

アジャイル開発手法スクラムを用いた PBL の実施と評価

宇津 説人 高橋 規介 竹澤 雄 山本 祥子 内藤 広志
大阪工業大学 情報科学部 情報メディア学科

1. はじめに

実践的なシステム開発力は、プロジェクトベース学習 (PBL) によってシステム開発を自ら体験することで獲得できる。しかし、PBL の初期フェーズでは、学習者は知識もスキルもなく、与えられた問題に対して適切な目標を設定し、計画を立てることは困難である[1]。そのため、アジャイル開発手法で提案されているように、反復を繰り返すことで知識とスキルを徐々に獲得しながらプロジェクトの目標を次第に明確にし、その目標を満たすシステムを段階的に開発できると考えた。この方法を我々はアジャイル型 PBL と呼んでいる。

アジャイル開発手法には様々な手法があるが[2]、2008 年度はどの手法も採用せず、単に反復を繰り返すことを方針にして学生自身が立案したシステムを開発した[6, 7, 8]。その結果として次の問題点が発生した。

- (1) 現実の場面で問題点を発見することが難しく、システムの目標が曖昧となり、機能の優先度付けや利便性の評価ができなかった。
- (2) 目標達成のための長期的な計画を立てることが難しく、結果として一週間の計画しか作成できず、目先の成果を追いかけてしまい、再計画もできなかった。
- (3) ホワイトボードを使ってプロジェクト管理を行ったが、ホワイトボードが開発環境から離れているために変更が面倒で週末にまとめて変更した。その結果、プロジェクトの進捗が共有できず協調して作業できなかった。

これらの問題点を解決するため、課題を「オーナーが要望する実際に使用する Web アプリケーションを Ruby on Rails[5]を用いて開発する」と設定し、アジャイル開発手法スクラム[3]とプロジェクト管理ツール Redmine[4]を用いて当研究室の 3 年次生をメンバーとする 5 ヶ月間のプロジェクトを実施した。本報告では、アジャイル型 PBL の実施法を述べ、アジャイル型 PBL の問題点と改善法を述べ、その効果と課題について考察する。

2. アジャイル開発手法スクラム

スクラムは、マネジメントを重視してチーム全体の生産性向上を目指すアジャイル開発手法である。スクラムのワークフローは、まず、開発システムに要求される機能や作業とその優先度をプロダクトバックログとして定義する。次に、スプリント計画ミーティングを行い、プロダクトバックログを詳細化して、30 日間の作業期間 (スプリント) で実装すべき目標としてスプリントバックログを定義する。各スプリントでは、15 分間のミーティング (日次スクラム) を毎日全員で行い、スプリントバックログに従って開発を進める。スプリントが終了すると、スプリントレビューミーティングを開催し、作成されたプロダクトを評価する。その評価から次スプリントに備えてプロダクトバックログの内容と優先度の見直しを行う。スクラムでのメンバーの役割は、バックログの定義

Apply Agile Development methodology SCRUM to PBL.

Akito Utsu, Kisuke Takahashi, Yu Takezawa, Shoko Yamamoto, Hiroshi Naito

Department of Media Science, Faculty of Information Science and Technology, Osaka Institute of Technology

や優先度を決定するプロダクトオーナー、プロジェクトの管理者であるスクラムマスター、開発を担当するスクラムメンバーに分けられる。以降、オーナー、マスター、メンバーと省略する。

スクラムでは、プロダクトバックログがプロダクトの最終目標であり、優先順や作業量によって作業を選択してスプリントバックログを作成することで反復の方向性と計画性が与えられる。そのため、スクラムを用いることで、問題点(2)の長期的な目標を設定して反復的に PBL を遂行することが可能と考えた。

3. アジャイル型 PBL の方針

アジャイル型 PBL では、スクラムに従うだけでなく、1 章で述べた問題点を解決するために次の方針を採用した。

- (1) オーナーの要望を満たす実システムを開発する。システム開発力を獲得するには、単にシステムを開発するのではなく、実際の利用を目標としたシステムを開発する必要がある。メンバーはオーナーとなってくれる教職員を自ら探すことから始めた。マスターは当研究室の 4 年次生が担当した。開発チームは 3, 4 名の規模である。
- (2) スプリントを 1 週間単位に区切って反復する。学生は受講授業が異なり毎日全員集まるのは困難なため、日次スクラムの代わりにオーナーを除く全プロジェクトのメンバーが週末に集まり進捗報告会を実施した。そのため、図 1 のように 1 週間の反復 (週次スクラム) を行い、報告会では今週の目標・進捗・問題点・次週の目標を発表し、1 週間で実装した機能をデモした。また、全プロジェクトに共通する問題点や解決案をグループ討議した。
- (3) プロジェクト管理ツール Redmine を用いてプロジェクトを管理する。研究室や自宅からプロジェクト情報を参照変更できるようにプロジェクト管理ツール Redmine を利用した。また、プロジェクト管理情報だけでなく、週間進捗報告書、ソースコード、設計情報などすべての開発情報を Redmine の機能を用いて一元的に管理した。

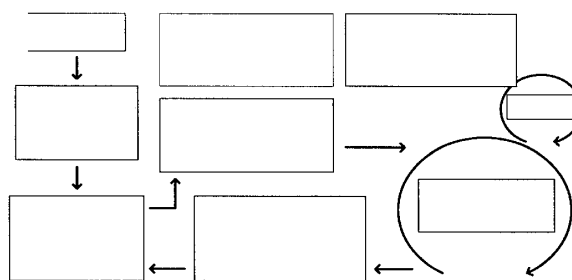


図 1 アジャイル型 PBL のワークフロー

4. Redmine を使用したアジャイル型 PBL の実施法

プロジェクト管理ツール Redmine は、プロジェクト中に発生する機能要求や障害対策などをチケットという形式で管理する課題管理システムである。チケットの分類 (トラッカーと呼ぶ) として、機能、バグ、サポートが予め定義されているが、PBL の問題解決プロセスを明示化するため、MPS の 6 段階方略[1]の各ステップに対応する、

専念、問題確認、探索、計画、実行、振り返りのトラッカーを追加し、プロジェクトの全作業をチケットとして管理することにした。

アジャイル型 PBL では、まずオーナーの要望をまとめ、プロダクトバックログを表形式で作成する。各行には、ID、要望名、優先度、予定工数、説明、開発スプリント番号を記述する。次に、プロダクトバックログの各要望を細分化して機能・作業を洗い出し、スプリントバックログを表形式で作成する。各行には、プロダクトバックログの情報に加え、対応するプロダクトバックログの ID と該当チケットを記述する。両バックログは Redmine の Wiki 機能のテーブルを用いて記述した。次に、スプリントバックログの機能に対応してチケットを作成し、タイトル、説明、優先度を記述する。担当者が決まり作業を開始するとき、チケットの担当者として予定工数を記述する。週次スクラムは、Redmine のバージョン機能を使って表現し、チケットを作成するとき週次スクラムを選択する。その結果、ロードマップページに週次スクラム毎のチケット一覧と進捗率が表示され、各週の作業内容と進捗状況がわかる。

表 1 スプリントの実施状況

番号	期間	目標	チケット数	見積り時間(h)	実績時間(h)	進捗率
1	9/17~10/1	オーナーを決定し、プロダクトバックログ、スプリントバックログを作成する	8	16.5	17	70%
2	10/1~11/5	事務員の予定を新規作成、編集でき、その予定を一覧できる機能を作成する	6	55	56.5	60%
3	11/5~12/3	ログイン機能、スケジュールの再利用機能を作成し、ツールの利便性を高める	4	33	51	40%
4	12/3~1/7	管理者ページを作成し、制限された範囲でしかアカウントを作成・編集・削除出来ないようにする	7	56	76	80%

5. プロジェクトの実施例

2009 年度は 4 つのプロジェクトを実施し、8 月は Ruby と Rails の学習などの準備をし、9 月よりスプリントを 4 回繰り返した。その成果を 11 月の学園祭と 1 月のスプリントレビューミーティングで報告した。表 1 は、事務員がオーナーの「スケジュール管理アプリケーション」プロジェクトの実施状況である。スプリント 1 と 2 では経験が乏しいため、オーナーの要望をプロダクトバックログにまとめるのが難しく、要望をただ羅列したものであった。機能間の関連性も明確でなく必要な機能も欠けており、重要さを考慮して優先度を付けていなかった。そのため、利用者の役割を明確にし、プロダクトを介して利用者間でどのようにやり取りするかを記述した利用シナリオを作成した。また、プロダクトの特徴や効果を考え、リリース計画書[7]を作成することで必要な機能や優先度が明確になった。その結果、必要な機能を網羅したプロダクトバックログを作成できた。しかし、見積り時間が 20 時間を越えるチケットもあり、各メンバーで分担して作業するのに適する大きさまで機能を細分化することができなかった。そのため、KPT[6]によって細分化の問題点や方法をグループ討議した。作業の見積りは最後まで難しい課題であった。スプリント 4 になると、開発すべき機能が複雑となり作業に必要な時間が増え、見積り時間と実績時間の差が返って増加した。

6. 評価と考察

スプリント 4 の終了後、プロジェクト管理に関するスキルの 5 段階の自己評価をメンバーに実施した。その結

果が表 2 である。目標の評価が高い。最初のスプリントでは目標や優先度を正確に立てることができなかったが、スプリントを繰り返して行っていくなかで、利用シナリオやリリース計画書を作成・修正をすることで目標や優先度が徐々に明確になっていった。その結果、作業が具体的にになりチケットを細分化できるようになった。その反面、品質の評価が低い。プロダクトの品質目標を設定しておらず、バグのチケットの記述法を決めていなかったため、バグが発生してもチケットを作成しなかった。

表 2 プロジェクト管理スキルの自己評価

プロジェクト管理		平均	合計平均
目標	①スクラムの作業フローとメンバーの役割を説明できるか	3.64	3.40
	②オーナーの要望とプロダクトの効果を考慮してプロジェクトの目標を作成できたか	3.64	
	③利用シナリオと優先順位を考慮してプロダクトバックログを作成できたか	3.45	
	④優先順位と見積り時間を考慮してスプリントバックログを作成できたか	3.18	
	⑤優先順位と個人の負荷を考慮して1週間のイテレーション計画を作成できたか	3.08	
品質	⑥プロダクトの品質目標を設定したか	2.36	2.87
	⑦問題点(バグ)が発生したときチケットに記録したか	2.82	
	⑧バグの解決策をチケットに記録したか	2.64	
	⑨開発者以外での他チームの報告によって自分のバグを解決できたか	2.91	
	⑩開発者以外でバグを報告できるようなチケットを作成できたか	3.64	
時間	⑪見積り時間が18時間以内になるようにチケットを分解できたか	3.91	3.40
	⑫チケットの見積り時間を作成したか	4.09	
	⑬作業の進捗が終了したチケットの実績時間を記録したか	3.64	
	⑭見積りと実績の差が大きかった場合その原因を調査したか	2.33	
	⑮見積りの精度を向上させるために見積り法の改善をしたか	2.64	
	⑯Redmineを使って進捗管理を行う方法を説明できるか	3.91	
分担	⑰メンバーの能力や作業可能時間を考慮しスプリント計画を作成したか	3.27	3.48
	⑱チーム全体の進捗を常に把握しメンバーの遅れがないかを問いただしたか	2.91	
	⑲進捗が遅れていた場合遅れを取り戻すための行動をとったか	3.82	

7. おわりに

4 つのプロジェクトとも 5 ヶ月間でオーナーが要望する基本機能を実現できた。1 月のレビューミーティングでは「4 年生に頼らずに自分で調べた」、「オーナーに対する責任感を感じた必死にやった」などの感想が述べられ、自己主導型学習の意識が芽生え、アジャイル型 PBL は効果があったと評価できる。

しかし、プロジェクト管理の基本スキルは習得できたが、Rails、データベース設計、オブジェクト指向設計の理解は不十分だった。実用的なプロダクトを開発するには、更に、効率性、信頼性、保守性、セキュリティを高めるための技術を習得する必要がある。自己主導型学習・相互学習によって、プロジェクト管理のスキルをより高めるとともに、これらの IT 技術の習得を支援する方法が今後の課題である。

参考文献

- 1) ウッズ, ドナルド R.: Problem-based Learning 判断能力を高める主体的学習, 医学書院, 2001.
- 2) ラーマン, クレーグ: 初めてのアジャイル開発, 日経 BP 社, 2004.
- 3) Schwaber, Ken: スクラム入門, 日経 BP ソフトプレス, 2004.
- 4) 前田剛: 入門 Redmine, 秀和システム, 2008.
- 5) Thomas, Dave ら: Rails によるアジャイル Web アプリケーション第 2 版, オーム社, 2007.
- 6) 橋本唯, 荒木光, 橋本彩香, 内藤広志: プロジェクトベース学習のためのファシリテーション技術の教材, 情報処理学会第 71 回全国大会 5ZB-5, 2009.
- 7) 藤原僚太, 東田恵治, 三上翔平, 米田典弘, 内藤広志: プロジェクトベース学習の反復型問題解決プロセスの振り返りと改善案, 情報処理学会第 71 回全国大会 5ZB-6, 2009.
- 8) 足立貴大, 山田拓人, 服部淳, 内藤広志: プロジェクトベース学習(PBL)を支援するプロジェクト管理ツール, 情報処理学会第 71 回全国大会 5ZB-7, 2009.