

ソフトウェア開発の教員主導型 PBL における 反復プロセスと EVM 導入の効果

松澤 芳昭^{†1} 塩見 彰 睦^{†1}
稜川 友宏^{†1} 酒井 三四郎^{†1}

我々は、先導的 IT スペシャリスト育成推進プログラムの一環として、大学院生にソフトウェア工学のプラクティスを体験・評価させる PBL(Project-Based Learning) コースの設計、およびその実施を行った。このコースは、LEGO Mindstorms のロボット制作を題材とした教員主導型の PBL である。最終的に学生 3 名から構成されるチームが約 1000 行のソフトウェアを 6 週間 (延べ約 100 時間) かけて開発する。コースの設計にあたっては、数週間単位の小さな反復プロセスによってソフトウェアの設計-実装のトレーサビリティを明確にするようにし、最後の反復では EVM(Earned Value Management) の手法を用いて学生にマネジメントを行わせるようにした。その結果、学生プロジェクトの工程間や学生間の工数や生産性などの詳細な基礎データが得られ、学生のレポートからは設計と実装のトレーサビリティの問題、品質とコストのバランスについてデータに基づいて考察していることが読み取れた。

Effects of Promoting EVM and Iterative Process in Scenario Based PBL for Software Engineers

YOSHIAKI MATSUZAWA,^{†1} AKICHIKA SHIOMI,^{†1}
TOMOHIRO HARAICAWA^{†1} and SANSHIRO SAKAI^{†1}

We have developed a PBL(Project-Based Learning) course for master course students to give them experiences and opportunities to evaluate the common software engineering processes, in the Leading IT Specialist Education Program of MEXT. The theme of the PBL is to develop a pair of Robots by using Lego Mindstorms. The project team composed of three students try to develop the system for six weeks (about a hundred hours per team), and the developed software had about thousand lines. Although the major PBL scenario is given by teachers, designated iterative curriculum requires learners to manage the design-implementation traceability and cost performance by using EVM(Earned Value Management) methodology. As a result, stu-

dents succeeded to collect fundamental data of the project and report their design-implementation traceability and cost performance of the project based on numerical data that is collected by themselves.

1. はじめに

近年、情報技術者の育成方法として PBL(Project-Based Learning) の必要性が叫ばれ、教育現場に取り入れられるようになってきた。ソフトウェア開発には、自然科学だけでなく社会科学をも含む、柔軟で高度な問題解決とマネジメントの能力が必要とされる。しかしながら、そうした能力を系統的に育成する教育方法論は確立されていない。情報技術者育成現場への PBL の導入は、こうした方法論確立へ向けたアプローチと考えることができる。

一方で、PBL の方法論、概念が未だ確立されていない現状がある。世界で様々なモデルが提案されており¹⁾、日本でも様々なモデルが提案されてきている。その中には、ソフトウェア工学の実践を試みるもの²⁾、産学連携により組み込み教材を開発するもの³⁾、企業人プロジェクトマネージャを導入するもの⁴⁾、情報システムの教育を重視するもの⁵⁾、文部科学省の先導的 IT スペシャリスト関連の教育研究⁶⁾⁷⁾⁸⁾ などがある。

今後、上記研究の発展を通して PBL の概念形成が行われていくものと思われるが、その際には何らかの方法で PBL の評価をしていく必要がある。この問題への対応として、学習者自身によるスキルチェックが行われ始めている⁷⁾⁸⁾。しかしながら、学習者ごとに異なる評価基準でのスキルチェックがどこまで有効な数字であるかは、疑問が残る。

筆者の PBL に対するもう一つの危惧は、「実践的」という謳い文句を盾にして、単なる文書作成体験が行われる可能性である。筆者の経験から、「設計をしてみただけで、実装へのトレーサビリティが全くない (設計が実装に反映されない)」という現象はしばしば見られる。しかしながら、この問題に言及している研究は少数である。

本研究では、こうした問題点を踏まえて、まず開発プロジェクトのデータを学生に収集・分析させることをテーマとし、EVM(Earned Value Management) を導入することを考えた。ソフトウェア開発の正否と学習効果は必ずしも一致しないことは理解しつつも、学習効果を論じる際に開発のパフォーマンスは論拠として客観的なデータとなると考えるためであ

^{†1} 静岡大学情報学部
Faculty of Informatics, Shizuoka University

る。次に設計-実装のトレーサビリティについて、反復的に設計-実装の経験と検討をさせることによって、問題の解決を試みた。

本研究では、上記の試みについて、大学院生向けの講師主導型(2.1節で説明)のPBLに取り入れ、実際の教育を行った。本稿では以下、2章でこの教育の背景とカリキュラム設計について述べ、3章で実際の受講生が収集・分析した結果を報告する。この結果を踏まえて4章でこの試みの検討を行う。5章でまとめを述べる。

2. 教育の背景とコースの設計

本稿で述べるPBLは、文部科学省先導的ITスペシャリスト推進プログラムの一部として実施された。本章では、教育の背景とPBLの位置づけについて説明し、続いてPBLコースの設計方針について述べる。

2.1 PBLの背景と位置づけ

本稿で提案するPBLは、文部科学省先導的ITスペシャリスト推進プログラムの中で名古屋大学を中心とし、南山大学、愛知県立大学、静岡大学、ほか6社が連携する「OJLによる最先端技術適応能力を持つIT人材育成拠点の形成」(OCEAN:On the job Centered Education for Advanced engNeers)プロジェクトにおけるカリキュラムの一部である。

このOCEANプロジェクトのうち、静岡大学以外の3大学は名古屋地区に位置し、教育コンセプトの構築整備を統括している。静岡大学の役割はリモート拠点として、名古屋地区で整備されたカリキュラムの移転試行を担当している。名古屋地区の3大学の学生は合同で講義科目を受講する。静岡大学の学生は、インターネット回線を利用した遠隔授業で名古屋拠点と同様の授業を受講する。PBL科目については、遠隔で実施するのは困難であるため、名古屋拠点で開発されたコンセプトを踏襲しつつ、独自PBLの構築を行っている。

本稿はこの静岡拠点独自のPBLについて述べるものであるが、先にOCEAN全体のPBLの位置づけについて説明しておく。OCEANプロジェクトのコンセプトの概要と名古屋地区が開発したPBL教材については文献⁷⁾にまとめられている。OCEANプロジェクトにおけるPBLは、修士2年間のカリキュラム当初で受講するソフトウェア工学の講義と、終盤で受講する実際の企業と共同で実用レベルのソフトウェアを開発する実践研究(OJL: On the Job Learning)をつなぐ授業として位置づけられている。このPBLで目的は、講義で得た知識を実際に適用することで理解を深めることと、チーム開発やソフトウェア開発マネジメントを体験し、続くOJLが効果的なものとなるように準備と心構えを行うことである。

従って、本PBLの目的は、一通りのソフトウェア工学の講義を履修済みである学習者に

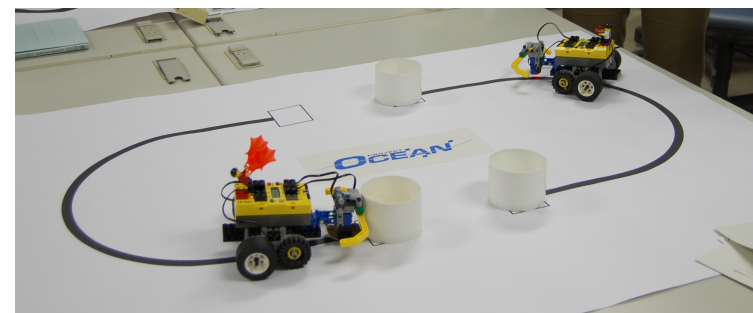


図1 課題となる自動搬送システム

Fig. 1 The auto transportation system: a project goal.

対して、演習によって理解を深めさせ、スキルとしての技術を身につけさせることにある。このPBLの過程では、大まかな課題とシナリオが与えられることから、井上らによるPBLの分類⁵⁾における「チュートリアル型:シナリオやシミュレーションで示される問題を理解し、既知の知識・経験を基に新たな知識を主体的に学習する方法」に分類されると考えられる。本稿のタイトルを「講師主導型」としているのはこのためである。

2.2 教材のベース

本稿で提案するPBLは、株式会社アフレルが開発した、組込みソフトウェア一連の開発を学習する教材⁹⁾を基礎として、我々の目的にかなうよう改変して展開した。

上記の教材においては、Lego Mindstormsによって2台のライトレース型ロボットを構築し、それぞれ自動搬送システムを実現する。設計段階ではUML(Unified Modeling Language)によるオブジェクト指向分析・設計を行う。その後実際にC言語によってシステムを構築してロボットの動作をテスト・検証し、分析・設計の良し悪しを議論する。実際に構築されたシステムの様子を図1に示す。

Mindstormsを使った教材自体は、新しいものではなく、多くの先行研究でも利用されている²⁾³⁾。近年では、「ETロボコン」と銘打つコンテストが開催され、この催しを利用した教育効果についても報告されている¹⁰⁾。上記の教材は、「ETロボコン」と同様、Legoを使って組立てた車体を利用するが、制御に関しては知識を必要とせず、モデリングと状態遷移といったソフトウェアの基本設計が問われる課題となっている。完成すべきソフトウェアの規模も500ステップ前後と、組込み開発の入門とソフトウェア設計の教育を行う目的に対して適正規模の課題と考えられる。

2.3 カリキュラムの設計

提案する PBL カリキュラムは、1 章で述べた問題意識に基づいて、大学院の 1 セメスター 2 単位のコースとして 2.2 節で紹介したベース教材を改変し構築した。その際、受講生に対して以下の条件を想定した。

- C 言語は流暢に扱える。
- 組込み開発の経験はない。
- ソフトウェア工学の基礎、UML、プロジェクトマネジメントについては講義で知識のみ学習している。

カリキュラム設計（再構築）の大まかな方針は以下の 2 点である。

- (1) ベース教材では単一のウォーターフォール型を体験するのみにとどまっているため、反復プロセスを導入し、設計を体験するだけでなく、学生自らが試行錯誤を行って設計法やアーキテクチャ、および要求仕様書を作ることの意義について吟味・討論できるようにする
- (2) 構成管理ツール、メトリクス計測ツールの導入、プロジェクトマネジメントの概念を取り入れて、時間の見積もりや構成管理計画を自ら行って進めるようにする。

設計された 15 回の演習授業の構成を表 1 に示す。このカリキュラムは、大きく 3 つのフェーズに分かれている。フェーズ I では、C 言語を用いた組込みプログラミングの基礎プロセスの体験を行う。ここで学習者はハードウェアの特性を把握し、並行プログラミング、状態遷移プログラミングと単体テストを行う。

フェーズ II は、UML による設計とチーム開発がテーマである。課題である自動搬送システムのプロトタイプについてモジュール分割を行って実装し、結合する。そのため、設計に不備があると、結合で不具合が発生する。こうした過程で設計の良さ、および設計図の引き方について理解を深める。さらに、ここで、CVS(Concurrent Versions System) や CASE ツール（本 PBL では Jude を利用した）の導入も行う。

フェーズ III は、総合演習とプロジェクトマネジメントとがテーマであり、このフェーズにおいては、最終的な自動搬送システムの完成品をチームが発注者、受注者役となって発注しあう。ユースケース、要求仕様書、およびアーキテクチャについて導入した後、プロジェクトマネジメントと EVM の技法を説明して、プロジェクトを学生にゆだねる。プロジェクトはプロジェクトマネージャ、設計リーダー、品質リーダーに分かれてプロジェクトを進め、最終回で結果を報告する。

提案する PBL における各フェーズと V 字プロセスの対応を図 2 に示す。こうした V 字

プロセス+3 フェーズの反復方式は、OCEAN における PBL 設計のコンセプト⁷⁾を取り入れている。ただし、OCEAN の設計においては、ソフトウェアの確認と検証工程が重視されて、全てのテスト工程から始まり、大きな反復を二度行う”N V モデル”となっているのに対して、我々の開発したモデルでは、V 字プロセスにおける下流工程から始まり、フェーズが進捗するごとに大きな V のプロセスを実践していくモデルになっている。

マネジメントの要素をフェーズ III から導入するのは、NV モデルとほぼ同様である。これは、マネジメントと内容を同時に新規に学習しなければならない状況避けるためである。フェーズ III は内容に関しては復習の意味合いが濃くなり、学習者はマネジメントの学習に集中できる。こうして、フェーズ I, II は、マネジメントの学習の足場組みの役割を果たす。

2.4 指導方法とツールの利用

本節では、本稿の課題である EVM の導入と、設計-実装のトレーサビリティ検討について、それぞれの指導方法と利用したツールについて述べる。

2.4.1 EVM の導入について

EVM の教育現場での利用に関しては、ソフトウェア開発に係るコストは概ね人件費で

表 1 提案する PBL のカリキュラム
Table 1 Curriculum of the proposed PBL.

フェーズ I: 組込みプログラミング・及びテストの基礎	
第 1 回 (180 分)	: 概要説明と環境設定
第 2 回 (180 分)	: BrickOS による組込みプログラミングと単体テスト
第 3 回 (180 分)	: ハードウェアの特性検証実験
第 4 回 (180 分)	: 状態遷移モデルの設計と実装, テスト
第 5 回 (180 分)	: 状態遷移モデルの設計と実装, テスト
フェーズ II: 設計プロセスの体験と評価	
第 6 回 (180 分)	: UML を利用したモジュール分割の設計
第 7 回 (180 分)	: UML を利用したモジュール分割と実装
第 8 回 (180 分)	: モジュール結合テスト
フェーズ III 総合演習とプロジェクトマネジメント	
第 9 回 (180 分)	: ユースケースと要求仕様書作成
第 10 回 (180 分)	: 抽象レイヤーアーキテクチャによる設計
第 11 回 (180 分)	: プロジェクトマネジメントの基礎と EVM による進捗報告の方法
第 12 回 (180 分)	: 計画に沿ったプロジェクト遂行 (進捗報告)
第 13 回 (180 分)	: 計画に沿ったプロジェクト遂行 (進捗報告)
第 14 回 (180 分)	: 計画に沿ったプロジェクト遂行 (進捗報告)
第 15 回 (180 分)	: 最終プレゼンテーションとまとめ

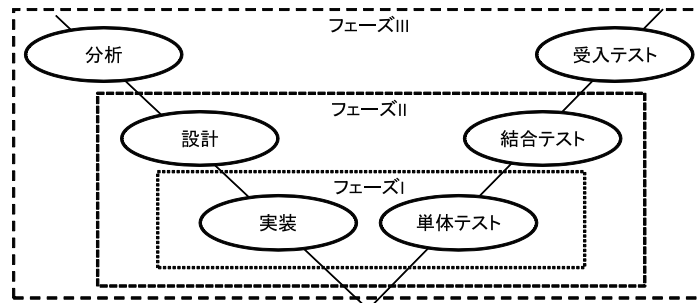


図 2 各フェーズとV字モデルとの対応
Fig.2 Instructional phases mapped into the V-model.

あることを考慮して、時間=コストとする。この考え方は、先行研究¹¹⁾と同様である。本PBLの活動時間は、概ね授業時間の3時間とその後の引き続きの作業時間程度で、授業日以外の活動は予定されないため、進捗報告は1週間毎とし、EVMの基礎データの単位を「分」とする。

EVMの導入の際には、以下の基本概念を次のように説明する。

- 何ができていた予定であるか (PV : Planned Value)
- どのくらいの時間 (コスト) をかけてやったか (AC : Actual Cost)
- 実績はどのようであったか (EV : Earned Value)

このとき、EVの算出方法については一通りの方法を説明した後、一番導入が容易と思われる「出来高パーセント見積り法¹²⁾」を活用するよう指導する。同時に、正確なEVの算出には正確なWBS(Work Breakdown Structure)が必要であることを講義して、4段階のレベルまで詳細化するよう指導する。

EVMの記録と分析のためのツールには、表計算ソフトを利用する。これは、市販のプロジェクトマネジメントツールは、機能が多すぎて理解するのが容易ではないことと1週間ごとに「分」で管理できるという要求を満たすソフトウェアがないという理由による。

表計算ソフトを利用するにあたっては、初学者が容易に利用できる、簡易的なテンプレートを用意した。その画面イメージを図3に示す。このテンプレートを利用し、計画段階では画面左側にWBSを入力し、右側にPVを週ごとに入力させる。進捗報告時にその週のAC, EVを入力すると、EVM推移チャートが作成される。3章で示すチャートは、このツールを利用したものである。

	A	B	C	D	E	T	U	V	W	X	Y
1	単位/分	レベル1	レベル2	レベル3	レベル4	一名ずつ	～2/3	～11/25	～12/2	～12/9	
2	レベル1	レベル2	レベル3	レベル4	担当	PV	AC	EV	AC	EV	AC
10	配達単システム										
11	要求分析										
12	要求分析										
13	要求分析				学生B		120	120			
14	要求仕様書作成				学生C		120	120			
15	要求仕様書作成										
16	作成				学生A		120			120	
17	作成				学生B				150	150	
18	作成				学生C			120	390	270	
19	設計										
20	アーキテクチャ設計										
21	設計議論				学生A			60	10		
22	設計議論				学生B			240	90		
23	設計議論				学生C			240	90		
24	設計図添削										
25	クラス図				学生C			40	30		
26	全体の状態遷移図				学生B			30	30		
27	実行状態の状態遷移図				学生C			30	30		
28	実行状態の状態遷移図				学生C			30	30		
29	キャリアレシジョンの状態遷移図				学生C			30	30		
30	引継ぎのシークンス図				学生C						
31	キャリアレシジョンのシークンス図				学生A						
32	実行状態のシークンス図				学生B						

図 3 EVMを計測・分析するスプレッドシート
Fig.3 Spreadsheet for EVM.

ACの測定は基本的に自己申告であるが、なるべく正確に記録するように指導する。このとき、個人ごとの生産性分析を行うが、決してその生産性や能力差は成績評価に影響を与えないことを約束している。

2.4.2 設計-実装のトレーサビリティ検討について

本PBLでは、設計が実装に反映されないということがないように、設計-実装のトレーサビリティの検討について、各フェーズで指導する。フェーズI, IIにおいては、記法の正しさをレビューして、内容に関しては、実装が終わってから「設計がどのように役に立ったか」という視点で考察させる。

フェーズIIIにおいては、設計や実装の変更履歴を記録し、分析するように指導する。このときに、設計の内容に関する評価は成績に考慮せず、設計変更履歴の正確さと分析の精密さを評価の対象となることを学習者に伝える。ソースコードの管理については、フェーズIIから導入するCVSの利用を技術者の常識として定着をはかる。基本的なデータとして、コードメトリクスによる分析を行うよう指導する。本PBLにおいては、C言語用のツールで履歴の追跡も可能なSource Monitor¹³⁾を利用した。設計段階のCASEツールはJUDEを利用する。後で履歴が追跡できるように過程をできるだけ保存管理するよう指導する。

3. 試行結果

2章で述べた環境を構築し、教育を実施した。本章では、フェーズIIIの結果を主として、

学習者自らが収集したデータ*1と分析結果を示し、教育効果を検討する。

3.1 実験環境とプロジェクトの概要

教育は2008年度後期におこなわれた。対象は静岡大学大学院情報学研究科のITスペシャリストコース修士1年生6名であり、学習者は総じて、2.3節での想定を充足する学習者であった。フェーズI, IIにおいて、各回において60-180分程度の追加演習時間が必要であったが、日程的にはほぼ予定通りに進行した。

フェーズIIIにおいて、初期フェーズを修了した6名を2チームに分けてチーム編成し、プロジェクトが実施された。プロジェクトの基礎データをまとめて表2に示す。2チームはそれぞれ学生3名から構成され、約1000行のソフトウェアが6週間(延べ約100時間)かけて開発された。要求仕様書、クラス数、テスト項目についてもほぼ同様の規模となり、費用対効果の結果についても約50%のコスト超過となっており、これも同様であった。

その他、全体の雰囲気として、学生たちに、楽しみながら真摯に取り組む姿勢が見受けられた。受講後の無記名アンケートにおいても、概ね満足度は良好であった。

3.2 フェーズIII: プロジェクトマネジメントの観点から

本節では、プロジェクトマネジメントの観点からプロジェクトの様相と学習成果について詳しく検討する。まず、EVMのデータを示し、その後学生によるデータ分析例を示す。

3.2.1 EVM データ

まず、プロジェクトの時系列の様相を俯瞰するために、編成された2チーム(A, Bとする)のEVM推移グラフを図4, 図5に示す。両チームともに、EVがPVにほぼ一致する形となっている。これは、今回の受講者がスケジュールのマネジメントを行って、計画通り

表2 プロジェクトの基礎データ
Table 2 Fundamental data of two projects.

		チーム A	チーム B
学生数		3	3
規模 (行)		1334	959
要求仕様書 (p)		16	14
クラス数		12	18
統合テスト項目数		33	34
EVM (分)	PV	4235	4000
	AC	5980	6035
	EV	4235	4000

*1 本稿における時間データの単位は全て「分」に統一している。

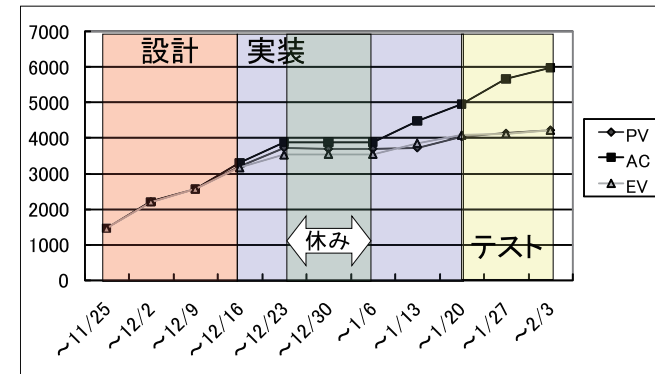


図4 チーム A の EVM 推移チャート
Fig. 4 EVM line chart for Team A.

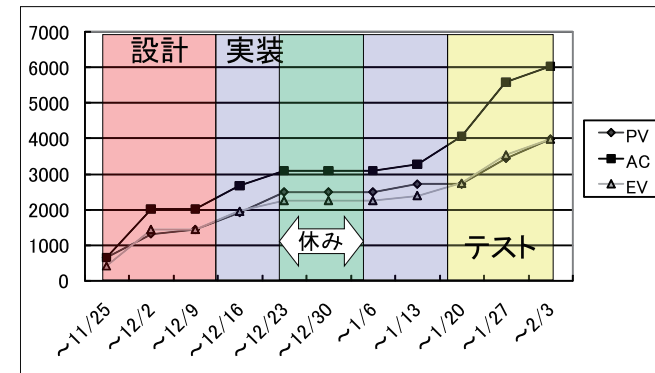


図5 チーム B の EVM 推移チャート
Fig. 5 EVM line chart for Team B.

に進め、期待される品質の成果が得られたことを示している。ただし、コスト見積もりについては甘く、実装・テストの各フェーズにおいて、毎週予定を超過している。

次に、工程別の予実比較の結果を図6, 図7に示す。それぞれのプロジェクトにおける工程の分類方法が異なるので、単純な比較はできないが、テストの見積もりが軽視され、実績が大幅に見積もりを超過する現象については、2チームとも同様となった。要求分析・設計

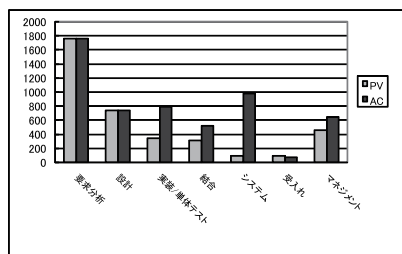


図 6 チーム A の予実比較 (工程別)

Fig. 6 Plan/actual comparisons for TeamA (by processes).

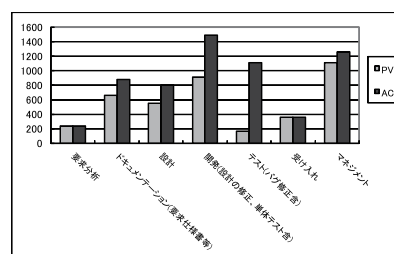


図 7 チーム B の予実比較 (工程別)

Fig. 7 Plan/actual comparisons for TeamB (by processes).

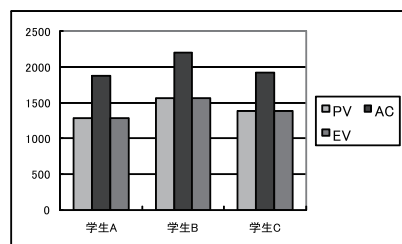


図 8 チーム A の予実比較 (開発者別)

Fig. 8 Plan/actual comparisons for TeamA (by developers).

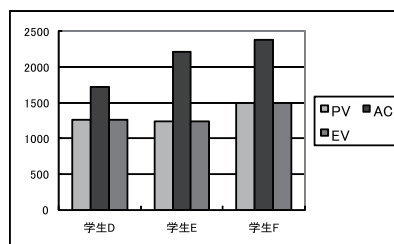


図 9 チーム B の予実比較 (開発者別)

Fig. 9 Plan/actual comparisons for TeamB (by developers).

等, 上流工程の予実には差が見られないことに関しては, 当該工程については, 教員主導で実施しているためである。

最後に, 開発者別の予実比較の結果を図 8, 図 9 に示す. このデータにおいては, 全ての学生がほぼ均等にコスト超過という結果となった. 特定のメンバに仕事が集中することではなく, チームワークよく仕事が進められたことを示している. この結果について, 学生自身も, 「能力別に PV を設定し, ほぼ均等な残業を行った」と説明している。

3.2.2 学習者によるデータの分析

本節で示してきたマネジメントデータに対する, 学習者による分析例を下記に列挙する。

- コスト超過の原因としては WBS 作成時にテストの実施にコストをあまり割いていなかったこと (プロジェクト全体 PV におけるテスト PV の割合は 11.7%) が一番の原因であり, コストを気にするあまり実装もテストも同一人物で行ってしまった. その

ため, 単体テストを十分に実施できておらず, システムテストでは 10 倍以上のコストをかけても修正できないという結果となった. 結合テストに注目すると, 結合テストで単体テストをカバーする PV としては少なく, 後々エラーが発生した場合に修正に困難になる要因を生んだと考えられる. 以上のことから, コスト超過と品質の問題は, 講義で学んだテストの重要性を軽視した結果が招いたと結論づけることができる。

- 今回, 要求仕様を満たすことに加え, 品質のことも考えて設計を行った. そのため, 柔軟性は高まっているが, 実装が難しくなり, コストも高くなってしまったという問題があった. テストのしやすさ, コスト, スケジュールについては全く考えずに設計を行ったので, 単体テストが面倒なものとなり, コストも超過してしまった.

これらの記述から, 学習者達はプロジェクトの結果について, 定量的な観点から分析を行い, コストという制約をも考慮した設計の善し悪しについての分析も行っていることが読み取れる。

3.3 フェーズ III: ソフトウェア設計の観点から

本節では, ソフトウェア設計の観点からプロジェクトの様相と学習成果について, ソフトウェアメトリクス, および設計変更の履歴というデータを用いて, 学生がそのデータをどのように分析しているかを示す。

3.3.1 コードメトリクス分析

2 つプロジェクトのコードメトリクスの計測結果を表 3 に示し, そのデータに対する, 学生による分析結果の例を以下に列挙する。

- キャリブレーション機能が一番複雑であり, 関数の機能の割には複雑である. これは状

表 3 コードメトリクス計測結果
Table 3 Measured code metrics.

	チーム A	チーム B
行数	1334	959
ステップ数	635	399
コメント率	30.7 %	32.4 %
ファイル (モジュール) 数	25	30
関数数	57	50
関数/ステップ	12.9	7.4
深さ	平均	1
	最大	5
サイクロマティック 複雑度	平均	3.92
	最大	9

態の数が多いことが原因であった。

- クラスごとのメソッド数として最大値であるのはクラスAとBであり、それぞれ8個である。一つのクラスにあるメソッド数としては決して多くなく、これらのクラスへ機能を詰めすぎたことはないと考えられる。
- 開発初期に比べ、コメント率が減少している。保守性の観点からもう少し意識してコメントを書く必要がある。

実際のレポートでは、おのおののデータに対して、こうした分析が加えられている。この分析に際しては、数値の意味を理解するために、講義で学習したメトリクス値の意味を調べ直して分析している形跡が見受けられた。

3.3.2 設計の分析

2つプロジェクトの設計変更の記録の結果を表4に示し、そのデータに対する、学生による分析結果の例を以下に列挙する。

- 各クラスへの関数追加：実装担当部分である mail_watcher.c への機能追加が多かった。この原因としては、引き継ぎ部分の仕様について、収集車側との再検討の結果、変更があったためである。
- 先に状態遷移図を作成してみると、文書だけでは捉え切れていなかったシステム全体の挙動をつかむことができ、効果を実感した。
- 実装後は、依存関係の複雑化、ハンドラを表すインターフェースクラスの分割などの変更を反映させるため、クラス図を大きく書き直す必要があった。この理由としては、初期化等、一部の処理をクラス図に記述していなかったことや、レイヤーを飛び越えた依存関係を許していたことなど、設計に穴があったことが上げられる。このような設計工程の甘さから、実装時には設計工程に戻り、議論を行う必要が生じ、コスト超過の原因の一因となってしまった。
- シークエンス図は、クラス図や仕様の変更の影響を除けば、ほとんど修正する必要もな

表4 設計変更の記録
Table 4 Record of the design changes.

	チーム A	チーム B
モジュールの追加・削除	0	2
モジュール依存関係の変更	0	約 10
関数追加	2	4
関数入出力仕様変更	9	0
状態統合・分割	0	3

く、実装時には参考にすることでコーディングを効率よく行うことができた。

本PBLでは、設計-実装のトレーサビリティを確保、吟味することを強調して指導方針とした。その結果として、上記のような量・質双方の観点からの分析が加えられていることで、その教育効果が認められる。

4. 考 察

本章では、3章で述べた実践の結果を踏まえて、本研究のテーマであった「EVM 導入」と「反復型プロセス」の効果について考察を行う。

4.1 EVM を導入することの効果

EVMによるプロジェクトマネジメントの結果は3.2節にて述べたように、2チームとも品質は計画を満足したが、コスト約50%超過という結果となった。筆者による実際の学習者の様子の観察においても、

- (1) 締切前の徹夜作業ではなく、スケジュールを守ってプロジェクトを進めていた
- (2) 毎週ある程度の居残りをして作業をしていた

ことが観察されたことから、学生が収集したデータは一定の信頼性があるものと考えられる。

このデータに基づくプロジェクトの評価として、学生達はプロジェクトを「失敗」と結論している。しかしながら、この結果はEVMによる工数管理を行ったからこそ浮き彫りになった問題であり、納期と品質は守られている。メンバー間による工数差異も少数である。この結果は、学生のみで行う短期間のプロジェクトとしては、良い成果と言ってよい。

学生の報告書からは、マネジメント・設計の分析とともに単なる感想ではなく、データに基づく客観的な評価が行われていることが読み取れる。コストを考慮した設計に関する分析も読み取れ、全体的にコスト意識が高まっていることが確認できる。こうした結果から、本稿で提案したEVM導入の教育効果に一定の評価が与えられる。

また、教材として用意したEVMの概念講習とツールは学生に速やかに受け入れられ、AC, PV等の専門用語は頻繁にコミュニケーションに使われるようになった。その後行われるOJLでも活用されていることから、本PBLによりこの概念は適用できるレベルに定着したと言って良く、定性的ではあるが、ツールの適正評価もできたと考えている。

4.2 反復型プロセスを導入することの効果

2.3節で述べたように、反復プロセスを導入した動機は、「設計を体験するだけでなく、学生自らが試行錯誤を行って設計等の意義について吟味・討論できるようにする」ことであった。この結果として、フェーズIIIにおいて3.3節で示したように、設計変更の理由や設計

の意義について、学習者からデータに基づいた報告がなされていることから、反復型プロセス導入に一定の効果があつたと考えることができる。学習者からのコメントとして「フェーズ I で設計、実装、テストの小さな流れから体験できたのはよかった」、という主観的なものがあつたことも付け加えておく。

しかしながら、反復プロセス導入の効果であることの直接的な証拠を得るための調査を行わなかったため、本質的に何が重要な要素であつたかの解明に関しては今後の課題となる。

また、2.3 節において、OCEAN カリキュラムにおける同じ反復型の NV モデル⁷⁾ と本 PBL で用いたモデルの相違について言及した。このモデル差異について、NV モデル担当者と筆者が相互に最終発表を見学し、意見交換を行った。その結果、「NV モデルでは検証プロセスが重視されており、発表においても検証技術に関する深い考察がなされている」のに対して、本 PBL に関しては、「マネジメントおよび設計の発表については優れているが、検証に関する意識は甘い」と第三者からの指摘が得られた。検証についての議論が甘いことに関しては、学生の EVM データおよびその分析からも裏付けられる。これはモデルによる特性とも考えられるが、本 PBL においても、検証技術のカリキュラム改善を今後の課題としたい。

5. ま と め

本稿では、我々が開発した OCEAN カリキュラムにおける静岡大学版 PBL について、その設計思想と実践結果を述べた。

この PBL は、大学院生を対象とし、講義で学習したソフトウェア工学の知識について、実践によって理解を深めることを目的とした教員主導型の PBL である。この PBL 設計においては、学生に EVM を用いたマネジメントの基礎データ収集とその分析を行わせること、およびソフトウェア開発における設計-実装のトレーサビリティが確保され、各工程の意味の吟味がなされるようにすることを目標とした。この目標を実現するため、Lego Mindstorms の教材を基礎として、下流工程から進化する 3 フェーズの反復プロセスからなるカリキュラムを用意し、最終フェーズで EVM を学生に実践させるため、教材とツールを PBL 向けに整理して用意した。

実践の結果からは、学生が EVM というツールを完全に使いこなし、精度の高いデータ収集が行われ、学生のコスト意識、設計に関する考察も単なる感想ではなく、数値データに基づいて客観的に行われていることが分かった。

この結果から、教員主導型の PBL においても、学生主導でプロジェクトの基礎データを

収集・分析させることが可能であることが確認された。本稿が示したデータは一般的なものとなっており、他の PBL との結果の比較が可能となることも期待される。

謝辞 本稿で示した PBL の研究開発は、文部科学省研究拠点整備等補助金（先導的 IT スペシャリスト育成推進プログラム）による助成のもとで行われた。カリキュラム研究開発には森孝夫氏、および株式会社アフレルに協力いただいた。関係各位に謝意を表す。

参 考 文 献

- 1) Erik de Graff and Anette Kolmos: *Management of Change -Implementation of Problem-Based and Project-Based Learning in Engineering*, Sense Publishers (2007).
- 2) 松浦佐江子：実践的ソフトウェア開発実習によるソフトウェア工学教育，情報処理学会論文誌， Vol.48, No.8, pp.2578-2595 (2007).
- 3) 花野井歳弘，有田五次郎，澤田 直，牛島和夫，吉元健次，牧菌幸司：双方向型産学連携実践教育，情報処理学会論文誌， Vol.48, No.2, pp.832-845 (2007).
- 4) 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育成の試みと成果，情報処理学会論文誌， Vol.48, No.8, pp.2767-2780 (2007).
- 5) 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， Vol.49, No.2, pp.930-943 (2008).
- 6) 坂本憲昭，深瀬光聡，峯 恒憲，日下部茂，中西恒夫，大森洋一，北須賀輝明，ウッディンモハメッドメスバ，荒木啓二郎，福田 晃，安浦寛人：大規模な産学連携による高度 ICT 人材に向けての取り組み，情報処理学会論文誌， Vol.49, No.8, pp.2830-2842 (2008).
- 7) 沢田篤史，小林隆志，金子伸幸，中道 上，大久保弘崇，山本晋一郎：飛行船制御を題材としたプロジェクト型ソフトウェア開発実習，組み込みシステムシンポジウム 2008, pp.5-14 (2008).
- 8) 駒谷昇一：実践的 PBL によるエンタープライズ系システム企画設計開発の授業実践，情報処理学会研究報告 (2009-IS-107)， pp.177-184 (2009).
- 9) 株式会社アフレル：組込みシステム開発実践コース UML-C 編 http://www.afrel.co.jp/engineer/exp_embed.html (2009/04/19 に参照).
- 10) 小倉信彦，渡辺晴美：ロボットコンテストを利用した組込み教育の実践，情報処理学会論文誌， Vol.49, No.10, pp.3531-3540 (2008).
- 11) 松澤芳昭，大岩 元：産学協同によるプロジェクトマネージャ育成システムの提案と実証実験，情報処理学会論文誌， Vol.48, No.3, pp.976-987 (2007).
- 12) Quentin W. Fleming, Joel M. Koppelman (PMI 東京支部訳)：アード・バリューによるプロジェクトマネジメント，日本能率協会マネジメントセンター (2004).
- 13) Campwood Software: SourceMonitor <http://www.campwoodsw.com/sourcemonitor.html> (2009/04/19 に参照).