

クラウド型ソフトウェア開発環境を利用した遠隔地との PBL

中鉢 欣秀*1・小山 裕司*1

Email: yc@aiit.ac.jp

*1: 産業技術大学院大学産業技術研究科情報アーキテクチャ専攻

◎Key Words 遠隔 PBL, クラウド, ソフトウェア開発

1. はじめに

従来のソフトウェア開発型 PBL (Project Based Learning) は教室等で実施し、face to face によるグループワークの形態で行うことが多かった。しかしながら、実務におけるソフトウェア開発では、遠隔地にいるプロジェクトメンバーと協働で開発プロセスを遂行する例も多く見られる。特に、近年ではオフショア開発ということで、海外のメンバーと英語等でコミュニケーションしながらソフトウェアを共同開発する場合も多い。

産業技術大学院大学 (AIIT) では、このようなソフトウェア開発体制の多様化を踏まえ、海外の大学 (ベトナム国家大学) や、国内遠隔地 (琉球大学) の学生と共に、分散型でのソフトウェア開発を経験できる PBL を行っている。遠隔地との分散 PBL を実施すると、開発プロセスやコミュニケーションにおいて発生する課題が、従来の PBL よりも更に強調されることになる。このことは、学生に対して PBL で解決すべき課題の難易度を高めることにつながり、開発の経験者にとっても挑戦しがいのある、実りの多い PBL となる。

本発表では、これまでに遠隔地との PBL を実施して得た知見のうち、クラウド型のソフトウェア開発環境の活用について焦点を当てて論じたい。これらのツールを活用して実施している本学での分散 PBL について述べ、今後の地方教育への展開についても考察する。

2. 遠隔 PBL のための事前学習

遠隔地とのソフトウェア開発プロジェクトを、PBL 教育の一環として実施するには、参加する学生に、共同開発のために利用する開発ツールについての事前学習をしておくことを推奨する。

PBL のために何らかの事前学習を行うということには、議論がある。ある知識を学習する必要性は、学習者がプロジェクト実施中に直面する実課題を経験することで認識する。従って、プロジェクトの実施中に、学習者が知識取得の必要性を認識した後、勉強をすることが効果的である、という考え方もある。

しかしながら、遠隔地とのソフトウェア開発においては、これらのツールの使用方法について事前に学んで置かないと、そもそもの開発プロジェクトのスタートラインに立てない。

また、優れたツールには先人が問題解決のために実装した多くの知恵が詰まっている。教員は予め学習者にツールについて事前学習を行い、一定の理解をさせておく。そして、プロジェクトで実際に使用することでより、ツールの機能についてその本質的意義を更に深く認識できると期待できる。

3. 遠隔地との PBL で利用するツール

本研究では、ソフトウェア開発環境として、言語処理系や OS の基本操作は既に知っているものと仮定する。その上で、特に遠隔地とのプロジェクトのために押さえておきたい、分散開発環境について事前に学習すべき事柄について論じる。

分散型の開発では、複数のソースコードファイルから構成されるソフトウェアを、様々な場所にいる開発者が同時並行で実装していくことになる。これを実現するためには、分散型のバージョン管理システムを利用することが一般的だ。

近年、ソフトウェア開発者が特に注目しているのが「Git」と呼ばれるバージョン管理システムであり、そのリポジトリを一定の制限のもと無料で利用できる「GitHub」と呼ばれるクラウド型のシステムである。分散型のバージョン管理システムには、他にも古くから利用されている CVS (Concurrent Versions System) や SVN (Subversion) は有名だ。PBL においてこれらを利用しているケースも多い。

4. Git の特徴

ここで、他のシステムではなく、Git 及び GitHub を PBL のために事前学習させることの狙いについて述べる。Git は、Linux の開発者である Linus Torvalds が開発した。Linux と言えば、オープンソース型のソフトウェア開発として最も巨大なものの一つである。このプロジェクトのバージョン管理のために、比較的最近である 2005 年になって Linus 自ら Git というツールを改めて開発したことは興味深い。

他の類似するシステムが既に存在するにも関わらず、Linus が新たなツールを開発しなくてはならなかったのは、既存の他のツールでは満足できなかったからだという。そこに新たに開発された Git には、大量のソースコードのバージョンを長年管理し、世界的なオープン

ソース開発を行ってきた Linus 及び開発コミュニティの豊富な知見が含まれていると見るべきである。

実際に、Git に触れてみると、このことがよく分かる。一例として、Git において、ソースコードを変更したときの差分を管理するための「コミット」という概念について述べる。Git では、この「コミット」に基づき、ブランチやマージと行った各種の機能を実現している。つまり、コミットという概念を 1 つ理解すれば、その概念を自然に演繹することにより、ブランチやマージという別の機能を理解することができるようになってくる。

他にも、リモートにあるソースコードの差分の管理など、全てコミットを単位として操作することができる。このように、Git はツールとして非常に筋の良い設計になっている。反面、この事自体が、初心者にとっては Git を理解しづらくしている原因の一つにもなっている。初心者にとって、別なものとして理解している機能が、実は、同一の概念に基づいて実装されているということは、設計の本質的理解をしなければ Git を使いこなすことが難しいということに繋がる。

そこで、Git の実装に含まれる設計概念については、指導者が事前にポイントを踏まえて説明しておくことが求められる。この際、単にコマンドの操作方法を教えるのではなく、その実装の背景にある概念について、しっかり理解させなくてはならない。

学習者がこれらの知識の本質を理解できるのは、PBL での開発プロジェクトにおいて、実際にツールを利用して各種の課題解決を自らが行った時であろう。このことは念頭に置きながらも、概念体系の全体像は予め指導しておいたほうが良い。

5. Git/GitHub の学習項目

Git に関連する学習項目として、コミットメッセージの書き方のガイドラインも説明しておく。特に、遠隔地との PBL では、コミットメッセージに作業内容を適切に記述し、他のメンバーにとって理解をしやすいようにすることが求められる。このためには、作業内容を端的に表現するための文章を構成して表現することが必要だ。

また、Git の遠隔リポジトリを無料で提供する「GitHub」も、遠隔地とのソフトウェア開発 PBL では是非活用したいツールである。GitHub は、Git が提供する様々な機能に加えて、「GitHub Flow」という開発プロセスを提案している⁴⁾。これも、事前の学習項目に加えるべきであろう。

そして、GitHub が提供する課題管理機能の使い方についても、前述のコミットメッセージの書き方と同様、文章の表現法も含めて指導しておくといよい。Wiki を使った文書の管理も、遠隔 PBL で有効に活用できる。

6. enPiT における遠隔 PBL の取り組み

本学では、ベトナムのハノイ市にある、ベトナム国家大学の学生と協働でソフトウェアを開発する PBL を実施してきた。

2013 年度からは、本学の enPiT プログラム²⁾の一環となり、2014 年度はベトナムの他、ブルネイ、ニュージーランドの学生と共に分散 PBL を実施する。また、国内の遠隔地として、琉球大学の学生ともアジャイル型ソフトウェア開発をテーマとして遠隔 PBL を実施する。

これらの PBL の事前学習科目として、「ビジネスアプリケーション演習」を開講している。この授業は発表者（中鉢）が担当し、Git 及び GitHub を PBL で活用するための事前学習を行う。

この科目は、enPiT プログラムの選択科目として提供している。しかしながら、昨年度は、この科目を受講した学生とそうでない学生とで、PBL におけるツールの利用スキルが大きく異なった。そこで、本年は、講義の内容をビデオ教材にすることで、誰でも事前学習できるようにする予定である。

7. おわりに

以上述べてきたとおり、遠隔地との PBL は従来の PBL よりも高度で実践的なスキルを習得するための場として今後も広く活用できる。

特に、クラウド型のツールの本質理解を行うことができれば、実務でも利用できる実践的なスキルの習得に貢献する。今後は、琉球大学との分散 PBL と同様に、enPiT の参加校や連携校を足がかりとし、東京以外の地方教育への展開も進めていきたい。

参考文献

- (1) GitHub Flow – Scott Chacon,
<http://scottchacon.com/2011/08/31/github-flow.html>
- (2) enPiT BizApp - 産業技術大学院大学
<http://enpit.aait.ac.jp/>