

Scrum に基づきコミュニケーションを重視した ソフトウェア開発 PBL の実践

木崎悟^{†1} 田原康之^{†1} 大須賀昭彦^{†1}

我々は、協創的なソフトウェア開発者の育成を目指し、産学連携で対話をしながら PBL (Project-Based Learning) 環境を整備してきた。2011 年度は、ソフトウェア開発プロセスにアジャイル手法を採用して、小規模なプロジェクトを試行した。しかし、メンバの技術力の差による個人への負荷と開発遅延、顧客と不平等な関係、メンバ同士でのコミュニケーション不足が問題となった。2012 年度は、これら問題を解決するために、開発プロセス「Scrum」を採用した。そして、プロジェクトの結果を分析し、メンバの多様性を活かした問題解決、顧客のプロジェクトへの積極的参加によるチームとの関係改善、チームの自主的に活動できる「場作り」によるコミュニケーション基盤の構築がプロジェクト成功に役立つことがわかった。

Practice of the software development PBL which thought communication as important based on Scrum

SATORU KIZAKI^{†1} YASUYUKI TAHARA^{†1} AKIHIKO OHSUGA^{†1}

We improved the PBL (Project-Based Learning) environment aiming at the collaborative-software development education by industry-university cooperation. In the fiscal year 2011, the agile technique adopted as the software development process, and the small-scale project was tried. However, there were problems like the shortage of communication between members, the customers and difference of a member's technical capabilities and the burden was placed to the individual. In the fiscal year 2012, in order to solve these problems, the development process "Scrum" was adopted. We analyzed the result of projects. In the result, Scrum was being able to perform problem solving in which the member's diversity was harnessed. It could carry out improved relations as the team by the positive participation in customer's projects. And, it could build the communication base by "making the place team". Therefore, we understood using Scrum for success of projects.

1. はじめに

近年、情報系教育機関において、Project Based Learning(以下、PBL)の形式で授業が実施される例が増えている[1, 2, 3]. PBL とは、具体的な課題のあるプロジェクトに取り組み、問題解決を自ら行うことでコンピテンシーの向上を図り、実社会において有用な人材の育成を行う教育手法である。PBL の導入により、ソフトウェア開発の工程を体系的に学ぶことが期待されている。さらに、学習者のコミュニケーション能力や問題解決能力も育成できるとされている。

ソフトウェア開発 PBL の場合、模擬プロジェクトという形式で仮想的に顧客を設定して、実施されるケースが多い。それらのプロジェクトの場合、要件定義が最初から設定されており、その要求を満たすことでプロジェクトが完了する。しかし、実際のソフトウェア開発において、このような「言われた通りに作る」という受託開発やシステムインテグレーションは時代遅れである。ユーザーのニーズを的確にとらえられていなければ、プロジェクトの成果は期待できない。

慶應義塾大学 SFC では、学部生を対象に PBL 型の科目「協創型ソフトウェア開発」を実施している。この授業は、

2005 年度秋学期に開講され、2012 年度で 7 回目である。毎年、複数の開発チームが結成され、プロジェクトのメンバは 3~5 名で構成される。また、プロジェクトマネージャは社会人技術者が担当し、実際にシステムを必要とする企業や個人の顧客が設定され、ユーザーに利用されるソフトウェアの開発を目標としている[4, 5, 6].

2011 年度は、ソフトウェア開発プロセスにアジャイル手法を採用して、小規模なプロジェクトを試行した。しかし、メンバの技術力の差による個人への負荷とそれに伴った開発遅延、顧客と不平等な関係、メンバ同士でのコミュニケーション不足が問題となった。

2012 年度は、これら問題を解決するために、開発プロセス「Scrum」を採用した。そして、プロジェクトの結果を分析し、メンバの多様性を活かした問題解決、顧客のプロジェクトへの積極的参加によるチームとの関係改善、チームの自主的に活動できる「場作り」によるコミュニケーション基盤の構築がプロジェクト成功に役立つことがわかった。本論文では、アジャイル型開発プロセス「Scrum」[7]をベースとして実施された 2012 年度のプロジェクトを通して得た知見を報告する。

構成は以下の通りである。まず、2.でアジャイル型ソフトウェア開発 PBL について説明する。次に、3.で具体的なプロジェクトの実施について、4.でプロジェクトの結果につ

^{†1} 電気通信大学大学院情報システム学研究所
Graduate School of Information Systems, The University of
Electro-Communications

いて、5.で評価と考察を行い、6.で今後の課題を述べ、最後にまとめる。

2. アジャイル型ソフトウェア開発PBL

2.1 アジャイル開発

アジャイル開発とは、変更に対して柔軟に対応することに主眼を置く軽量な開発プロセスである。2010年の米国フォレスター・リサーチ社の調査[8]によると、米国におけるアジャイル開発の採用は35%である。アジャイル開発手法には、エクストリーム・プログラミング（以下、XP）[9]やScrumなどが考案されている。

2.2 ソフトウェア開発PBLへのアジャイル導入の問題点

日本において、ソフトウェア開発PBL形式でアジャイル開発を実践する場合、各教育機関が独自に実施する例がある。しかし、技術と知識の習得が目的になっているため実際の顧客がいない場合や要求変更が起こらない状態で実施されている。さらに、指導する側の教員がアジャイル開発に不慣れであり、効果的な学習ができない場合が多い。

2.3 アジャイル型のソフトウェア開発教育

アジャイルの先進国である米国では、アジャイル型開発をソフトウェア開発教育に取り入れている。フォートリス大学では、サービス・ラーニングモデルに基づき、クライアントのためのアプリケーションを開発するプロジェクトを実施している[10]。非アジャイル型の開発手法を採用していた時期は失敗プロジェクトもあったが、アジャイル型を採用後は、すべてのプロジェクトが成功することができた。カーネギーメロン大学のFeng Jiらの研究[11]では、ウォーターフォール型とアジャイル型(XP)の開発手法を比較して、同じ機能を完成させたが、ウォーターフォール型の場合、アジャイル型よりも、多くの時間をコーディング、設計書などのドキュメント作成に費やしたとしている。また、アジャイルについてPBLの形態で実施している。ここでは、産学連携により、実際の顧客が参加すること、顧客とゴールを共有することで、チームのコラボレーションやコミュニケーションの重要性を学んでいる。さらに、欧米ではアジャイルコーチを起用して、技術情報や経験が伝播されている。

2.4 Scrum

Scrumとは、欧米などで最も多く普及しているアジャイル開発手法である。竹内弘高氏・野中郁次郎氏による日本製品開発における研究[12]が基礎となり、J.Sutherlandにより、ソフトウェア開発プロセスとして体系化した。その名の通りチームが一丸となって開発を進められるように、プロジェクト運営の側面に焦点を当てている。

メンバの役割は、責任者として要件と優先度を定める「プロダクトオーナー」、プロジェクトが円滑に進むようにサポート（アドバイス）をする「スクラムマスター」、開発者の集団である「チーム」から構成されている。

進め方は、プロダクトオーナーが要件を「プロダクトバックログ」と呼ばれる機能単位に分割する。機能に関してはストーリー形式で記述する。次に、スクラムマスターとチームはプロダクトバックログの機能を実現するために実施する作業項目（タスク）を洗い出して、「スプリントバックログ」を作成する。

スプリントバックログは、プロダクトオーナーが決めた機能をより詳細化する。内容はスプリント計画会議で決定される。スプリントとは、最大4週間と定められた開発期間（イテレーションサイクル）のことを指す。各スプリントの長さは一定であり、1つのスプリントで出荷可能な製品を作成する。また、毎日の作業は、デイリースクラムから始まる。デイリースクラムでは、作業報告と問題共有をチームのメンバ全員で行う。そして、タイムボックスという一定の時間を決めて作業する。作業終了時には、KPT法[13]などにより、レトロスペクティブ（振り返り）を行う。

2.5 アジャイル型ソフトウェア開発PBLの実施

我々の取り組みであるアジャイル型ソフトウェア開発PBL「協創型ソフトウェア開発」について述べる。本授業はアジャイル型開発手法「Scrum」を採用している。

本授業のプロジェクトモデル（図1）では、プロジェクトでは、顧客（プロダクトオーナー）を設定している。また、IT企業からスクラムマスターが参加して各チーム（3～5）に加わる。そして、教員の他にアジャイルの専門家（アジャイルコーチ）[14]が参加して手法に不慣れなチームをサポートする。このモデルを授業に適用することにより、学生は実際のソフトウェア開発に限りなく近い環境で学ぶことができる。

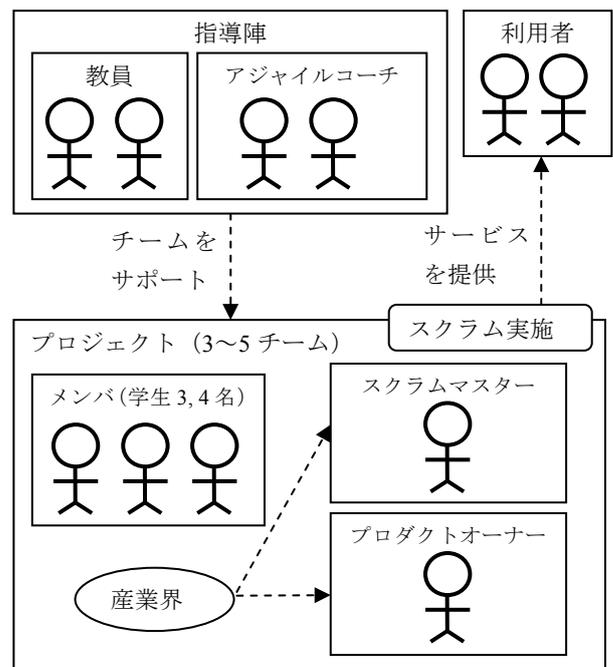


図1 プロジェクトのモデル

Figure 1 Model of a project in the proposed PBL system.

2.6 アジャイル型ソフトウェア開発 PBL の評価

アジャイル開発では、顧客にとってビジネス価値の高いソフトウェアを生み出すことが望まれる。そのため、アジャイル型 PBL におけるプロジェクト評価指標は、ビジネス価値が高い製品を作れたかという点になる。製品が作れたとしても、顧客満足を得ることができなければ、プロジェクトは失敗である。しかし、実務経験がない学生は、顧客満足が得られる製品を作ることができず、プロジェクトが失敗するケースがある。本授業でも、プロジェクトの評価基準を顧客（プロダクトオーナー）の評価とした。顧客からよい評価を得ることができればプロジェクトは成功である。逆に顧客が満足しなければ、プロジェクトは失敗である。

2.7 アジャイル型ソフトウェア開発 PBL の試行

本授業は、2005 年度秋学期に開講されてからウォーターフォール型や RUP などのイテレーション型の開発手法を採用していたが、2011 年度はアジャイル型のみで実施することになった[15, 16]。授業という性質上、時間および人的に制約があり成果物が完成せずに授業期間が終わってしまうプロジェクトもあったためである。

2011 年度は、アジャイル型開発手法を取り入れたプロジェクトが 3 チーム結成された。それぞれのチームは XP のプロセスを参考に、試行錯誤しながら、アジャイル開発を進めることになった。学生への授業アンケートでは、「企業や大学から依頼を実際に受けて開発するということに意味があった」と肯定的な意見もあったが、アンケート全般では、メンバの技術力の差とそれに伴う開発遅延、顧客との不平等な関係、メンバ同士でのコミュニケーション不足などの問題が指摘された。そのため、これらの問題解決が、プロジェクトを成功に導くための手段だと考察した。

2.8 2012 年度における対策案

2012 年度はアジャイル開発手法の中でも Scrum を採用した。Scrum は、「コミュニケーションコストをなるべく上げず、有機的なチームを作ること」を目的としている。

Scrum ではチームの最適な人数を 3~10 人としている。例えばミーティングにおいて、少ない人数の方がチームの置かれている状況が把握しやすいなど利点がある。また、チームの課題を解決するために多様性が重視される。開発者、デザイナー、インフラなどから様々な知見がでるからこそ問題点を解決できるとしている。

2011 年度のプロジェクトでは、問題が発生しても技術的に強いメンバが対応にあたるような傾向があった。問題点に関してチームで検討する時間をとれば、解決できた可能性があった。また、Scrum では顧客もプロダクトオーナーという形でプロジェクトに参加する。そのため、チームとの親密な関係が構築できる。さらに、メンバ同士のコミュニケーション不足についても、Scrum のフレームワークを採用することで、問題点や疑問点をチーム内で共有するプ

ロセスであるため、早期解決が期待できる。

これらの見解から 2012 年度は、Scrum に基づきコミュニケーションを重視したソフトウェア開発 PBL を実践することになった。

3. 2012 年度のプロジェクト

3.1 事前学習

2012 年度は、20 名の学生が授業を履修した。プロジェクト開始前に、アジャイル開発についての理解と、ソフトウェア開発環境やコラボレーションツール (GitHub [17]) に関する講義（授業の 2 回目, 3 回目）を行った。実際に、方法論とツールを利用する前にこれらの理論やツールの使用方法をプロジェクトに参加する教員、社会人技術者、学生が理解する必要がある。講義をするのは、後述のアジャイルコーチの役割である。彼らは毎週の授業に 1~5 人参加した。アジャイルコーチによる知識の伝播により、方法論やツールを効果的に活用することが可能となった。

3.2 プロジェクトの結成

3 回目の授業でプロジェクトの企画案が各プロダクトオーナーから発表された。それぞれの企画は、企業からの案件から、個人でほしいアプリケーションの作成依頼まで様々であった。

これらの要求から興味のあるものを希望した学生がそれぞれのプロジェクトに参加した。プロジェクトは、5 プロジェクトで各チーム 4~5 名の学生が集まった。さらに、スクラムマスターになる社会人技術者の紹介がされ、各チームでスクラムマスターを選択する形をとった。

プロジェクト A, プロジェクト C, プロジェクト E は、スマートフォン向けのアプリケーション作成。また、プロジェクト B は Facebook アプリケーションの開発、プロジェクト D は、iPad 対応のソフトウェア開発を行った。

3.3 アジャイルコーチの参加

アジャイルコーチとは、アジャイルの導入支援を行っている専門家である。アジャイルコーチは、アジャイル開発をプロジェクトに取り入れようとしている企業に対して、集合研修を実施する、オンサイトで支援する、イベントでの登壇を行うのが主な仕事である [18]。

彼らが必要な理由は、Scrum などのアジャイル開発のフレームワークには多くのプラクティスが含まれているためである。アジャイル開発未経験の状態では、それらのプラクティスを一度に導入することは不可能である。学生のプラクティスの理解度により混乱が起き、開発作業が進まなくなってしまう。どのプラクティスを実践し、何をするかを、彼らに参加することで的確なアドバイスをすることが可能となった。

3.4 チームのイテレーションサイクル

各チームは 2.4 で紹介した Scrum のイテレーションサイクル（スプリント期間）を 2 週間単位で実施した。本授業

は、1週間に2コマ連続の科目である。授業の3回目以降は、プロジェクト毎の作業となる。メンバ同士が机を向かい合わせ、お互いに会話をしながら作業できる環境を作った。最初の週はスプリント計画会議を行い、スプリント期間に何をするか決定した（その後、2週間に1回実施）。

毎週の授業では、まずデイリースクラム（スクラムミーティング）が行われる。最初の15分間に1) 前回以前にした作業 2) 次回までにやる作業 3) 困っていること をチーム内で情報共有した。

次にタイムボックスでは、ブレインストーミングによりアイデア出しや、チームの開発作業を実施した。授業の最後の15分間にKPT法などの振り返り（レトロスペクティブ）手法を利用して、プロジェクト進行に関する改善案を検討した。授業内のミーティングでは、スクラムマスターも参加し、議題のファシリテートを行った。スクラムマスターは作業には参加せず、スムーズに開発できるようにチームを補助した。

スプリント期間の終わりには、プロダクトオーナー（クライアント）に対して、成果物のデモンストレーションを行い、評価を頂いた。そして、評価後に結果を元にした振り返りがチーム内で実施された。

3.5 コミュニケーション基盤の構築

コミュニケーションは、単にプロジェクトに関わる情報交換といったレベルだけでなく、プロジェクトへの不安やおびえについても、理解し、励ましたり、サポートできる関係を構築することが欠かせない。

本授業における授業時間外の作業時間であるが、授業後のアンケートでは、1時間～4時間の学生が約6割ともっとも多かった。次に5時間～9時間の学生が約2割であった。授業時間外に全く作業していない学生はいなかった。

授業時間内だけでなく、授業時間外の作業も必要となることから学生同士がコミュニケーションをとれる環境作り（場作り）はもっとも重要である。コミュニケーションツールには、SNSとして急速に広まりつつあるFacebookを採用した。全チームがそれぞれのグループを作成して、コミュニケーションに利用した。

そして、プロジェクトのコミュニケーション円滑にするために、重要な役割をするのがスクラムマスターである。スクラムマスターは、2.4で述べた通り、プロジェクト活動を円滑に進めるのが役割である。本授業では、スクラムマスターは社会人技術者が担当する。それぞれのチームに1人ずつ、スクラムマスターが配置され、学生のサポートを行う（原則、毎回の授業に参加）。

スクラムマスターの役割は、「チームをサポートすること」と漠然としており、特定の作業をしないため、それぞれのスクラムマスターが独自に作業を決定した。また、社会人技術者の経験年数も様々であり、経験年数が多いスクラムマスター程、多くの作業をしている傾向があった。ス

クラムマスターの最も重要な役割は、チームの雰囲気作りであり、メンバ同士がコミュニケーションを取ることを第一に考える。

本授業は社会人教育の意味もあり、スクラムマスターの対象はScrum未経験の技術者である。しかし、業務経験が長いほど、プロジェクトに支障をきたす箇所に気付く可能性は高いということが言えた。一例として、大手SIerに勤務しているプロジェクトAの作業内容を表1にまとめる。

表1 スクラムマスターの作業内容

Table 1 The work of scrum masters.

スクラムマスター	作業内容
大手SIer勤務のプロジェクトマネージャ（プロジェクトA）	<ul style="list-style-type: none"> ・ファシリテート ・チームビルディング ・保有スキルの確認 ・役割の割当て ・作業のスケジュール作成 ・コミュニケーションの場作り ・コーチング ・仕様調整 ・意識作り、動機付け ・場の雰囲気作り ・学生の成果を褒める ・学生の話聞く

4. 結果

4.1 プロジェクトの成果

各チームで製品が作成され、成果物が最終発表会で報告された。そして、最終発表会後に授業に参加した学生（20名）とそれぞれのチームのスクラムマスター、プロダクトオーナーにアンケートを実施した。

プロダクトオーナーからの要求に対して、69%の学生が「完成した」と述べた。そして、「(部分的に)完成した」と回答した学生が12%であった。

プロジェクトCのメンバのみ、「未完成に終わった(19%)」という評価であった。このチームでは、プロダクトオーナーとのコミュニケーションがほとんど取れていなかった。「製品を作ること」に主眼が置かれてしまったため、本当に必要な製品ができなかったと推察する。

プロジェクトDも製品が不完全であったという回答だったため、「(部分的に)完成した」という評価であった。このプロジェクトではメンバが協力して開発することが出来なかった。連絡手段としてFacebookを使ったが見ていないことも多かった。そして、メンバ間のコミュニケーションが少なかつたため、スキルの高いメンバが他のメンバに教えるということがほとんどなかった。

プロジェクトの成功例として、プロジェクトBを挙げる。このプロジェクトは、Facebookアプリケーションを開発した。プロダクトオーナーは、自社で開発・運用している漫画総合コミュニティサイトのFacebookページにアプリケーションを設置して、「Like」を集めたいという要求だった。このプロジェクトの場合、開発当初からサンプルアプリケーションと要望書を頂いて、開発イメージがついた。開発

初期は、まずそれぞれのスキルを確認して、足りない部分は勉強することで補うといった方針であった。

しかし、サンプルアプリケーションを元に開発を進めたが、サンプルアプリケーションのソースコードの理解が難しく、メンバのスキル不足もあり、開発を進めることができなくなった。そこで、既存のオープンソースを組み合わせて（再利用し）つつ、最初から作り直した。車輪の再発明はしない方針で開発コストを抑えつつプロジェクトを進めた。中間発表会では動作するアプリケーションを発表することができ、課題を整理しつつ、優先度順に作業を行った。作業時間は他のチームと比較少ないものとなったが、授業時間中に着実に開発できたため、良好な結果が得られた。プロダクトオーナーの要求通りに、ユーザーからの「Like」数増加につながる1つの要因となった。

チームでは、「授業内の進行はスムーズに行えた」「アジャイル型の開発方法が身についた」「役割分担がうまくいった」などの評価があった。

4.2 2011年度とのプロジェクト比較

アジャイル開発プロジェクトの失敗理由に手法への不慣れが挙げられている。2011年度は、アジャイルコーチが参加し、アジャイルに関する指導が行われたが、プロジェクトは、プロジェクトマネージャに任せられた。また、チームのメンバは教員が決めていた。顧客の授業への参加であるが、2011年度の場合は、ほとんどなかった。また、コミュニケーションツールにはRedmineを採用したが、学生から使われることがなかった。それと比較して2012年度はチームも学生主体で決めた。そして、活発的に顧客（プロダクトオーナー）が授業に参加した。コミュニケーションツールは、学生が利用しているFacebookを採用した。

5. 評価と考察

5.1 プロダクトオーナーからの評価

それぞれのプロジェクトで、完成した製品に関して、プロダクトオーナーから評価をいただいた。評価基準は、大変不満足、不満足、普通、満足、大変満足の5段階評価とした。プロジェクトA、Bはプロダクトオーナーからの良い評価を受けることができた。プロジェクトAでは、プロダクトオーナーの要求をすべて満たすことができた。4.1で解説したプロジェクトBでは、プロダクトオーナーの要求以上の製品ができたため評価がよかった。

しかし、プロジェクトCは、アプリケーションとして完成されていなかったため、プロダクトオーナーからの評価は「不満足」であった。

プロジェクトDは、4.1で述べた通り、製品が不完全である。クライアントの評価は、今回の製品は、実用することできないが、これから先の開発で参考にするといった評価であった。

プロジェクトEは最低限の機能は実装されているという

意味で評価は「普通」であった。この点から、要求である最低限の機能を満たすことができて、プロジェクトBのような要求以上の付加価値がなければ、プロダクトオーナーは満足しないことがわかった。付加価値を付けるためには、プロダクトオーナーが実現したい内容を的確にとらえる必要があった。

5.2 コミュニケーションと成果の関係

各チーム内のコミュニケーションとプロジェクトの成果の関連について考察する。各チームにアンケートを取り、コミュニケーションについて「よくとれていた」「普通」「不十分だった」の3段階で評価した。

プロダクトオーナーの評価がよかったプロジェクトA、Bでは「普通」と回答したメンバが多かった。プロジェクトCの評価は「不満足」である。しかし、チームメンバ内のコミュニケーションは「よくとれていた」としている。プロジェクトDは、メンバ全員がチーム内のコミュニケーションが「不十分だった」と回答した。プロジェクトEは「普通」と回答するメンバが多かった。この結果からプロジェクト内でコミュニケーションがうまくとれていて、チーム内の関係が良好でもプロジェクトの成果には影響しないことがわかった。

また、プロダクトオーナーとチームのコミュニケーションと成果の関連について考察する。プロダクトオーナーとの関係を「よかった」「普通」「よくなかった」の3段階で評価した。プロダクトオーナーの評価がよかったプロジェクトA、Bであるが、「よかった」と回答するメンバが多かった。また、プロジェクトDも「よかった」という回答が多かった。プロジェクトCはメンバ全員が「よくなかった」と回答した。プロジェクトEは「普通」と回答するメンバが多かった。この結果から、プロダクトオーナーとの関係はプロジェクトの成功・失敗に影響することがわかった。チーム内の関係が良好であっても、プロダクトオーナーとの関係が悪ければ、プロダクトオーナーの求める要求を実現することはできず、プロジェクトは失敗する。そのため、アジャイル型のPBLを実施する場合は、顧客（プロダクトオーナー）との関係が最も重要であった。

また、プロダクトオーナーとの関係が良好であるチームはスクラムマスターが積極的にプロダクトオーナーとチームの橋渡しをする傾向があった。スクラムマスターが間に入り顧客とチーム間の良好な関係を保つことも重要である。

5.3 学生に対するScrum教育の効果

本節では、Scrumの教育効果について考察する。学生に対しては、下記の理由により、通常の社会人に対する教育よりも困難であるといわれている。

- 開発言語やツールの理解が不足している。
- 授業での教育では週に1回しか演習時間がとれない。

しかし、実施してみると結果は良好であった。授業全体では、「Scrumの開発手法を理解できた」という感想はす

すべての学生から得ることができた。概念だけの説明では理解が難しいところを実践することで体感的に学習することが可能となった。Scrum は毎日、同じ場所にチームが集合し開発することが原則である。しかし、Facebook, GitHub 等のツールの利用により、分散的な開発環境でも集合環境と同等の効果があることを確認した。

5.4 スクラムマスター、プロダクトオーナーの教育効果

アジャイル型のプロジェクトの実施により、Scrum のマネジメント方法論を知らないスクラムマスターやプロダクトオーナーにも学習効果がみられた。Scrum を実践した社会人技術者は、『Scrum は初めてであったが、たくさんのお話を学ぶことができた。』と述べている。同様の意見は他のスクラムマスターにもみられた。

プロダクトオーナー側も一方的に仕事を投げるのではなく、チームとの関わり方を考えながら進めていた。2011年度では、顧客からの単一方向のコミュニケーションであったが、2012年度では、双方向のコミュニケーションが取れていた。

6. 今後の課題

2012年度のプロジェクトでは、プロダクトオーナーの評価が良いプロジェクトと悪いプロジェクトに分かれた。良好な評価のプロジェクトを分析した結果、プロダクトオーナー、チーム間のコミュニケーションがよく取れていたことがわかった。プロジェクトを成功させるためには、コミュニケーション基盤の構築が重要である。そのためには、学生だけでなく、社会人参加者への教育も必要である。また、プロダクトオーナーにより、完成した製品の評価基準が異なるため、統一的な評価基準を作る必要がある。また、本授業は、産業界から多くの方が参加し、その役割を果たすことで、教育効果の向上が見込まれるが、多くの大学で実施するためには、持続的な体制づくりが必要である。

7. まとめ

本論文では、アジャイル型ソフトウェア開発 PBL「協創型ソフトウェア開発」における 2012年度のプロジェクトを通して得た知見を報告した。アジャイル型ソフトウェア開発 PBL は、欧米では、積極的にソフトウェア開発教育に取り入れている。しかし、国内では普及していない。我々は、協創的なソフトウェア開発者の育成を目指し、産学連携で対話をしながら PBL 環境を整備してきた。

2011年度の試行では、メンバの技術力の差による個人への負荷と開発遅延、顧客と不平等な関係、メンバ同士でのコミュニケーション不足が問題となった。2012年度は、これら問題を解決するために、開発プロセス「Scrum」を採用した。そして、プロジェクトの結果を分析し、メンバの多様性を活かした問題解決、顧客のプロジェクトへの積極的参加によるチームとの関係改善と双方向コミュニケーション

ョン、スクラムマスターによるチームの自主的に活動できる「場作り」によるコミュニケーション基盤の構築がプロジェクト成功に役立つことがわかった。また、教育効果は学生のみならず、アジャイル型のプロジェクトを実施により、Scrum のマネジメント方法論を知らないスクラムマスターやプロダクトオーナーにも学習効果がみられた。

謝辞 本研究は慶應義塾大学における産学連携プロジェクト「協創型ソフトウェア開発」の中で実施しました。産業技術大学院大学の中鉢欣秀先生、慶應義塾大学の岡田健先生、静岡大学の松澤芳昭先生、アジャイルコーチの皆さま。そして、プロジェクトに参加した学生の皆さまに多大なご協力をいただきここに謝意を表します。

参考文献

- 1) Davenport, D.: Experience Using a Project-Based Approach in an Introductory Programming Course, IEEE Trans. Education, Vol.43, No.4, pp.443-448 (2000)
- 2) 沢田篤史, 小林隆志, 金子伸幸, 中道上, 大久保弘崇, 山本晋一: 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, 情報処理学会論文誌 Vol.50, No.11, pp.2677-2689 (2009)
- 3) 井垣宏 他: 実践的ソフトウェア開発演習支援のためのグループ間比較にもとづくプロセスモニタリング環境, 日本教育工学会論文誌 34(3), pp.289-298, (2010)
- 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教育の試みと成果, 情報処理学会論文誌 Vol.48 No.8 pp.2767-2780 (2007)
- 5) 松澤芳昭, 杉浦学, 大岩元: 産学協同の PBL における顧客と開発者の協創環境の構築と人材育成効果, 情報処理学会論文誌 Vol.49 No.2 pp.944-957 (2008)
- 6) Satoru Kizaki, Cyubachi Yoshihide: Improvement of collaborative work on software development PBL in a distributed environment, Invitation to 3rd International PBL Symposium 2012 (2012)
- 7) Linda Rising, Norman S. Janoff: The Scrum Software Development Process for Small Teams, IEEE Software, pp.26-32 (2000)
- 8) Jeffrey Hammond: Five Ways To Streamline Release Management, Forrester (2011)
- 9) Kent Beck, B. Boehm: Agility through discipline A debate, IEEE Computer, pp.44-46 (2003)
- 10) Hanks, B.: Becoming Agile using Service Learning in the Software Engineering Course, Agile Conference (AGILE) 2007, pp.13-17 (2007)
- 11) Feng Ji: Comparing extreme programming and Waterfall project results: Software Engineering Education and Training (CSEE&T), 2011 24th IEEE-CS Conference on, pp.482-486 (2011)
- 12) H. Takeuchi and I. Nonaka: The new new product development game. Harvard business review, Vol. 64, No. 1, pp. 137-146 (1986)
- 13) Esther Derby, Diana Larsen and Ken Schwaber: Agile Retrospectives: Making Good Teams Great (2006)
- 14) Silva, K.; Doss, C.: The Growth of an Agile Coach Community at a Fortune 200 Company, Agile Conference (AGILE), pp.225-228 (2007)
- 15) 木崎悟, 丸山英通, 土屋陽介, 中鉢欣秀: ソフトウェア開発 PBL へのチケット駆動開発の適用による共同作業の改善, 2011 年度プロジェクトマネジメント学会秋季大会 (2011)
- 16) 木崎悟, 大野貴行, 小松真, 馬場匠見, 室山大輔, 中鉢欣秀, 土屋陽介, 加藤由花: Android を利用したロボット遠隔操作システムの構築, 第 30 回 日本ロボット学会 学術講演会 (2012)
- 17) GitHub, <https://github.com/>
- 18) アジャイルコーチって何するの?, Ryuzee.com, <http://www.ryuzee.com/contents/blog/6220> (2012)