

# ソフトウェア開発 PBL の 定量的評価

—クラウドコンピューティングの活用—

井垣 宏<sup>1</sup> 福安直樹<sup>2</sup> 佐伯幸郎<sup>3</sup> 松本真佑<sup>3</sup>

<sup>1</sup>大阪大学大学院情報科学研究科 <sup>2</sup>和歌山大学システム工学部 <sup>3</sup>神戸大学大学院システム情報学研究科

## Cloud Spiral

我々は文部科学省の「情報技術人材育成のための実践教育ネットワーク形成事業」である「分野・地域を越えた実践的情報教育協働ネットワーク」(略称: enPiT)の一環として、クラウドコンピューティングの実践的活用ができる人材教育を目指したカリキュラム「Cloud Spiral」を2013年から実施している<sup>1)</sup>。Cloud Spiralは、大阪大学・神戸大学を中心とした関西9大学と関東、関西の複数企業が連携し、前半ではクラウド環境を利用したチームによるソフトウェア開発手法、後半ではクラウド環境を対象とした負荷分散や大規模データ処理といった各種技法の獲得を目的として構築されている。本解説では、Cloud Spiralの前半部分の最後の8月19日～23日の期間に49名の受講生を対象として実施されたソフトウェア開発PBL (Project-based Learning) 「クラウド基礎PBL」(図-1)について紹介する。



図-1 クラウド基礎PBLの実施風景

## ソフトウェア開発 PBL における課題

ソフトウェア開発PBLは、ソフトウェア開発プロジェクトをテーマとし、ソフトウェア開発に伴うさまざまな課題をチームで管理・解決する学習手法である。この種のPBLを実施するにあたって我々は、PBL環境と開発プロセスに関する以降に述べる課題に直面した。

### □ PBL 環境

一般に、チームによるソフトウェア開発では、アプリケーションを実装し、実行するための開発環境(統合開発環境やWebサーバ、アプリケーションサーバ等)とリソース共有のための版管理システムやプロジェクト管理ツールといった開発支援環境が必要である。ソフトウェア開発PBLにおいても、チーム内のすべての受講生がPBLのデザインに応じた共通の環境(PBL環境と呼ぶ)を利用する必要がある。しかしながら、コンパイラやライブラリ、サーバ等のバージョンまで完全に同一の環境を導入することは容易ではない。また、導入が完了したとしても、初期導入されていた開発とは無関係なアプリケーションや、アップデートなどの実施により意図しない不具合が発生することもあり得る。結果としてPBL環境の導入や維持に要するコストは非常に高くなることがある。

## □ 開発プロセス

ソフトウェア開発 PBL では、ソフトウェア開発におけるライフサイクルを体験し、開発プロセスに関する知識や技術を実体験に基づいて獲得することが1つの目的である。そのためには、受講生自身が所属するチームの開発プロセスを定期的に振り返り、問題点を考察し、工夫・改善するという PDCA のサイクルを回すことが重要となる。しかしながら多くの PBL において、受講生はプロセスを省みず、プロダクトを完成させることのみ注力しがちである。また、開発効率を重視するあまり、チームの特定の受講生に作業が偏る場合も少なくない。そのようにして実施されたプロジェクトでは、受講生ごとに得られるスキルや知識の大幅な偏りや、開発プロセスの理解や改善といった機会の喪失が発生することがある。

プロセスについては、プロダクトと比較して評価が難しいという課題もある。教育においては、受講生のモチベーションを向上させる1つの方法として、成果に対する適切な評価を与える必要がある。評価を行うためには、基準を明確にした上で受講生に事前に開示し、受講生から集約したデータに基づいて適切な判断を行う必要がある。しかしながらプロセスの指標は、納期のような一部の項目を除いて、データの収集や定量的な定義が困難であることが多い。結果として、評価基準が曖昧になり、プロジェクト中の開発プロセスが軽視されやすくなることになる。

## Cloud Spiral における 3 つの取り組み

### □ PBL as a Service

PBL 環境における課題改善を目指すにあたり、PBL 環境そのものを図-2に示すようにクラウド化 (PBL as a Service) することを考えた。受講生が利用する開発環境や開発支援環境はクラウド上に仮想化されて実現される。これにより受講生は各自の仮想マシンにネットワーク越しにアクセスするだけで、すべての開発環境と開発支援環境をセットアップさ

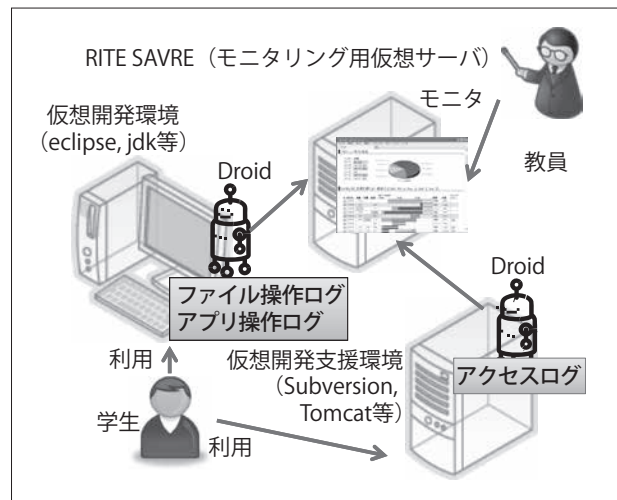


図-2 PBL as a Service

れた状態で利用できるようになる。

さらに、仮想化された開発環境を利用することで、全受講生のあらゆるログを Droid と呼ぶログ収集エージェントがモニタリングサーバ (RITE SAVRE) に集約することが可能となった。

### □ Quality/Assignment/Delivery

通常のソフトウェア開発プロジェクトでは QCD と呼ばれるプロジェクト評価基準が用いられることが多い。これらはそれぞれ、Quality (プロダクトの品質)、Cost (プロジェクトに要したコスト)、Delivery (納期) を表している。

我々はクラウド基礎 PBL において、QCD のコンセプトを変更し、Quality (プロセスとプロダクトの品質)、Assignment (タスク割り当ての均等化)、Delivery (納期) の QAD をプロジェクトの評価基準として策定した。QCD から変更した内容は2つ存在する。1つ目の変更は、Quality の観点においてプロセスの品質を具体化したことである。クラウド基礎 PBL では、受講生が開発すべきプロダクトの種類 (java ファイル, html ファイル等) ごとに、実装、単体テスト、レビューといった実施すべきタスクを教員が図-3のように定義し、受講生に周知した。この図は java ファイルを対象とした開発フローを示しており、開発完了には実装、単体テストおよびレビューが実施されなければならないことを表して

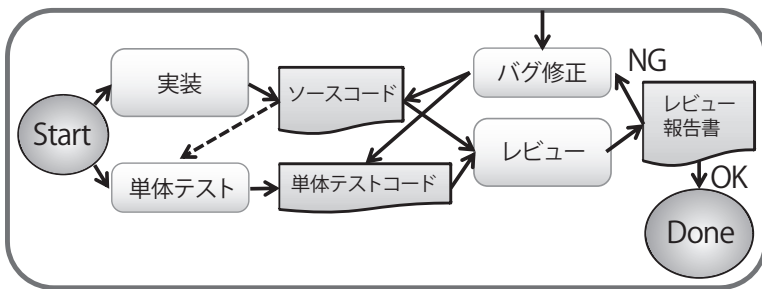


図-3 開発フローの例

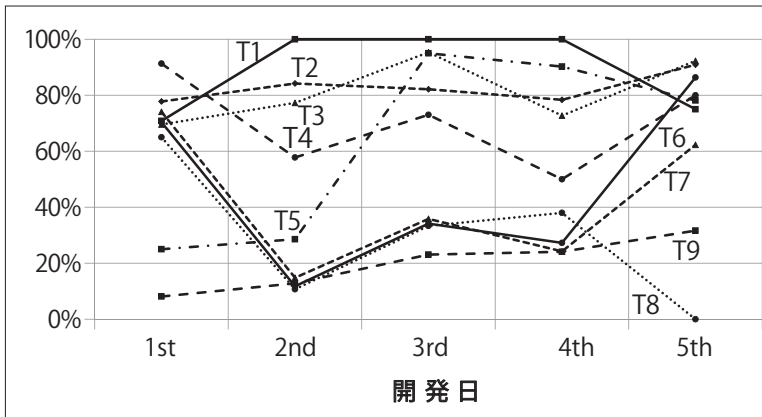


図-4 コミット時ビルド成功率

いる。このようなプロセスに関連する守るべきルールを策定しておくことで、ルールの遵守度合いとして、受講生のプロセスを評価することが可能となる。

2つ目の変更は Assignment である。これは Quality の項目で策定した実装、単体テスト、レビューといったさまざまな種類のタスクをチームのメンバでどの程度均等に担当しているかを表す指標となっている。以下は本 PBL で実際に受講生に提示した Assignment にかかわる指標の1つである。

「開発終了時に各メンバの単体テスト実施回数がチームの平均値×0.8～1.2以内に収まること」

我々はこのような指標をレビュー、単体テスト、実装といったさまざまな種類のタスクごとに策定した。この指標を守るためには、タスクごとの実施回数を受講生自身が開発中に計測し、作業の偏りが発生しないよう調整する必要がある。

#### □ チケット駆動開発の導入

チケット駆動開発とは、開発者の実施するタスクをチケットとして登録しておき、開発者のタスク

実施状況とチケットの状態を連動させることによって開発を駆動するアジャイルソフトウェア開発手法である<sup>2)</sup>。チケットに入力された担当者情報や「着手/完了」といった状態を見ることで、誰が何を実施しているのかを容易に把握することができる。また、タスクの実施結果がチケットとして残るため、誰がどのタスクをいつ実施したのかといった情報を用いて、前節で述べた QAD の各項目を定量的に評価することが可能となる。以降ではこれらの取り組みの適用結果について紹介する。

### クラウド基礎 PBL への適用とその効果

#### □ PBL へのクラウドの適用

我々は国立情報学研究所「edubase Cloud」<sup>3)</sup>を用いて、すべての PBL 環境のクラウド化を実現した。これにより、環境導入・保守コストが大幅に削減された。実際に、1週間の開発期間において、受講生の開発環境に依存するトラブルは1件も発生しなかった。実施後のアンケートでは、開発環境のクラウド化について、約80%の受講生が良いと評価した一方で、通信のオーバーヘッドや開発環境のカスタマイズの自由度がないことに関して約20%の受講生が不満を感じていた。通信のオーバーヘッドについては、49名もの受講生が同時にアクセスする環境であったことや無線LANでの運用であったことにも起因すると思われる。開発支援環境やモニタリング環境についても同様に、複数チームのための環境を1つ作成し、複製するという形で、容易に構築できることが確認できた。

#### □ 評価基準(QAD)とチケット駆動開発手法の適用

チケット駆動開発と QAD の導入により、受講生にとっても教員にとっても、プロジェクトの評価基準が明確になるというメリットが得られた。図-4

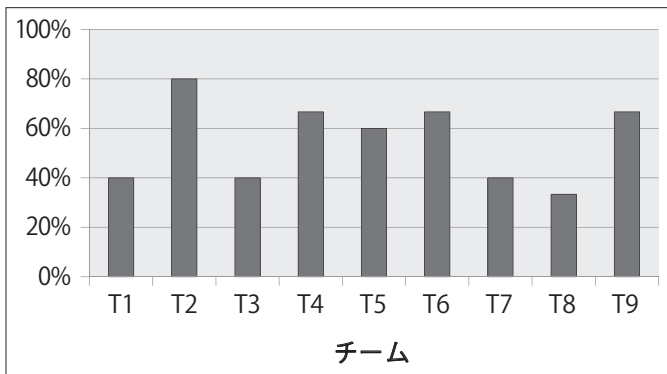


図-5 単体テストにおける Assignment 遵守率

は Quality に関して我々が定めたルールの一つである「ソースコードを版管理システムにアップロード(コミット)する際に、コンパイルと単体テストが正常に実行できること」について、計測した結果である。このグラフは、各開発日(横軸)におけるチームごとの総コミット回数のうち、どの程度の割合(縦軸)でコンパイルと単体テストが正常に実行できたかを示している。開発最終日において成功率が下がっているチームに状況を確認したところ、「締切が近づいたため、コミット内容について精査する余裕がなくなった」との回答が得られた。

図-5は、教員が策定した単体テストにおける Assignment 指標を遵守できたメンバの割合を示している。たとえば、T2の Assignment 遵守率は80%である。これはT2のメンバ数5名のうち4名までが、単体テストにおける Assignment 指標が基準値以内であったことを意味する。半数弱のチームで50%以下という値(すなわちメンバの半数が守れなかった)になっているが、これはそのチームが Assignment 指標遵守を放棄したというわけではなく、特定のメンバが0.8~1.2という基準値からほんの少し外れていた程度であった。また、PBL終了後のアンケートでは複数の受講生から、「Assignmentに関するルールのおかげでさまざまな知識を習得できた。またチーム内の教え合いが活発に行われ、結果としてチームビルディングの助けになった」との回答があった。

クラウド基礎 PBL では、ほかにも開発フロー

の遵守率、テストにおける網羅率といった QAD に関する定量的なプロジェクトの計測・評価を実施した。プロジェクトに関する指標が評価基準として明確になり、計測・可視化されることで、タスクの割り当てやプロセスの記録、プロセス品質の改善といった、これまで軽視されがちだった納期以外の項目について、多くのチームが意識するようになった。各開発日に実施される振り返りにおいても、全チームが QAD の項目に関する分析と改善のための具体的

な工夫について、客観的なデータに基づいて議論を行っている様子が確認できた。教員としても、評価基準が具体的かつ定量的になることで、一貫性のある指導が容易になったと考えている。

今後はチームレベルだけでなく、受講生一人ひとりについてのフィードバック内容を充実させていくことを検討している。

#### 参考文献

- 1) Cloud Spiral, <http://cloud-spiral.enpit.jp/>
- 2) 小川明彦, 阪井 誠: チケット駆動開発, 翔泳社 (Aug. 2012).
- 3) edubase Cloud, <http://edubase.jp/cloud/>

(2013年8月31日受付)

井垣 宏 (正会員) [igaki@ist.osaka-u.ac.jp](mailto:igaki@ist.osaka-u.ac.jp)

2005年奈良先端科学技術大学院大学博士後期課程修了。博士(工学)。同大学院特任助手。南山大学講師。神戸大学特命助教。東京工科大学助教を経て、2011年より大阪大学大学院情報科学研究科特任准教授。ソフトウェア工学教育に関する研究に従事。

福安直樹 (正会員) [fukuyasu@sys.wakayama-u.ac.jp](mailto:fukuyasu@sys.wakayama-u.ac.jp)

2000年名古屋大学大学院工学研究科情報工学専攻博士後期課程修了。博士(工学)。同年和歌山大学助手。2007年より同助教。ソフトウェア開発環境やソフトウェア工学教育に関する研究に従事。

佐伯幸郎 [sachio@carp.kobe-u.ac.jp](mailto:sachio@carp.kobe-u.ac.jp)

2009年高知工科大学大学院工学研究科博士後期課程修了。博士(工学)。同大学助手。同助教を経て2013年より神戸大学大学院システム情報学研究科特命助教。デジタル信号処理・ソフトウェア工学教育に関する研究に従事。

裕本真佑 [shinsuke@cs.kobe-u.ac.jp](mailto:shinsuke@cs.kobe-u.ac.jp)

2010年奈良先端科学技術大学院大学博士後期課程修了。博士(工学)。同年神戸大学大学院システム情報学研究科特命助教。エンピリカルソフトウェア工学の研究に従事。