

TODO アジャイル開発スタジオの構築と enPiT プログラムでの活用

中鉢 欣秀¹⁾

TODO Style and Layout of an AIIT Bulletin – 2013 (L^AT_EX) –

Yoshihide Chubachi¹⁾

Abstract

TODO Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec aliquet hendrerit dui at. Nunc blandit egestas felis non aliquet. Proin malesuada dictum lacus eget elit accumsan, eu convallis urna malesuada. Donec quis neque erat tempus congue ac eget est. Donec nec dolor auctor, laoreet nisi id, tincidunt metus. Nunc eu scelerisque nisi. Mauris vitae laoreet malesuada risus. Quisque at pharetra quam. hendrerit augue sollicitudin vitae. Proin eget malesuada dictum erat. In ullamcorper leo in volutpat bibendum hendrerit.(approx. 80 Words)

Keywords: TODO AIIT, bulletin, 2013 (approx. 5 Keywords)

1 はじめに

本論文は・・・を再構成した。

2 研究の方法

2.1 アジャイル教材開発スタジオ

TODO

基本コンセプト

昨年の紀要に関連文献なかったか？

3 研究の結果

3.1 プログラムの全体像

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。

3.2 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。

Received on 2014-10-03.

¹⁾ 産業技術大学院大学

Advanced Institute of Industrial Technology

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う効果が期待できる。

3.3 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。

3.4 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いる。

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。

3.5 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(1)。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ことよい。Wiki を使った文書の管理も、遠隔 PBL で有効に活用できる。

3.6 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。

3.7 enPiT プログラムについての考察

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.aiit.ac.jp/>

4 研究の考察

5 はじめに

本稿では産業技術大学院大学紀要の書式について記す。

6 原稿

6.1 カメラレディ原稿

原稿は、日本語もしくは英語による完全版下 (camera ready) 原稿とする。製版後の校正は原則として不可能であるため、誤字や脱字がないよう、特に念を入れて仕上げる。刷り上がりは、6 頁以上が望ましい。

6.2 余白

天地左右余白 (マージン)・段間余白 (コラムスペース) はこの見本に従う。上下の余白には製本時にヘッダとページ番号を挿入するので、空白にしておくこと。

7 標題について

7.1 標題

標題は和文ならびに英文とする。英文原稿の場合は、和文表題を記述する箇所に英文標題を記述し、英文標題の箇所は削除すること。

7.2 アブストラクト・キーワード

和文ではなく英文で記述すること。アブストラクトは 80 語程度とし、キーワードは 5 つ程度とする。

7.3 標題等の割付

見本に従って、[和文標題、和文著者名、英文標題、英文著者名、英文アブストラクト、英文キーワード] 及び [受領日、所属] の割付を行う。

7.4 英文原稿の場合

英文による原稿の場合は、和文著者名のところに英文で記述し、英文著者名のところは削除すること。所属も英文で記述すること。

8 本文について

8.1 句読点

和文の句読点には全角ピリオド (.), 全角コンマ (,) を用いること。

8.2 見出し

原稿には、大見出し、中見出しなどを設け、それらを明瞭に区分する。さらに細分を要するときは著者に委ねる。

9 参考文献について

参考文献は、通し番号とし、本文中では、当該事項または人名などの参考とする後に、[1], あるいは、[2, 3] のように記す。文章の末尾に記す必要がある場合には、句読点の前に記す [4]。

参考文献は、原則として、雑誌の場合は、著者、標題、雑誌名、巻、号、頁、年の順に記す。また、書籍の場合は、著



図 1 図のキャプション

表 1 表のキャプション

	A	B	C
R1	A ₁	B ₁	C ₁
R2	A ₂	B ₂	C ₂
R3	A ₃	B ₃	C ₃
R4	A ₄	B ₄	C ₄

者、書名、発行所、発行年の順に記す。参考文献例を本文の最後に挙げるので参考されたい。

10 脚注等について

10.1 脚注

脚注は段組の下部に記載する*1。

10.2 ルビ・圈点

難読漢字にはルビを振ることができる。また、強調したい箇所には圈点をつけることができる。

11 図・表について

11.1 図表のキャプション

図・表には、図 1、図 2、表 1、表 2 のように論文全体で通し番号をつけること。英文の場合には、Fig. 1, Fig. 2, Table 1, Table 2 のように、番号をつけること。通し番号、標題は本文と同じ書体を使用すること。

表のキャプションは表の上に、図のキャプションは図の下につけること。

11.2 図表に関する注意

図・表は、印刷に十分耐えうるものでなければならない。刷り上がり時の文字が小さすぎないように十二分に配慮し、線の太さにも注意する。

図・表に色刷りを必要とする場合は、別途連絡すること。ただし、製本上の都合で色刷り頁を設けることができない場合もありうる。

*1 脚注の例

12 テンプレート

L^AT_EX を利用する執筆者は、`aiit.bulletin.tex` を書き換えて使うこと。また、**MS Word** を利用する場合は、`aiit.bulletin.docx` ファイルを用いること。

13 おわりに

本稿では産業技術大学院大学紀要のフォーマットについて記した。

参考文献

- [1] 奥村晴彦. **L^AT_EX** 美文書作成入門. 技術評論社, 2010.
- [2] H. Takeuchi and I. Nonaka. The new new product development game. *Harvard business review*, Vol. 64, No. 1, pp. 137–146, 1986.
- [3] J. Sutherland and J. Schwaber. Scrum guide. <http://www.scrum.org/Scrum-Guides>, 2011. (visited on 2013-06-30).
- [4] 独立行政法人情報処理推進機構 (IPA). 「IT 人材白書 2012」行動こそが未来を拓く～進むクラウド、動かぬ IT 人材～. 独立行政法人情報処理推進機構 (IPA), 2012.

付録 A 注意事項

A.1 フォントの準備

LaTeX と **Word** で同じ字体に揃えるため、和文には「IPAex フォント (2 書体パック)」を用いる。次のサイトからインストールすることができる。

<http://ipafont.ipa.go.jp/ipaexfont/download.html>

なお、**TeX Live 2012** 以降および **W32TeX [2013/06/06]** 以降の `luatexja.tar.xz` に は同梱されている。

A.2 版面と余白

何らかの事情で書式を自分で設定する場合、本文の書式は次の通りとする。

- フォントの大きさは **8.5pt(=12Q)** とする。
- **2 段組**とし **1 行 27 文字**で、段間は **2 文字分**あける。
- 行間は字の高さの **75%** とし、**1 頁 47 行**とする。
- 余白は天 (上部) **30mm**, ノド (内側) **25mm** とする。

これら以外の書式は可能な限りこの見本に合わせる。