M-021

学生のソフトウェア開発実習におけるコラボレーション支援に関する考察

Collaboration Support Tool for Practical Software Development Experiments

栗原 紘樹†　　松浦 佐江子‡
Hiroki Kurihara　　Saeko Matsuura

１．はじめに

本学では 2002 年度より、Java・UML 等のオブジェクト指向開発による、グループワークの実践的ソフトウェア開発実習を実施している（以下、情報実験と呼ぶことにする）。

我々は、開発作業支援・コミュニケーション支援・学生の評価を行うための情報収集を目的として、グループワーク支援システムを開発し、2003 年度より情報実験に適用している。本システムではメンバー間のコラボレーション支援を行っているが、利用者の声や利用状況から見て、問題点や作業状況把握に役立っているとは言い難い点がある。

本稿では、これまでのログ・利用者の声を元に、機能・情報の使われ方を、学生・教員の視点から分析し、目的のために修正すべき点を明らかにする。さらに、その改善案について考察した結果を報告する。

２．グループワーク支援システム

グループワーク支援システムは、以下４項目の支援を目的としている[1]。

●授業時間外の学生グループ内コミュニケーション支援
●授業時間外の教員と学生個人ならびにグループとのコミュニケーション支援
●学生の授業参加への意欲の向上
●教員の学生の作業状況の把握

また、グループワーク支援システムは、上記の項目を実現するために以下の機能をもつ。

(1)作業計画書管理

本機能は作業計画書の作成および作業計画書のバージョン管理を行うものである。役割分担、作業項目、スケジュール、コメントを入力することができる。各フェーズにおける最終目標を達成するために行うべき作業を「タスク」と呼び、作業項目はタスクのインスタンスのことを指す。任意に作成し、開始日、終了日、担当者を決めてスケジュール管理を行う。

(2)作業報告書管理

本機能はグループではなく、個人で担当した作業項目に対する作業報告と自己評価を行うものである。作業項目ごとに担当者が進捗度、自己評価、作業ステップ、作業時間、考察、問題点を入力し、報告する。また、閲覧も可能である。

(3)議論内容管理

本機能は電子掲示板を用いてグループメンバー間の議論を行うものである。掲示板の種類はグループ全体用、議事録用がある。教員、管理者は全てのグループの全掲示板を閲覧することができる。

(4)教員・管理者との連絡・質疑応答管理

本機能は学生が電子掲示板を用いて教員、システム管理者と連絡を取るためのものである。教員から指定された報告や管理者への連絡、質問を本機能によって行うことになる。また教員・システム管理者が学生に向けて連絡を送るときも本機能で行う。

(5)ファイル共有管理

本機能はアップローダを用いてファイル管理を行うものである。ファイル管理用のアップローダは作業計画書で作成された作業項目ごとに生成される。学生は対応した作業項目のアップローダに成果物を投稿し、グループ間で共有する。

(6)インスペクション管理

本機能は、特に検査が必要と思われる特定の種類の成果物に対して、成果物の検査を非対面で行う際に使用するものである。指定された成果物を作成しアップローダに投稿した後複数のグループメンバーで成果物が正しいことを検査する。

３．支援システムの分析

３.１　個人作業における「問題点」の記録場所

情報実験での話し合いは、授業内外での対面の他、掲示板やインスペクション機能で行われる。その際、その時々で様々な問題点が出てくるものである。その問題点についてはその場で行うか、後で話し合う場合にも、対面の場合は議事録に、掲示板やインスペクション機能であればそのまま記録として残しておくことができる。

それに対し、個人作業で出てきた問題点の記録に関しては特に明確な機能はなく、作業報告書の「問題点」項目に記録することが多い。しかし、経験上、作業報告書に書かれた問題点が議論されることはほとんどない。

そこで、2006 年度のログから、作業報告書の「問題点」項目に書かれた内容が次の話し合いの場で議論もしくは解決された記録があるかを集計した。授業が１週間ごとなので、「次の話し合いの場」までの期間を１週間以内とし、作業報告書・掲示板・議事録・インスペクション機能のいずれかに記録があるか否かで判断した。

以下はあるグループの結果であるが、未議論・未解決 22、議事録 15、掲示板８、作業報告書３、インスペクション０となり、半数近くの問題点が未議論のままになっているということがわかった。ただし、全ての問題議論を記録しているとは限らないため、未議論・未解決の数は 22 より少ない可能性がある。

また内容をみると、「クラスの責務の理解がそれぞれ違い、話が進まない」「ドキュメント規約を守っていない人がいるので、見て作成するようにしていくべき」など、個人のものだけではなく、グループ全体に関わるような問題のものもあった。

†芝浦工業大学大学院,Shibaura Institute of Technology graduate school engineering research course

‡芝浦工業大学, Shibaura Institute of Technology

３.２　インスペクション指摘数と担当者の関係

現在の仕様では、作業項目作成時に担当者を設定し、作業項目ごとにインスペクションの依頼を出すことになっている。インスペクション機能を用いてインスペクションを行う場合、担当者に設定されたメンバー以外は、インスペクション機能を用いて投稿することはできない。

そのため、アンケート結果でも数人が指摘していたように、理解度の低い人ばかりが評価者（成果物の検査を行う人）になると、エラーが発見されずにインスペクションが終わってしまうことがある。

下表１は、あるグループにおいての担当者ごとの指摘数（投稿されたコメントのうち、内容が指摘として妥当なものの数）と評価者担当になったインスペクションの数を表したものである。

表１　あるグループの指摘数と担当者の関係

担当者	a	b	c	d	e	f	g
指摘数	67	38	17	16	16	4	2
評価担当インスペクション数	10	10	7	8	9	6	9

評価担当しているインスペクション数にはあまり差はないものの、指摘数には大きな差が出ているのが見てとれる。担当者によってインスペクション内容の密度に偏りがでてしまっている。これは特に差が顕著なグループであるが、メンバー全員が同程度の指摘数になっているグループはほとんどなかった。

４．考察

４.１　作業報告書の「問題点」について

未議論・未解決の数は、本当に全てがそうとは限らないとはいえ無視できない数に達していると考える。後々の手戻りやエラーの原因になる可能性があると考えると、個人の問題もグループで把握し解決する必要がある。

作業報告書の問題点が議論されない原因として考えられる点は以下５つある。

(ｉ)作業項目を一覧できるページで、作業項目ごとの問題点を確認することができる。しかし、複数の作業項目にまたがる問題点やあるメンバーの問題点を把握することは難しい。

(ⅱ).様々な種類の問題点が混在しているが、(ｉ)で述べた機能では作業項目ごとに分類されるために把握しづらい。

(ⅲ).(ｉ)で述べた機能では問題点が報告された日時を確認することができず、作業報告書で確認する必要がある。

(ⅳ).その問題が解決したかどうかを記録できないために、解決・未解決の把握ができない。

上記の問題を解決するために、作業報告書で書かれた問題点を一覧できる機能を新たに設けることを考える。

問題は、掲示板やインスペクション機能での議論と内容が重なってしまうものも出てくることが考えられることである。しかし、教員への報告という側面で考えると報告書に書くことは必要になると考えられる。

４.２　インスペクションの担当者について

担当者の理解度の違いによって、指摘数に差が出てしまうことがわかった。対象成果物の内容や、成果物作成者の理解度による成果物の出来の良さの違いによって指摘数が変わってくることも考えられるが、それはやはり評価側でも同じことがいえる。

この場合、理解度の高い人がすべて検査することがよいと考えられるが、作業量が増えることや見落しがあった場合にそのまま終わる可能性があることが問題になる。

５．改善案

５.１　問題点リスト

本機能は、各作業報告書に書かれた問題点を集めて問題点リストとして表示する機能で、４.１で述べた問題を解決し、現状での問題点把握・共有や話し合いの議題挙げに役立てることを目的としている。以下はその機能の内容である。

(1)問題点一覧表示

作業報告書に書かれた問題点、及びその対象作業項目、報告日時を一画面に一覧表示する。この機能により、(ｉ)で問題となっていた、複数の問題点の確認を行うことができる。したい場合のページ遷移や選択の手間が省かれる。また、カテゴリによる階層構造で表示することで、何に対する問題のかを把握しやすくする。下図１は、一覧表示画面のイメージ図である。

図１　一覧表示イメージ簡易版

(2)問題点のカテゴライズ機能

(ⅱ)の問題を解決するための機能。図１に示した一覧表示画面において、各問題点に対し「成果物」「グループの方針」「スケジュール」等カテゴライズできるようにする。作業報告書提出直後の問題点は「未分類」カテゴリに配置されているが、任意にカテゴリ移動可能にする。カテゴリは上記４つを考えているが、グループごとにわかりやすい分類ができるように任意にカテゴリの追加を可能にする。

(3)チェック機能

(ⅳ)の問題を解決するための機能。図１に示した一覧表示画面において、各問題点にチェックボックスを設け、解決した問題や議論の必要がないと思われる程度の問題点に対してチェックをつけることで、解決済み・未解決の確認をすることができるようにする。また、この機能でチェックを入れられた問題点は、(1)による一覧表示時に表示されなくなる。

(4)検索機能

特に(ⅲ)(ⅳ)の問題について解決するための機能。図１に示した一覧表示画面において、検索キーに合致した問題点のみが表示される。確認したい問題点を見つけたい場合に、この機能は必要になると考える。考えている検索キーとしては、以下の４つがある。

・日付：特定の日（例えば、ある話し合いを行った日）に出た問題点を知りたい場合など

・期間：ある期間内（例えば、前回の話し合いの日から今日まで）に出た問題点を知りたい場合など
・作業項目名：ある作業項目について（例えば、「ユースケース」という単語を含む作業項目について）出た問題点を知りたい場合など

５．今後の展望

インスペクションの問題に対しての改善案を考える。その後改善案を考慮したシステムの開発を進め、今年度後期の情報実験に適用する。

また、情報実験での利用状況、アクセスログにより、
・作業報告を行った日（授業日に報告することが多い）後に本機能へのアクセスが増えているか。
・カテゴライズ機能、チェック機能、検索機能の利用による問題点に対する意識が見えるか。
を分析する。それに加えアンケートを実施することにより、問題点リスト機能の有効性評価を行う。

６．参考文献

［１］ S.Matsuura : Practical Software Engineering Education based on Software Development Group Experiments, E-Learn2005, pp.1701-1708, 2005.
［２］松浦佐江子：ソフトウェア開発実験における PBL の評価方法,FIT2006 第５回情報科学技術フォーラム,LN-003,2006
［３］ S. Matsuura and H. Kurihara : Collaboration Support Model of Software Development Experiment, Proc. of the 18th International Conference of Software Engineering and Knowledge Engineering, pp.23-28, 2006.