

ソフトウェア開発実験における PBL の評価方法

An Evaluation Method of Project Based Learning by Software Development Experiment

松浦 佐江子 †
Saeko Matsuura

1. はじめに

近年、大学教育においてもプロジェクトベースの教育の必要性が認識され、様々な授業が実施されつつある[1, 2, 3]。このようなプロジェクトベースの教育は学生が講義で得た知識を活用しながら主体的に問題解決を行う力を養う効果があるが、グループで行動するため、授業としての個人評価が難しくなっている。われわれは 2002 年度よりグループワークによるソフトウェア開発実験を実施している[3]。本授業はオブジェクト指向開発に基づきソフトウェア開発技術とプロジェクト・マネジメント技術を修得することを目的とし、グループで半期をかけて要求分析から実装・テストまでのソフトウェア開発の全工程を体験する PBL (Project Based Learning) である。本稿では、本実験がグループワークにおいて授業目的の達成度による個人の評価を行うためにどのように設計・実施されたかを報告し、PBL における成績評価について議論する。

2. PBL 実践における問題点

PBL による授業では、達成感を得ることができる学生も多い反面、次のような問題が発生している。

- グループ作業のため、自分がやらなくても単位は取得できると考える学生がおり、学生の授業への取り組みには温度差がある。
- 積極的に問題解決を行う学生が不在のグループは結果が出せないことがある。また出せたとしても全員の協力による結果ではなくなっている。
- 教員はグループ単位での成果物やプレゼンテーションを評価することはできるが、問題解決の過程において個々の学生がどのように関与し、貢献したのかを評価する材料をもたないことが多い。
- グループの評価のみが個人の評価になると、一部の学生の努力の結果で良い評価を得た場合、それらの学生の学習意欲が削がれることもある。

これらの問題を解決するためには、個人の達成度を評価する材料を明確にし、学生に周知することで、参加意欲を高める必要がある。しかし、成果物は個人単位ではなくグループで提出されるため、通常の授業と異なり、成果物のみを個人成績評価の対象とはできない。われわれの対象とする PBL はソフトウェア開発であることから、目的に対する達成度を測るためのプロジェクト評価モデルを提案し、本モデルに基づいて評価を行う。次節において、本ソフトウェア開発実験の概要を説明した後、プロジェクト評価モデルを定義する。

3. ソフトウェア開発実験の概要

3.1 前提知識

実験は、学部 3 年生の後期に実施した。受講生は 2002 年度が 121 名、2003 年度が 66 名、2004 年度が 84 名、

2005 年度が 69 名である。受講した学生のすべてが、C 言語の講義・演習を各 2 コマ、Java 言語の授業ならびに演習を各 1 コマ受講している。また、数名を除いた学生がオブジェクト指向開発ならびに UML (Unified Modeling Language) に関する講義 1 コマも受講しており、小規模な例題に関する要求分析・システム分析・システム設計を行った経験がある。ただし、設計書に基づいた実装とテスト計画ならびにテストの実施については未経験である。グループワークに関しては、本学の総合科目「創る」やシステム工学演習において経験している。

3.2 課題

2002 年度ならびに 2003 年度は情報処理学会オブジェクト指向シンポジウムでモデリングの共通課題として提示された「販売管理システム」(課題 A) ならびに「自動販売機の制御システム」(課題 B) を取り上げた。2004 年度は前者の課題に代わり「芝浦工業大学会議室予約システム」を課題とした。半期 (3 ヶ月) に複数人で開発が可能であり、ある程度の複雑度をもつ問題を選定した。課題 A に関してはビジネスロジックを、課題 B に関しては制御ロジックを正確に分析することを目標とした。2005 年度には上記課題に加えて LEGO MINDSTORMS のロボットを使用した「荷物の自動搬送システム」(課題 C) を取り上げ、実機を用いた組込みソフトウェア開発におけるモデリング技術を修得することを目標とした。

3.3 グループ構成

学生が各自課題の中から 1 つを選択する。その後 2002 年度はくじ引きにより、課題 A を 15~17 名、課題 B を 11~12 名の 9 つのグループに分けた。2003 年度からは前期に実施した実験授業の成績によって 9~10 名のグループに分けた。2003 年度は 7 グループ、2004 年度は 9 グループである。2005 年度は課題 A が 6~7 人で 4 グループ、課題 B が 7~8 人で 3 グループ、課題 C が 5 人~6 人で 3 グループとした。各グループにはリーダー 1 名、サブリーダーを 1 から 2 名置くこととした。その他の役割は各グループで決定した。

3.4 実験計画

スケジュールは各フェーズを 3~4 週間として計画した。各授業時間は 2 コマ (計 180 分) で、授業回数は 14 回 (2004 年度より 15 回) である。最終授業時間に発表会 (説明とデモンストレーション) を行った。2003 年度以降は授業終了後に最終検査 (教員が用意した項目についてテストする) を実施した。本実験は教員 2 名、TA 2 名 (2004 年度は 4 名、2005 年度は 6 名) で担当した。

4. PBL の設計および評価モデル

4.1 プロジェクト評価モデル

ソフトウェア開発の最終的な目標は要求を満たすソフトウェアを完成する、すなわち高品質なプロダクトを作成することであるが、この目標を達成するには何時、何をどのように行えばよいかという完成までのプロセスを工夫し、

† 芝浦工業大学システム工学部電子情報システム学科
Shibaura Institute of Technology Department of
Electronic Information Systems

グループメンバー（人）がどのように協力して効率よく作業を行えるのかを考えながら、プロジェクト（＝実験）を進めることが重要である。そこで、ソフトウェア開発実験プロジェクトをプロダクト（成果物）、プロセス（実行の仕方）、グループメンバー（実行する人）のコラボレーションの3つの要素で構成されると捉え、これらの達成目標に対してグループおよび個人の観点から評価を定義する。そして、重み付けを行い総合的にプロジェクトの評価を定義する。表1に示すとおり、プロジェクト評価モデルは定義・達成目標・評価データ・評価基準・評価対象（個人またはグループ）の各要素により定義する。

表1. プロジェクト評価モデルの構成要素

プロダクト	定義	各種ドキュメントおよびシステム
	達成目標	要求を満たすプログラムを完成させる
	評価データ	教員のレビュー・プロダクトの測定・テスト・インタビューによる理解度テスト
プロセス	評価基準	開発方法に基づく評価・テストの可否
	定義	プロダクトを作成するための作業
	達成目標	高品質なプロダクトを作成するためのプロセスを工夫する
コラボレーション	評価データ	作業の報告のレビュー・作業ログのレビュー・最終レポートの評価
	評価基準	理解度・実践における達成度
	定義	プロダクトをプロセスに従い作成する過程（作業モデル）における個人の行動
コラボレーション	ゴール	グループの一員として責任を果たす
	評価データ	作業計画、議論のログのレビュー・議論、作業報告、ファイル投稿の回数・学生の相互評価
	評価基準	ログの測定値における達成度・役割の実践度

本実験ではオブジェクト指向開発(UML, Java)をソフトウェア開発方法として用いる。以下にプロジェクト評価モデルの各要素を説明する。評価対象は個人(Ind.)、グループ(Gr.)と記す。

4.2 プロダクト

4.2.1 定義

ソフトウェア開発の目的は品質の高いソフトウェアを作成することである。ソフトウェア開発方法は品質の高いプロダクトを効率良く作成するための知識である。本実験ではソフトウェアの開発プロセスを要求分析フェーズ、システム分析フェーズ、システム設計フェーズ、実装フェーズ・テストフェーズの5段階に分け、各フェーズの成果として以下の成果物を作成する。

表2. プロダクトの構成要素

フェーズ	内容
要求分析	要求文 ユースケース図* ユースケース記述 ユーザインタフェース・イメージ図 システム・アーキテクチャ計画
システム分析	ユースケース図* ユースケース記述 クラス図* シーケンス図*

	[状態図*] [オブジェクト図*] [アクティビティ図*]
システム設計	クラス図* シーケンス図* パッケージ図* [コンポーネント図*] [配置図*] 統合テスト仕様 単体テスト仕様 テスト計画書
実装	ソースコード一覧 コード システム概要 インストールガイド 操作説明書
テスト	テストクラスコード一覧 テストクラス・コード [パッケージ図*] テストケース一覧 テスト結果一覧

[]はオプション。*はUMLに従う。

4.2.2 達成目標

要求を満たすプログラムを完成させることが最終目標である。表2に示す各中間成果物の完成度および成果物のフェーズ間整合性が取れていることが、中間ゴールとなる。

4.2.3 評価データと評価基準

中間ゴールは表3に示すように、対象データ毎に、レビュー手法[4,5]、ソフトウェアメトリクス[6]による数値測定、ソースコードからクラス図を生成するといったリバースエンジニアリング手法を用いて、各ドキュメントを教員が評価する。また、インタビュー形式により、自グループの仕様書に記載されたUML図の意味を説明させることにより、個人の理解度とその仕様書への貢献を評価する。さらに、各人が最終成果物であるシステムを操作することにより、システムに対する理解度を評価する。

表3. プロダクトの評価データと評価基準

ゴール	データ	評価基準	対象
完成度	UML図	オブジェクト指向のレビュー技術	Gr.
	ソースコードシステム	ライン数・クラス数・クラス構成・複雑度・統合テストケースによる検査	Gr.
整合性	UML図	オブジェクト指向のレビュー技術	Gr.
理解度	UML図	オブジェクト指向のレビュー技術	Ind.
	システム	統合テストケースによる検査	Ind.

4.3 プロセス

4.3.1 定義

各フェーズの目標は前述の成果物を完成することである。それぞれのフェーズで行うべき作業のガイドラインを定義する。ガイドラインでは、開発プロセスの目標を詳細化し、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「タスク」を規定した。タスクはフェーズのはじめに行

う作業（方向付け）、プロダクトを記述する作業（定義）、定義されたプロダクトに含まれる間違いや矛盾を発見する作業（検査）、修正された定義に基づいて成果物を作成する作業（まとめ）の4つのカテゴリに分類されており、フェーズは各カテゴリに属するタスクの列で構成される。「検査」としては各仕様書のインスペクション[5]とソースコードのテストを義務付けた。例えば、要求分析フェーズにおけるタスクには、ユースケース図の作成・ユースケース記述の定義・ユースケース記述のインスペクション等がある。タスクには各成果物を測定するためのメトリクスを定義する。さらに、タスクは学生自身が定める具体的な作業（これをステップと呼ぶ）から構成される。

4.3.2 達成目標

高品質なプロダクトを作成するためのプロセスを工夫することが最終目標である。タスクとして規定された作業の目的を理解し、適切なステップによりタスク列を実践することが求められる。

4.3.3 評価データと評価基準

学生は各フェーズのタスクを用いて作業計画を立案する。タスクのインスタンスを作業項目と呼ぶ。学生は作業項目毎に作業報告を行うことを義務付けられる。作業報告には作業期間、進捗度、成果物のメトリクス、作業経過（ステップと作業時間）、問題点、コメントを記入する。これらにより、表4のように各ゴールの評価を行う。ステップの構成は学生がそのタスクの目的をどのように捉えて実践したかの1つの指標となる。また、授業の最後に、最終レポートとして、技術の理解度・ソフトウェア開発に対する意識・各フェーズの実験内容・グループワーク・オブジェクト指向開発・実験に対する取り組みの自己評価等に関する選択式および記述式の質問（平均103項目）を行い、この中からプロセスの理解度を調査した。

表4. プロセスの評価データと評価基準

ゴール	データ	評価基準	対象
理解度	最終レポート	記述内容の妥当性	Ind.
実践	作業計画	タスク構成・繰り返しの妥当性	Gr.
	ステップ	ステップ構成・開発方法およびプロジェクト・マネージメントの観点からのステップ名称の妥当性	Ind.
	メトリクス	数値の妥当性	Ind.
	問題点	問題点把握の妥当性	Ind.
	インスペクションのログ	インスペクション内容の妥当性	Ind.

4.4 グループメンバーのコラボレーション

4.4.1 定義

グループのコラボレーションを図1に示す作業モデルによって支援する。作業は作業計画によって運営される。作業計画は前述の作業項目で構成され、作業項目毎に該当作業に関する議論およびデータの共有が行われる。学生は作業項目毎に1回以上の作業報告をする義務がある。

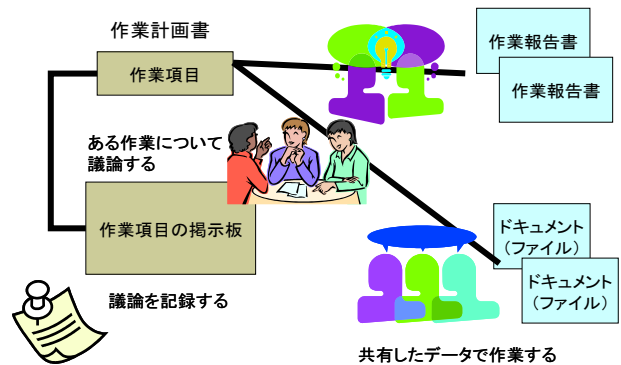


図1. コラボレーションの作業モデル

4.4.2 達成目標

グループの一員として責任を果たすことが最終目標である。前述の作業モデルにおける議論・データの共有コラボレーションへの参加、協調、貢献が求められる。

4.4.3 評価データと評価基準

前述の作業モデルを支援するためにグループワーク支援システムを開発し、実験において運用した[3]。本システムでは作業内容の報告・各種連絡・議論および質問の場を提供することにより、非同期型グループウェアにおけるアウェアネス（協調作業者の作業状況、作業履歴等の認識）[7]を実現している。学生はこの場を通じて、授業時間以外にも情報を共有し、他の学生の作業状況を知ることができる。また、授業時間以外の作業を支援するために、学内外を問わず、自由な時間にアクセスできる環境として本システムを実現した。本システムでは担当者および作業期間を定めた作業項目に基づく定期的な作業報告、会議の議事録、掲示板による議論、仕様書の書庫の4つの場を管理する機能をWebアプリケーションとして提供している。本システムのログ情報により、表5のように各ゴールの評価を行う。最終レポートにおいては、学生にグループ内での貢献度（貢献した2〜3名を指名）を評価させた。

表5. コラボレーションの評価データと評価基準

ゴール	データ	評価基準	対象
参加	掲示板のログ	投稿回数	Ind.
	ファイル共有	投稿回数	Ind.
	作業報告	報告回数	Ind.
協調	掲示板のログ	記述内容の妥当性	Ind.
貢献	最終レポート	得票数	Ind.

4.5 総合評価

プロジェクト評価モデルのすべての項目に対して評価データを用いて評価基準に従い評価を行う。その他に欠席は減点とし、それぞれに重み付けを行って総合点を計算した。

5. 考察

5.1 適用結果

2002年度よりグループワークによるソフトウェア開発実験を実施してきた。年度により以下の差異がある。

- 2002年度には、グループワーク支援システムがなかったため、実践のログは実験終了時の最終レポートにより収集した。統合テストケースによる検査は実施しなかった。最終レポート質問項目数は137である。

- 2003 年度には、グループワーク支援システムにより、実践ログを収集したが、タスクの概念を導入しなかったため、作業報告にはステップ、メトリクスに関する明確な区別はなかった。最終レポート質問項目数は 99 である。
- 2004 年度は、プロダクトの個人評価以外の全ての評価データを収集した。インスペクションは 2004 年度よりタスクとして明記した作業項目である。インスペクション項目と修正方針を示したが、ログを見る限り、十分に理解されていない。そのため、評価の材料としては十分ではない。最終レポート質問項目数は 74 である。
- 2005 年度は、本稿で述べたすべての評価データを収集した。最終レポート質問項目数は 72 である。

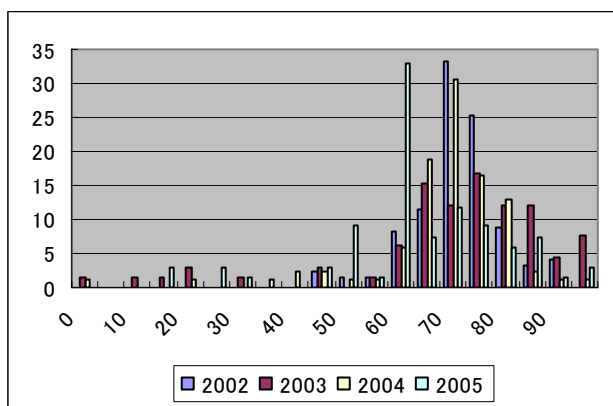


図 2. 成績分布

図 2 は 2002 年度から 2005 年度の成績分布である。横軸が点数を縦軸がパーセンテージを表している。グループワーク支援システムを導入して、議論・作業報告・成果物投稿状況等の作業ログを収集した結果の成績の方が点数のばらつきが生じている。さらにプロダクトの個人評価を導入した 2005 年度では理解度の低い学生が明確になっている。取得平均点は 2002 年度が 73.39 点、2003 年度が 70.64 点、2004 年度が 70.33 点、2005 年度が 62.8 点であった。グループ得点と個人得点の比率は 2002 年度が 4:3、2003 年度が 3:4、2004 年度が 1:2、2005 年度が 1:4 の割合である。アンケート結果からは 4 年間の実験を通じて 88~98% の学生が「実験を行って、自分にとって得るものはあった」と回答しており、分析・設計の重要性・プログラムを他人が理解できることの必要性等ソフトウェア開発への意識の高まりが大いに見受けられる。

5.2 関連研究

Hayes *et al.* [8] の研究ではグループの評価を以下の 5 つの項目の組み合わせで行うことが望ましいと述べている。

- (1) グループに対する評価は個人の評価となる。
- (2) 個人の作業は各人が報告する。
- (3) グループのメンバーが他のメンバーの貢献について報告する。
- (4) プロジェクトの詳細に関するクイズを実施する。
- (5) 個人の作業について相互評価を行う。

本評価モデルでは、インタビュー形式以外のプロダクトの評価はグループごとの評価であり、これがグループメンバーの評価点となっている。(2)の個人の作業状況はプロセスおよびメンバーのコラボレーションにおける個人評価点である。また、学生の貢献度評価が(3)に、最終レポート

の記述評価およびプロダクトの理解度テストが(4)に相当する。学生はグループワーク支援システムを通して他のメンバーのプロセスの評価データである作業状況を見ることができる。本実験では、(5)のような学生による評価は行っていないが、他のメンバーの作業状況の認知により、グループへの貢献が増加する効果があると考えられる。

5.3 今後の課題

個人のグループワークへの貢献度はグループワーク支援システムのログ情報を用いて測定している。しかし、学生によっては、対面で議論する場合には意見を提示しても、システム上の議論には参加しない場合もある。このように教員が直接観察することのできないケースは、現在のところ学生による貢献度評価によって判断している。

プロダクトの理解度に関する個人評価は、個別の理解度を把握するためには有効な手段であった。教員および TA が分担して、あらかじめ用意した質問と評価基準に従って、インタビュー形式で評価を行ったが、1 回分の授業時間が必要であり、毎行っている各グループとのディスカッションができないという問題も生じた。

6. まとめ

PBL においてはグループ作業における個人の評価が難しいが、ソフトウェア開発実験においては、開発方法論および開発作業モデルにより、グループならびに個人の評価項目を明確にした評価モデルを定義することができる。ただし、これらの評価を行うためには、グループワークを適切に支援するツールおよびプロダクトを評価するメトリクス測定ツールやリバーズエンジニアリングツールを有効活用することが必要である。

参考文献

- [1] M. I. Alfonso and F. Mora, Learning Software Engineering with Group Work, Proc. of the 16th Conference on Software Engineering Education and Training, 2003.
- [2] Davenport, D.: Experience Using a Project-Based Approach in an Introductory Programming Course, IEEE Trans. Education, Vol. 43, No. 4, pp. 443-448, 2000.
- [3] S. Matsuura: Practical Software Engineering Education based on Software Development Group Experiments, E-Learn 2005, pp. 1701-1708, 2005.
- [4] Braude, E. J.: Software Engineering: An Object-Oriented Perspective, Wiley, 2000.
- [5] G. H. Travassos, F. Shull, M. Fredericks and V. R. Basili: Detecting defects in Object-Oriented Designs: Using Reading Techniques to Increase Software Quality, Proc of OOPSLA, pp. 47-56, 1999.
- [6] Chidamber, S. R. and Kemerer, C. F.: A Metrics Suite for Object Oriented Design, IEEE Transactions of Software Engineering, Vol. 20, No. 6, pp. 476-493, 1994.
- [7] Dourish, P. and Bellotti, V.: Awareness and Coordination in Shared Workspaces, Proc. of the ACM CSCW'92, ACM, pp. 107-114, 1992.
- [8] J. H. Hayes, T. C. Lethbridge, and D. Port: Evaluating Individual Contribution Toward Group Software Engineering Projects, Proc. of the 25th ICSE, pp. 622-627, 2003.