

PBL 型教育におけるアジャイル人材育成のプラクティス：2024 年度の事例

Practice of agile development engineers in PBL: A case in 2024

中鉢 欣秀^{1*} 小澤 裕太¹ 木村 和郎¹ 高嶋 隆一¹ 塚本 耕司¹ 益田 匡史¹ 若松 隼也¹
Yoshihide Chubachi^{1*} Yuta Ozawa¹ Kazuro Kimura¹ Ryuichi Takashima¹ Koji Tsukamoto¹ Tadashi Masuda¹ Shunya Wakamatsu¹

¹ 東京都立産業技術大学院大学 Advanced Institute of Industrial Technology

*Corresponding author: Yoshihide Chubachi, yc@aait.ac.jp

Abstract Agile development has been widely adopted as a software development method other than the traditional waterfall method. Scrum is becoming established as an agile development method and is recognized as a standard process. The main theme of this PBL is the acquisition of agile development techniques. 2024 is to improve agile development methods by integrating Scrum-type agile development and UML modeling. We will examine further utilization of UML for visualization of architecture and as a communication tool with stakeholders.

Keywords agile development; scrum; uml Modeling

1 はじめに

ソフトウェア開発の手法として、従来型のウォーターフォール型に加え、アジャイル開発が広く採用されるようになってきた。アジャイル開発の手法としてはスクラム [1] が定着しつつあり、標準的なプロセスとして認知されつつある。

筆者（中鉢）が担当する PBL では、アジャイル開発技術の習得を大きなテーマとして設定している [2,3]。この PBL の過去事例は文献 [4-6] に発表した。

2024 年度は、スクラム型のアジャイル開発と UML のモデリングを融合させることで、アジャイル開発手法の改善を行うことをテーマに設定した。UML によりアーキテクチャを可視化したり、ステークホルダーとのコミュニケーション手段としての更なる活用を検討する。本論文は 2024 年度前期の成果発表会で学生が作成した資料をもとに、この活動の成果を述べる。

2 PBL のテーマと目的

2024 前期の PBL における全体テーマ

2024 年度前期の PBL のテーマとして、本学の PBL の全体テーマとしてアジャイル開発と UML のモデリングを併用することによりスクラムにおける課題解決を設定した。2024 年度前期の PBL のテーマを図 1 に示す。

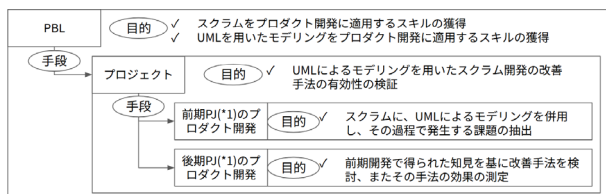


図 1 2024 年度前期の PBL の目的

スクラム開発手法の改善を要する点

アジャイル開発手法であるスクラムにおいて、プロダクトオーナーの要求を獲得するための方法としてプロダクトバックログが用いられる。スクラムガイド [1] によれば、プロダクトバックログは「創発的かつ順番に並べられた、プロダクトの改善に必要なものの一覧である。これは、スクラムチームが行う作業の唯一の情報源である」とされる。

スクラムガイドにおいてはプロダクトバックログの具体的な作成手法については言及されていない。「プロダクトオーナーは、効果的なプロダクトバックログ管理にも責任を持つ」とされているものの、プロダクトオーナーの要求をどのように獲得し、プロダクトバックログに反映させる方法についての具体的な説明はない。

また、プロダクトバックログは、プロダクトバックログアイテム (PBI) の優先順位付きリストである。プロダクトバックログアイテムにはユーザーストーリーを記述することが多い。ユーザーストーリーはプロダクトを利用する際の流れだと解釈できる。プロダクトの全体的なユーザーストーリーがあり、これを分割したものが PBI である。

ここで、PBI は優先度順に並べ替えるというルールがある。その結果、プロダクトバックログリストを開発に用いる際の制約として、開発期間のプロダクトバックログの順序関係は、プロダクト全体のユーザーストーリーを反映していないことが発生する。

加えて、PBI の内容は自然言語を用いて記述するため、書き手が十分に注意しない限り、記述に曖昧性が発生する可能性もある。

以上を踏まえ、スクラムガイドに則ってプロダクトバックログを利用する場合にはある種の制約が発生しているとし、そこから派生する課題と、より具体的な問題を分析した (図 2)。

これにより、スクラムを実施する際の問題として次のものが抽出された。

1. PBI 同士の関連が分かりづらく、重複作業や依存の見落としが生じる
2. スプリントレビューで全体のどこを実装しているのか分かり辛い
3. プロダクトオーナーの意思と齟齬がある PBI ができあがる

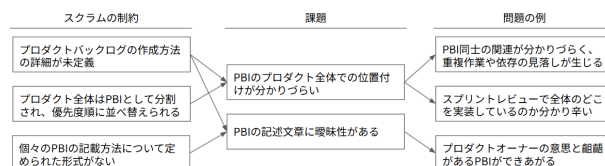


図 2 スクラムの制約と課題及び問題の例

本プロジェクトでは、これらのプロダクトバックログに関連する問題を UML によるモデリングを併用することで解決することを検討する。加えて、プロダクトオーナーの要求をより明確にし、

開発チームとのコミュニケーションを円滑にする方法についても探る。

取り組む課題（RQ）

PBLで取り組む課題として、「プロダクトバックログとUMLによるモデリングを併用する事で前節で述べた課題を解決できないか（RQ）」という仮説を立て、その検証を実施する。

PBIの取り扱い方法に起因する課題は、

1. PBIのプロダクト全体での位置付けの確認の難しさ
2. PBIの記述の曖昧さ

による。

一方、オブジェクト指向モデリングで用いるUMLはシステムの設計や仕様を視覚的に表現するための標準化された形式言語である。UMLは表記法が統一されているため、その表現により受け取り手側に解釈を任される部分が少ない。このような特性を備えるUMLと、自然言語により作成するPBIを組み合わせることによってどのような効果が得られるかを確かめる。

3 アジャイル開発とUMLを併用する提案手法

提案手法

設定したRQを検証するために、実際にUMLによるモデリングを併用したソフトウェア開発を実施する。開発のプロセスは図3に示すとおり、大まかに項番A、B、Cの3ステップからなる。項番AはUMLモデリングを通した要求分析の工程である。項番BがUMLを用いた分析結果からプロダクトバックログを作成する工程となる。項番Cでは前工程で作成したプロダクトバックログを開始点としてスクラムによる開発を行う工程である。

以上の開発プロセスを一通り実施することで、本提案手法の有効性を検証すると共に、新たに課題を抽出することを狙う。

| 項番 | 適用時期 | 改善手法 | 活用イメージ |
|----|---------------|---|--|
| A | 要求獲得・要件定義フェーズ | UMLモデリング(ユースケース図、アクティビティ図、クラス図)を元にした要求獲得、要件定義 | ・業務のAsIs、ToBeを定義したUMLモデリング ・クラス図を用いた情報に関する用語の統一 |
| B | | UMLモデリングを用いてユースケース図からプロダクトバックログを生成 | ・ユースケース図からのPBIの作成 ・アクティビティ図によるPBI記述の視覚化 |
| C | 開発フェーズ | ユースケース図、アクティビティ図を用いたプロダクトオーナー、開発チームでの情報同期 | ・スプリントレビュー時のPBIから紐づくユースケース図、アクティビティ図を用いた自検合わせ |

図3 アジャイルとUMLを併用した開発プロセス

先行研究

特定非営利活動法人UMLモデリング推進協議会（以下、UMTP）は「アジャイルソフトウェア開発向けUML適用ガイドライン Ver1.2[7]」を公開している。このガイドラインはUMLの経験者がアジャイル開発を始める際に参考になるように書かれている。p.9に「初期のプロダクトバックログの作成」について述べられており、PBIの作成にはユースケース図が対応することが記載されている。

本提案手法もユースケース分析の結果を用いて初期のPBIを作成することを基本的な方針とするが、同ガイドラインにはユースケース分析の結果からPBIを具体的にどのように作成し、その後のスクラム開発に活用していくのかについての具体的な説明は見当たらない。

サントスら[8]は産学連携R&Dプロジェクトにおいてスクラム開発にUMLによるモデリングを組み込むことにより得た筆者らの教訓を紹介している。アジャイル開発は産業界の実務家にとっては一般的に使用されているものの、R&Dを共同で実施する研究機関にとっては馴染みがないため、両者のニーズを満たす手段としてUMLとアジャイル開発の併用は有効であるとしている。

対して、本研究では単なる異なる立場の橋渡しのためにUMLを利用するのではなく、アジャイル開発とUMLを併用することで開発手法そのものの改善を探ることを目的としている。

アジャイル開発とUMLを併用した開発プロセスを進めるにあたり、図3の項番A,Bにあたる要求獲得・要件定義を行う手法としてRDRA[9,10]を採用した。RDRAはモデルベースのビジネスとシステムの可視化手法であり、ToBeモデルとAsIsモデルの両方に対応している。手法が体系的に整理されており、スムーズに実施できると判断した。

開発するプロダクト

今回の開発では教員がプロダクトオーナーとなり、学生がスクラムチームを構成し、交代でスクラムマスターを担当する。作成するプロダクトは教員のレポート採点業務を支援するためのサービス（manakan）とした。

図4にこのプロダクトのエレベーターピッチを示す。また、このプロダクトにより採点業務がどのように変化するかを図5に示す。

[レポート採点の負担を軽減]したい
[中鉢先生向け]の
[manakan]というプロダクトは、
[レポート採点支援ツール]です。
これは、[レポートの評価をグラフィカルにグルーピングを行うこと]ができ、
[既存の記憶に頼った手動での評価]とは違って、
[学生が提出したレポートを相対的に評価してグラフィカルにグルーピングを行い、個人への点数付け調整を行う機能]が備わっている。

図4 開発するプロダクトのエレベーターピッチ

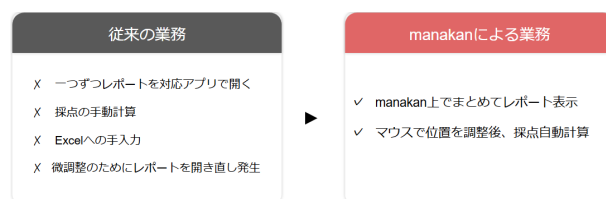


図5 プロダクトの導入前後における業務の変化

本開発ではRDRAに基づき、図6に示すプロセスに従い初期のPBIの作成まで実施する。図中、「情報モデル」はRDRAの用語であり、一般的なUMLモデリングでのクラス図に相当する。

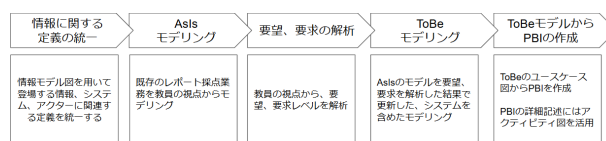


図6 RDRAを用いたプロセス

4 手法の適用結果

情報モデル図

情報モデル図は情報（用語）とそれらの関係を示した図である（図 7）。後段のプロダクトバックログの作成においては必ずこのモデルで定義した用語を使用することとした。

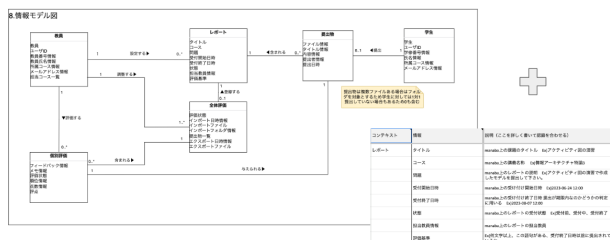


図7 情報モデル

AsIs モデル

AsIs モデルは既存のレポート採点業務を教員の視点からモデリングしたものである(図 8). このモデルはプロダクトオーナーに対するヒアリングに基づき作成した。

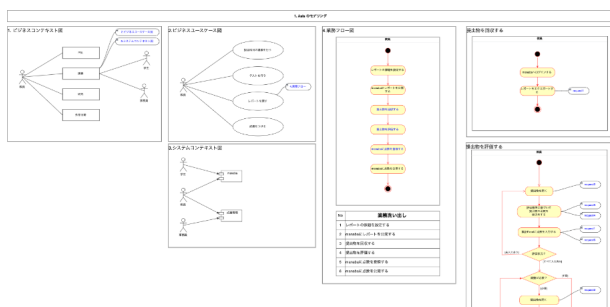


図 8 Asls モデル

要望・要求の解析

PO からの要望・要求をユースケース分析したものを図 9 に示す。この分析により、ユーザーが欲しているものを明確化している。

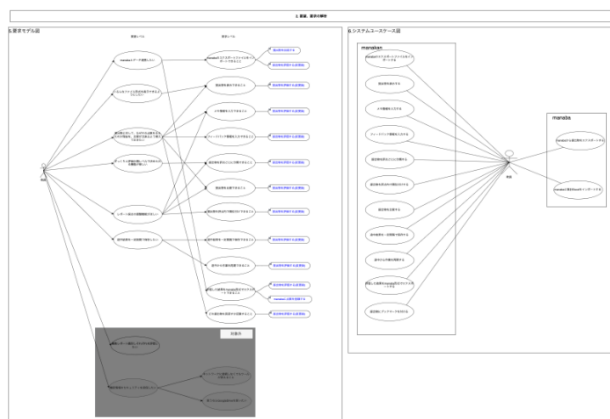


図9 要望・要求の解析

ToBe モデル

AsIs のモデルに対して要望・要求を解析した結果を反映し、ToBe モデルを作成した (図 10)。作成した内容はプロダクトオーナーとの合意を得て、初期の PBI に落とし込む。

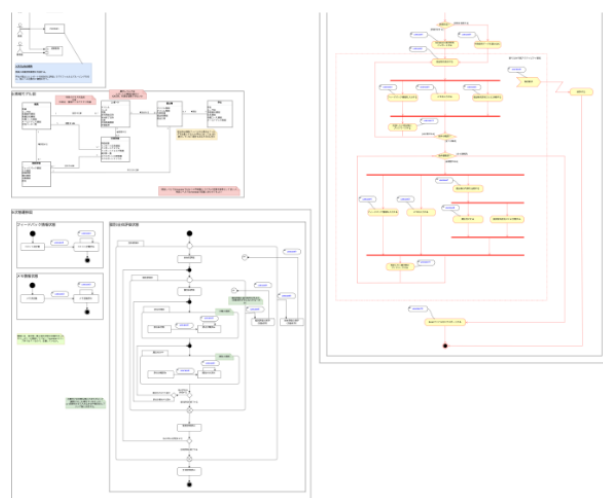


図 10 ToBe モデル

ToBe モデルから PBI の作成

各ユースケースに示される機能や要求事項をプロダクトバックログアイテムとして、プロダクトバックログリストの形式に変換した(図11).



図 11 Tobe モデルから初期の PBI の作成

5 提案手法の有用性の考察と課題

有用性評価の観点

本手法は上流工程における要求獲得・要件定義のみならず、その後のアジャイル開発においても継続して UML を活用することを目的としている。したがって、本手法の有用性を検証するためには、前節までで作成したモデルをその後のスクラム開発で実施するスプリントの中でどのように活用できたかを評価する必要がある。

また、アジャイル開発においてプロダクトバックログは常にリファインメント（洗練）される。その際には内容の追加・変更・削除や優先順位の変更等が発生する。このような変更に対してモデルの内容も変更する可能性があり、このような場合に対応するための工数も加味する必要がある。

スプリントレビューにおける効果

スプリントの終了時に毎回実施するスプリントレビューにおいては、スプリントで何が達成されたかをレビューする [1]。レビュー対象には「インクリメント」が含まれ、これはスプリントにおいて完成した成果物の増分である。

従来のスクラムにおいては、インクリメントのレビューにおいては、プロダクトバックログに基づいて行う。プロダクトバックログは開発する内容の優先度付きリストであるから、優先順位の高い順番でソートされている。スクラムチームは、このリストの上から順番にインクリメントを説明する。この際、プロダクトのインクリメント部分のデモを行うことが多い。

プロダクトバックログの例を図 12 に示す。リストの各アイテムは優先順に並べられており、また各アイテムは全体のユーザーストーリーが細切れになっている。従って、このリストを上から順番に閲覧しても、全体としてのユーザーストーリーを読み取ることが困難である。

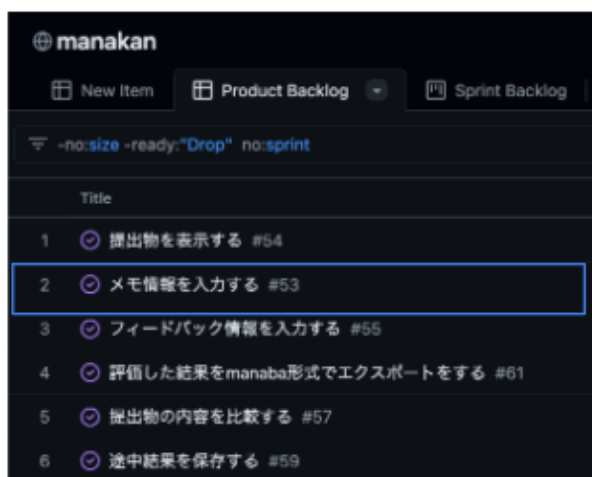


図 12 プロダクトバックログの例

プロダクトオーナーは、プロダクト全体が提供するユーザーストーリーのうち、一部のデモンストレーションを見て PBI が完了したかどうかの判断をしなくてはならない。

これに対して、本手法ではプロダクトバックログに加えて UML で作成したモデルも参照することができる。図 13 は同じプロダクトのアクティビティ図である。

プロダクトオーナーはこのアクティビティ図により示されるプロダクトの全体像を俯瞰しながら、インクリメントのレビューができるようになる。

提案手法の課題

今回の開発を通して、本手法の課題も浮かび上がってきた。

最初の課題は、モデリングを行うフェーズから開発を行うフェーズへ移行するタイミングである。本手法では、もともと、上流工程における UML を用いたモデリングとその後のアジャイル開発フェーズがシームレスにつながることを想定していた。しかしながら、初期のモデルを作成する段階でどこまでやればモデリングが完了し、アジャイル開発に入れるのかという見極めができなかった。この様子を図 14 に示す。

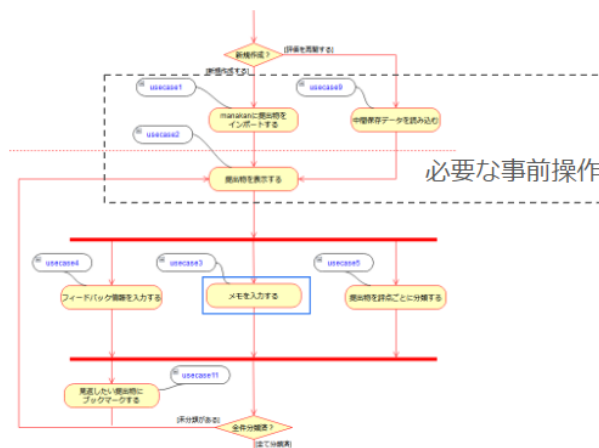


図 13 レビューに用いるアクティビティ図

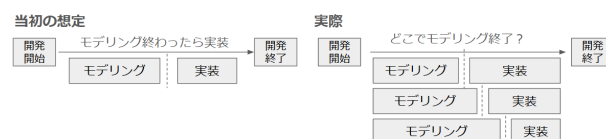


図 14 いつまでモデリングするべきか

これはどこまで精緻にモデリングをすれば実装を始めて良いのかという判断基準が準備できなかったことによる。モデリング作業の量が増えれば、迅速に開発を行うことができるスクラムの利点を損ねることは理解できているものの、実際の判断は難しい。

本プロジェクトのメンバーは、判断に悩んだポイントとして次のものを指摘した。

- クラス図で操作や型、可視性を考慮し、具体的にしたほうが良いか
- 業務ドメイン以外の、アプリケーションや UI のクラスもモデリングするのか
- シーケンス図は必要か、不要か
- 状態遷移図は必要か、不要か

大まかに言えば、実装レベルのモデリングが必要かどうかという点であろう。アジャイル開発においては、実装レベルのモデリングの価値はあまり高くはないと考えられる一方、アジャイル開発においてもプロダクト内部のアーキテクチャ設計をすべきかどうか、という点に関しては更なる検討が必要だ。

モデルに変更を反映する手間が大きい

スクラムのスプリントにおいてモデリングを併用すると、プロダクトバックログの変更に伴い、モデルも修正しなくてはならない。例えば、PBI を追加した場合、ユースケース図の更新も必要になる。

本研究では、図 15 に示す通り、モデルを更新する作業は各スプリントの最初において実施した。現状の方式では、このモデルの修正作業が全くの追加作業として発生している。

このような追加の作業が発生してもモデリングを併用することにそれ以上の効果があるかどうか、今後検証することになる。

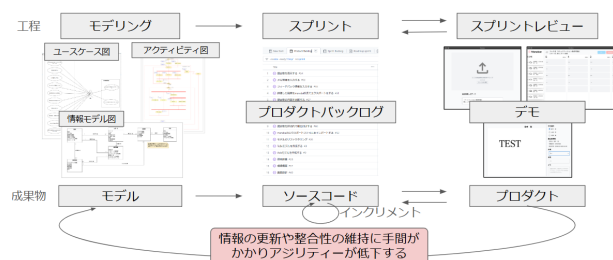


図 15 スクラムのスプリントとモデリング

6 おわりに

今回の PBL の成果により、現状のスクラムにおけるプロダクトバックログの扱いに起因する課題に対し、UML を併用することにより部分的に解決できることが示された。

特にスプリントレビューにおいてプロダクトオーナーはプロダクト全体のストーリーを俯瞰しながら、インクリメントのレビューを行うことができる。このことにより、開発の進捗状況もダイアグラム上で視覚的に判断できる。

一方、モデリングを併用することによる工数の増大は課題である。この点については今後も課題の解決に向けて研究を進めたい。

参考文献

1. Schwaber K, Sutherland J. スクラムガイドスクラム公式ガイド：ゲームのルール. 2020. Available: <https://scrummaster.jp/scrum-guide>
2. 中鉢, アジャイル技術者育成のための開発方法論教育. 教育システム情報学会 (JSISE2016) 第 41 回全国大会. 2016. pp. 313–313.
3. 中鉢, AIITにおける実践的 Scrum 技術者教育の取り組み. 日本ソフトウェア科学会第 33 回大会 (2016 年度) 講演論文集. 2016. pp. 1–6.
4. 中鉢, 間, 嵩下, 星野, 森, 保田. PBL 型教育におけるアジャイル人材育成のプラクティス～2021 年度の事例～. 東京都立産業技術大学院大学紀要. 2021;15: 139–142.
5. 中鉢, 天野, 網干, 斎藤, 澤村. アジャイル開発人材育成のための PBL 型教育における プラクティス : 2022 年度プロジェクトの事例. 日本ソフトウェア科学会第 39 回大会 (2022 年度) 講演論文集. 2022; 1–4.
6. 中鉢, 天野, 網干, 斎藤, 澤村. PBL 型教育におけるアジャイル人材育成のプラクティス～2022 年度の事例～. 東京都立産業技術大学院大学紀要. 2023;16: 23–26.
7. 特定非営利活動法人 UML モデリング推進協議会. アジャイルソフトウェア開発向け UML 適用ガイドライン Ver1.2. 2019. Available: <https://umtp-japan.org/download/%E3%82%A2%E3%82%B8%E3%83%A3%E3%82%A4%E3%83%AB%E3%82%BD%E3%83%95%E3%83%88%E3%82%A6%E3%82%A7%E3%82%A2%E9%96%8B%E7%99%BA%E5%90%91%E3%81%91%E3%82%AC%E3%82%A4%E9%96%8D%E7%9C%E9%81%A9%E7%94%A8%E3%82%A4>
8. Santos N, Fernandes J, Carvalho M, Silva P, Fernandes F, Rebelo M, et al. Using scrum together with UML models: A collaborative university-industry r&d software project. 2016. pp. 480–495. doi:[10.1007/978-3-319-42089-9_34](https://doi.org/10.1007/978-3-319-42089-9_34)
9. 善司. RDRA2.0 ハンドブック. NextPublishing Authors Press; 2019.
10. 善司. モデルベース要件定義テクニック. 秀和システム; 2013.



Open Access This article is licensed under CC BY 4.0. To view a copy of this license, visit <https://creativecommons.org/licenses/by/4.0/>