グラムであり、一つのプロジェクトは、3 から 5 名程度の学生で構成され、一つのソフトウェアを完成させる。分散開発を体験する目的があるため、対面での開発は原則行わない。開発開始時は、プロジェクト計画を立て、作業計画書を作成する。作業計画書では、各学生が担当する作業と作業期限日、作業量の情報を記載する。作業単位は、一週間でいう作業を一作業とする。

2.2 ソフトウェア分散開発の現状

分散開発において、プロジェクト状況がプロジェクトメンバーに伝わらず、問題発生時の対処が遅れるという現状がある。従来型の開発形態では、プロジェクト状況の確認を進捗報告書を用いて行うため、問題発生から解決策を議論し、対策をとるまでに時間がかかる。さらに近年は、IoT 開発等に用いられるスパイラル開発が増加傾向にあり、こうした開発では、仕様変更による作業遅延が頻繁に発生するため、遅延に対し、迅速な対応が求められる。

これまで、開発者の行動や進捗状況可視化による進捗管理の負担軽減を行う研究が盛んになされてきた [3][4]。このことから、情報共有や進捗管理、進捗管理負担軽減の重要性が伺える。しかし、ソフトウェア分散開発型 PBL 授業に適用した場合、協調して開発を進めるため、学生個人に遅延の程度を認識させ、解決に誘導する点が不十分である。

2.3 ソフトウェア開発 PBL 授業の現状

ソフトウェア開発 PBL 授業における TA は、複数のプロジェクト状況を把握し、状況に応じた助言を行うことで、プロジェクトが円滑に開発に進むように支援する。しかし、プロジェクトごとに学生の能力や進め方が異なるため、プロジェクトの進捗を正確に把握することは難しく、複数のプロジェクトの監視に負担がかかる。福安らは、ソフトウェア開発 PBL におけるチームの状況に合わせた適切なフィードバックが重要と考え、指導者の評価負担の軽減を目的に、状況把握支援環境のツール開発と、3 つの開発フェーズに合わせた適切なフィードバックについて検討している [5]。このような、プロジェクト評価に関する手法の提案は様々に行われている [6][7]。しかし、本研究では、学生の評価方法ではなく、TA の立場で、学生への助言を自動化し、学生のプロジェクト遅延を早期解決するという目的の部分が異なる。そして、ソフトウェア分散開発型 PBL 授業に有効な進捗管理手法は明らかにされていない。

TA の負担軽減に関する研究では、プログラミング講義に関するものが多く、プロジェクト開発を想定していない状況で、コーディングの進捗状況を管理することを支援する研究 [8] や、TA が巡回する際に学生に声をかけやすくする研究 [9] などがある。しかし、プロジェクト開発では、個人の遅延だけでなく、プロジェクトに与える影響を考慮する必要があるため、既存手法を適用した場合、適切なタイミングで助言を行う点が不十分となる。

3. 提案手法

3.1 研究目的

本研究では、分散開発を想定したソフトウェア開発 PBL 授業において、学生の開発プロセス遅延の拡大を防ぐことを目指し、TA の役割について着目し、TA の役割の自動化を行う。TA を自動化する上で生じる課題として、次の 2 点を設定する。

課題 1 助言するタイミングを、プロジェクトに与える学生個人の遅延の程度を考慮して、判断する。

課題 2 プロジェクトの遅延原因を分析し、遅延の状況に適した助言を選択する。

3.2 進捗管理手法

課題 1, 2 に対し、次のように対処する。

解決 1 他学生の状況を数値化した”深刻度”を用いて、遅延がプロジェクトに与える影響を可視化し、助言のタイミングを判定する。

解決 2 プロジェクトの進捗状況と開発中の発言情報から、web の情報や過去事例を検索し、助言内容を選択する。2 点の解決を実現するため、非同期情報共有と 3 段階支援モデルで構成される進捗管理手法を提案する (図 1)。非同期情報とは、時間にとらわれずに発言ができる SNS や web 掲示板等の情報である。非同期情報共有を提案手法に取り入れる目的は、学生の独立した開発を妨げることなく、進捗報告や学生間の議論を可能にし、情報の履歴を活用するためである。3 段階支援モデルは、非同期情報を活用し、助言のタイミングと内容を判断することにより、TA の役割を自動化するためのモデルである。

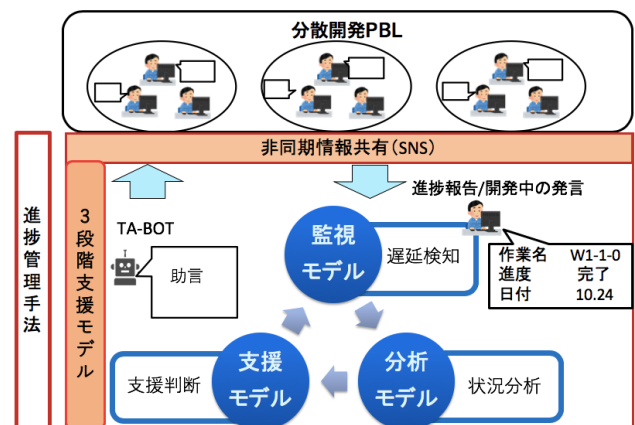


図 1 分散開発 PBL と進捗管理手法

3 段階支援モデルは、「監視」「分析」「支援」の 3 項目から構成される。第一は「監視モデル」である。これは、進捗を管理し、遅延の早期発見と、遅延の影響を予測するためのモデルである。第二は「分析モデル」である。進捗状況を分析し、他の学生の状況を踏まえた個人の遅延を深刻度として計算する。第三は深刻度に応じて支援 LEVEL を設定

し助言を自動化する”支援モデル”である。この3項目のサイクルを繰り返すことで、遅延検知や問題解決をスムーズに対処する工程を支援し、課題解決につなげる。手法を実装するためのシステム構成を図2に示す。

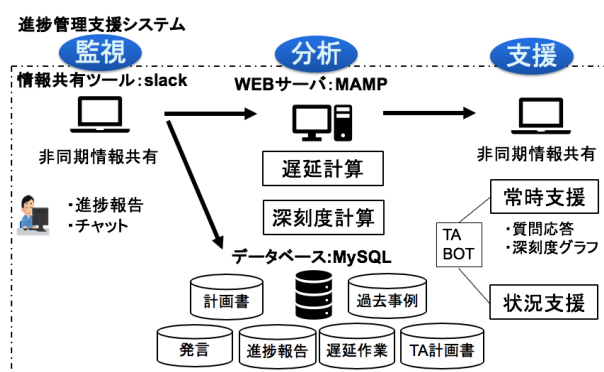


図2 進捗管理支援システム構成図

3.3 監視モデル

監視モデルでは、非同期情報共有における発言を常時取得し、発言監視を行い、プロジェクトの遅延を検知する。監視する発言は、開発に関する発言と進捗状況に関する進捗報告の2種類である。プロジェクトの遅延を検知するためには、進捗報告とプロジェクト開始時に決めた作業計画書の内容を用いる。遅延検知計算では、進捗報告の作業番号、作業量、進捗度と、作業計画書の作業番号、作業量を使用する。進捗情報の作業量を T_n 、作業計画書の対象作業の作業終了日までの作業量を T_p とする。また、TA が遅延判断の閾値を T_s としてする。進捗情報登録時に、進捗情報の進捗度が「未完了」の作業を対象とする。 T_n と T_p とを比較し、 T_s より小さければ遅延と判定する（式1）。

$$(T_p - T_n) > T_s \quad (1)$$

進捗報告以外の発言情報の履歴を管理し、活用する目的は、2点ある。1点は、非同期情報共有上で議論された遅延原因の解決策等を、今後同じような問題が発生した場合に再利用するためである。2点は、学生がTA に対して質問する場合、質問に対して自動応答するためである。

3.4 分析モデル

分析モデルでは、監視モデルにて出力された遅延作業情報を用いて、作業ごとに遅延による影響を計算し、プロジェクトの深刻さを分析する。従来のように、作業単独の遅れだけでなく、他の学生の影響を加味したプロジェクト全体の状況を表すことが狙いである。

そのプロジェクトの深刻さを表す指標として、“深刻度”を定義する。深刻度を定義することで、プロジェクト全体に対する助言だけでなく、各学生の進捗に合わせた助言を行うタイミングを測ることを可能にする。深刻度は、作業の遅延量とプロジェクトに対する作業者の能力から計算する。プロジェクトに対する作業者の能力は、プロジェクトに対する各学生の担当作業量の比から決め、作業計画書の

作成時に設定する。 n は進捗報告の回数、 T_n は進捗報告日、 T_{np} は、 T_n で計画されている作業量を表す。 T_{na} は T_n における実際の作業量であり、 W_{po} は計画された対象の学生の作業量である。 W_{pe} は、プロジェクト全体で計画されている作業量を表す。計算方法を式(2)に示す。

$$\frac{T_{np} - T_{na}}{T_{np}} \times \frac{W_{po}}{W_{pe}} \quad (2)$$

深刻度に対応した助言の支援を、5段階の“支援 LEVEL”として定義する。例えば、深刻状況が軽度な遅延である場合には、LEVEL1 と設定する。深刻度を入力として出力される主な遅延原因と支援 LEVEL を整理し、表1に示す。レベルが上がるほどプロジェクトの状況は深刻になる。LEVEL1, LEVEL2 は、作業の遅延が、軽度の遅延である。遅延原因は、学生個人の知識、経験不足による作業効率の問題と判断する。LEVEL3 は、遅延原因を早期対処する必要のある状況である。LEVEL4 は、プロジェクトの学生と協力し、早急に遅延を対処しなければならない、深刻な状況である。LEVEL5 は、全ての対策を講じて、間に合わない遅延であり、計画の見直しが必要な状況である。

表1 深刻度と支援 LEVEL の対応

深刻度	支援 LEVEL	遅延原因
1-15	LEVEL1	個人の作業効率
16-30	LEVEL2	個人の作業効率
31-50	LEVEL3	技術的スキル
51-70	LEVEL4	複数人の作業効率
71-100	LEVEL5	作業計画の見積もり

3.5 支援モデル

支援モデルは、従来の対面型 PBL 授業において、TA の役割であるプロジェクトに対する助言を自動化するため、非同期情報共有ツール上で Bot（以下、TA-BOT）を用いて、プロジェクト状況に応じた支援を行う。

支援の種類は、(A) 常時プロジェクトに対して支援を実行するものと、(B) 支援 LEVEL に応じて実行するものの2種類がある。(A) の常時行う支援は、次の2点である。

- TA に対する質問応答
- 状況の深刻さを可視化

1点目のTA に対する質問応答では、学生が発言した質問の内容から、過去の質問履歴と回答履歴から検索し、回答するものである。回答の提示を許容するタイミングは、プロジェクト開始前に設定する。この支援により、学生に問題が発生した際に、迅速に解決に導く効果が期待できる。

2点目の状況の深刻さについては、図3のように深刻度をグラフ化し、プロジェクト内の他学生と個人の作業がプロジェクトにどの程度影響するかを簡単に把握し、認知させる。例えば学生3の作業が順調に進んでいたとしても、学生1、学生2が大幅に遅延した場合、プロジェクトの作業を結合する際に、学生3は他の学生の影響を受け、共に深刻な状況になると予測できる。

(B) の支援 LEVEL に応じて行う支援では、プロジェク

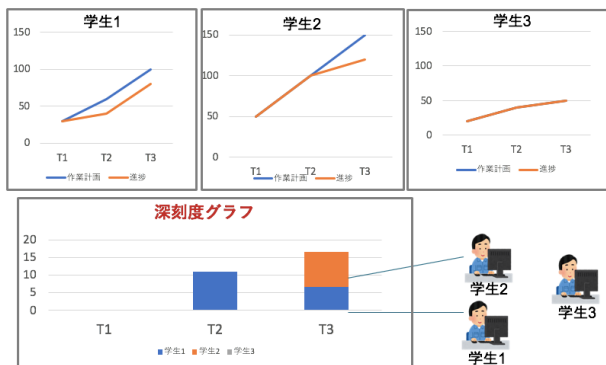


図 3 深深度グラフの可視化

ト状況に合わせ、適した助言をすることを目的とする。プロジェクトの状況は大きく分けて3点あり、(a) 他学生に影響を与えない程度のクリティカルパス上にない遅延、(b) 他学生に影響があり、技術的な原因の遅延、(c) 他学生に影響があり、計画見積もりが原因の遅延である。(a)の状況で従来のTAは、プロジェクト内で遅延情報を共有し、問題の対処をプロジェクトで解決するように助言する。(b)の状況では、技術的な原因に対する解決策を助言する役割がある。(c)の状況では、計画の見直しを助言する。(a)は支援LEVELのLEVEL1, LEVEL2, (b)はLEVEL3, (c)はLEVEL4, LEVEL5の状況に相当する。これらのTAの助言をTA-BOTを用いて自動化するため、次の3点を行う。

- 遅延作業・影響作業に遅延情報を通知
- 遅延作業に遅延作業の解決策を提示
- 今後の遅延作業予測とプロセス再構築を助言

(a)の状況では、学生個人の遅延で、プロジェクトの他学生に影響がない状況のため、遅延した学生に対して遅延状況を認識させることと、今後遅延し続けた場合に影響を受ける学生への遅延状況に関する情報共有が必要である。そのため、TA-BOTは、遅延作業と遅延作業、深深度をプロジェクトメンバに通知する。

(b)の状況では、遅延の影響が他の作業に関わる、早期解決が必要な状況であるため、遅延作業に対して解決策を助言する必要がある。ここでは、TA-BOTによる助言として2種類用意する。1点は、過去の履歴を検索し、解決策を提示する支援であり、2点は、進捗報告の内容と作業計画書の作業内容から、web上の情報を検索し、結果を通知する支援である。LEVEL3の支援例を図4に示す。

(c)の状況は、遅延の影響が深刻であり、計画の見直しが必要な段階である。そのため、支援ではTA-BOTがプロジェクトに対し、計画見直しを促すとともに、現状のまま作業を進めた場合の遅延予測状況を表した、深刻状況グラフをプロジェクトに通知する。

(a), (b), (c)それぞれの支援内容と支援LEVELの関係をまとめたものを表2に示す。この図の「○」は、支援のタイミングを表す。



図 4 LEVEL3 の状況の例と支援

表 2 支援 LEVEL と支援内容

(B)	支援内容					
	(a)		(b)		(c)	
支援 LEVEL	遅延情報	影響情報	Web 情報	過去事例	状況グラフ	計画見直し
LEVEL1	○					
LEVEL2	○	○				
LEVEL3	○	○	○	○		
LEVEL4	○	○	○	○	○	
LEVEL5	○	○			○	○
効果	遅延認知		技術課題解決		作業効率向上	

4. 実験

本研究では、今後3段階の実験を予定している。第1では、TA-BOTが助言した場合の開発進捗変化を知るために作業が遅延するケースを設定し、助言の効果を確認する。同時に、作業の遅延原因と遅延解決に必要な助言内容を明らかにする。この情報に基づいて深深度や助言内容を再検討し、第2実験では、作業時間に余裕をもたせ、一作業の期間が想定授業に近いケースで実施する。実施結果から、非同期型情報共有上の会話内容と開発進捗との関係を分析し、深深度と助言内容の妥当性を検証する。第1, 第2実験では、例えばメンバに極端な進捗の差がある場合や、だんだんと悪化してゆく場合、改善されてゆく場合等の状況が確認できない可能性も考えられる。そのため、最終的に進捗情報を自動生成し、現実で起こりにくいプロジェクトの状況も再現して、深深度による助言タイミングの妥当性について多くのケースを検証する計画である。この現実(実験1, 2)と仮想(シミュレーション)を組み合わせる方法により、一般に時間もコストもかかり再現が難しく、検証が困難な本提案手法の有効性を確認する計画である。本稿では、第1実験について述べる。

4.1 実験内容

本実験は、第1段階の実験であり、作業が遅延した際のデータを収集するため、作業が遅れるよう作業期間を短く設定する。今回は、想定授業の作業単位である1週間を、10分と仮定し、3期間分の開発を行うこととした。期間は、Term1, Term2, ..., TermNと表記する。今回は、一つのプロジェクトにつき2, 3人のプロジェクトとし、4つのプロジェクトを用意した(表3)。

プロジェクトを用意する際に、各プロジェクトの能力差

を均等にするため、事前アンケートを行った。事前アンケートでは、使用したことがある言語と一番得意な言語、そのレベルの自己評価を3段階で記入させた。事前アンケートの結果から、被験者はC言語を学んだ経験があり、全員授業レベルであると判断し、プロジェクトの題材としてC言語のプログラムを用意することとした。また、作業の完了/未完了を判断しやすくするため、各作業には、web上のプログラム問題[10]を用いた。作業計画書は、1つのプログラム問題を1作業と設定し、作業量は、事前にTAが作業した際にかけた時間として設定した。

各プロジェクトには、比較のため同じ作業計画書を渡し、実験を行った。作業計画書と作業関連図を図5、表4に示す。開発中のやり取りと、進捗報告を行う非同期情報共有ツールには、Slackを用いた。発言数変化や、進捗報告のデータを観察しやすくするため、プロジェクトごとにチャンネルを作成し、発言するスペースを限定した。今回は、45分間で3期間分の作業を行い、10分経過につき、5分間休憩を挟み、10分が経過しても作業が完了しない場合は、5分間の休憩のタイミングで進捗を必ず入力させた。実験時のルールは次の通りである。

- 作業完了次第、進捗を必ず入力する
- 口頭での会話は認めない
- インターネットを用いて検索してもよい

表3 プロジェクト一覧

	X	Y	Z	TA-BOT	被験者
プロジェクトA	AX	AY	AZ	あり	学部3年
プロジェクトB	-	BY	BZ	なし	学部3年
プロジェクトC	CX	CY	CZ	なし	学部4年
プロジェクトD	DX	DY	DZ	あり	学部4年

表4 作業計画書

作業番号	作業名	作業者	進捗報告	作業時間	作業量
D027	nまでの和	X	Term1	10分	5
D091	花粉の予報	X	Term2	10分	8
D011	アルファベットで何番目	X	Term3	10分	10.5
D047	メダリストの表示	Y	Term1	10分	5
D068	雨と晴れの記録	Y	Term2	10分	9
D075	足りないカード	Y	Term3	10分	11
D055	ワインのキャッチコピー	Z	Term1	10分	5
D048	台風の間隔	Z	Term2	10分	10
D113	初日の出	Z	Term3	10分	12

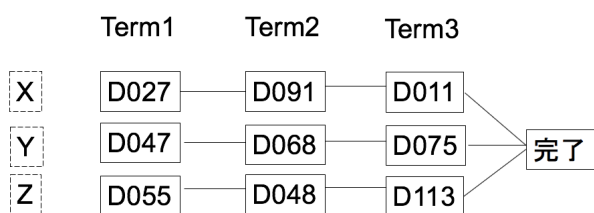


図5 作業関連図

4.2 開発状況結果

4つのプロジェクトで実験を行なった結果、全てのプロジェクトにおいて作業に遅延が発生した。また、被験者11名中9名が、作業1を未完了で終了した。作業1が完了し、作業2の途中で終了した学生は、プロジェクトB、Dの学生のそれぞれ1名であり、どちらもTA-BOTによる助言をしないプロジェクトの学生であった。深刻度グラフの例としてプロジェクトAの結果を図6に示す。

進捗報告を除いた会話や質問に関する発言数については、被験者が学部3年のプロジェクトでは発言がなく、学部4年のプロジェクトでは発言が確認できた(図7)。プロジェクトDでは、Term1の直後に発言があり、それ以降の発言はなかった。プロジェクトCは、Term1で作業に関する質問があったため、TA-BOTがweb情報を通知した。続くTerm2でも、作業の遅れが解決できなかったことから、Term3でさらに作業に関する具体的な質問があった。

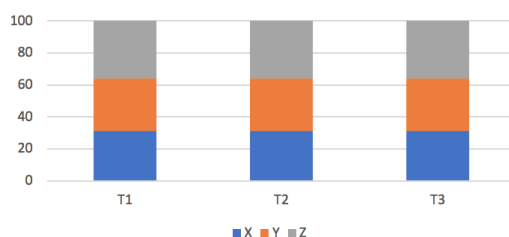


図6 プロジェクトAの深刻度グラフ

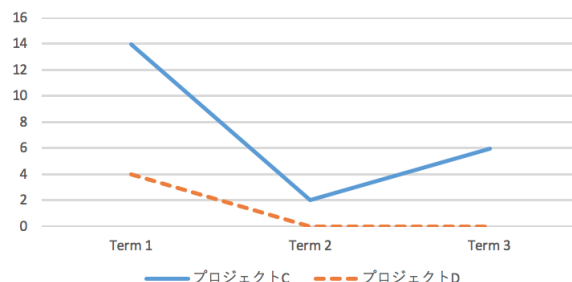


図7 発言数推移

4.3 アンケート結果

被験者の主観評価のために、事後アンケートを行った。アンケート項目は、(1)プロジェクト内の学生の進捗を意識したか、(2)何が原因で遅れたか、(3)どのような助言があれば作業効率の向上につながるか、(4)TA-BOTの助言は役立ったか(プロジェクトA、Cのみ)の4項目である。

(1)について、プロジェクトB、Dは、「意識しなかった」と回答した被験者が1名ずついたことに対し、プロジェクトA、Cは「意識しなかった」と答えた被験者がいなかった。このことから、遅延情報の通知や深刻度グラフを通知する支援によって、プロジェクト全体の進捗を意識しながら開発をする効果があったといえる。

(2)について、実験前に想定していた「知識・技術的スキル不足」が原因で遅延した、と答えた被験者は、被験者11名のうち9名であった。その他の回答としては、「エラー

処理」が2名、「問題理解」、「入力スピード」が各1名いた。「知識・技術的スキル不足」が原因で遅延した被験者の多くが、「検索に時間がかかった」と回答しており、「知識・技術的スキル不足」の中でも検索スキルが作業効率に大きく関わるということがアンケートから得られた。

(3) について、特に目立った回答は「ソースコードそのもの」に関する回答であり、その他には「遅延状況をより細かく把握し対応する」、「学生間の会話を活かすためのTA-BOTの助言の工夫」の回答を得た。そして、(4) について、TA-BOTが役立たなかったと回答した被験者は、6名中2名であった。

4.4 評価・考察

実験から、学生の開発中の発言内容や進捗状況、アンケート結果を確認することで、提案手法の改善点と有効性評価のための実験内容の改善点が見つかった。提案手法に関しては、遅延原因に対応する深刻度の数値を見直し、支援モデルの助言内容について検討する必要があることがわかった。プロジェクトCの結果からは、進捗報告後にTA-BOTが助言しても、遅延原因の解決には至らなかったため、助言のタイミングと助言内容の適切さが欠けていたと考えられる。しかし、TA-BOTの助言をしないプロジェクトに比べて、助言したプロジェクトは解決のための発言数が多いことから、TA-BOTの助言が、遅延作業者に作業の遅れを自己認識させ、解決のための議論を促す効果を確認できた。

助言内容の適切さについては、アンケート項目(3)、(4)の結果からも、内容を見直す必要があることがわかる。特に、アンケート項目(2)から、助言のタイミングを決めるための遅延原因として、これまで定義していなかった「検索スキル」や「エラー処理」について考慮する必要がある。これらから、分析モデルでは、想定する遅延原因の項目を増やし、これまでより早い段階で助言を行うことができるよう、深刻度の数値を見直す。そして、支援モデルでは、より学生個人の状況に適した助言内容にするため、学生個人のスキルや開発で苦手と感じることを事前に把握し、その情報を用いて、webサイトや過去事例から助言内容を選択するといった改善が必要である。

実験内容に関して、45分間の実験終了時点で、被験者11名のうち9名が一つ目の作業を完了しておらず、進度の差が確認しづらいという点から、事前に用意した作業の設定が適切でないと考えられる。そのため、次回は事前アンケートにプログラミング能力を測るためのテストを追加し、その結果を考慮した作業計画書を用意する必要がある。また、第1段階の結果からは、作業期間を短縮したため時間の制約が厳しく、深刻度の変化や非同期型情報共有の会話の傾向がわかっていない。この点については、当初の予定通り第2実験で、1週間単位でプロジェクト実験を実施し、作業期間に余裕がある状態で、開発時の非同期情報共

有の発言内容や開発進捗の傾向を観察する。

5. おわりに

本稿では、分散開発を想定したソフトウェアPBL授業において、プロジェクトの遅延を軽減するための進捗管理手法を提案した。提案手法では、TAの役割に着目し、助言のタイミングと助言内容の選択を自動化することを目的とし、深刻度を特徴とする3段階支援モデルを定義した。プロジェクト実験では、実際の開発遅延が起きた場合の進捗変化や遅延原因を観察し、助言の効果の確認と実験方法の検討を行った。実験の結果からは、TA-BOTの助言が遅延解決のための議論を促すという効果を確認できた。しかし、助言のタイミングをより早くするために深刻度の数値を調整し、助言内容の選択で学生個人のスキル情報を活用することが必要であるとわかった。実験内容に関しては、作業計画書の作成段階で事前のテスト等により、被験者のスキルを詳細に把握することが必要である。今後は、今回の結果を踏まえたプロジェクト開発実験を行うことに加え、多数の開発状況をシミュレーションし、最終的に長期的なプロジェクト実験にて手法の有効性を評価する。

参考文献

- [1] 成長分野を支える情報技術人材の育成拠点の形成(enPiT), 文部科学省(オンライン), 入手先<http://www.mext.go.jp/component/a_menu/education/detail/_icsFiles/afieldfile/2017/03/30/1383653.8.pdf> (参照2019-05-10).
- [2] 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(enPiT), 文部科学省(オンライン), 入手先<http://www.mext.go.jp/component/a_menu/other/detail/_icsFiles/afieldfile/2015/07/01/1358704.08.1.pdf> (参照2019-05-11).
- [3] 大平雅雄, et al. “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.” 電子情報通信学会論文誌D 88.2 (2005): 228-239.
- [4] 玉田春昭, et al. “プロジェクト遅延リスク検出を目的とするソフトウェア開発プロセス可視化ツール: ProStar.” 奈良先端科学技術大学院大学テクニカルレポート, NAIST-IS-TR2007002 (2007).
- [5] 福安直樹, et al. “チーム内の役割分担を考慮したソフトウェア開発PBLの評価基準と状況把握支援.” 電子情報通信学会論文誌D 98.1 (2015): 117-129.
- [6] Paasivaara, Maria, et al. “Teaching students global software engineering skills using distributed scrum.” Proceedings of the 2013 International Conference on Software Engineering. IEEE Press, 2013.
- [7] Kilamo, Terhi, Imad Hammouda, and Mohamed Amine Chatti. “Teaching collaborative software development: A case study.” Software Engineering (ICSE), 2012 34th International Conference on. IEEE, 2012.
- [8] 井垣宏, et al. “プログラミング演習における進捗状況把握のためのコーディング過程可視化システムC3PVの提案.” 情報処理学会論文誌54.1 (2013): 330-339.
- [9] 市村哲, 梶並知記, and 平野洋行. “プログラミング演習授業における学習状況把握支援の試み.” 情報処理学会論文誌54.12 (2013): 2518-2527.
- [10] paiza ラーニング: プログラミングスキルチェック, Gino, 入手先<<https://paiza.jp/challenges>> (参照2019-05-10).