

アジャイル人材育成プログラムの専門職大学院修士課程への導入

Introducing an Agile Software Development Engineers Education Program to the Master Course Curriculum in AIIT

中鉢 欣秀^{*1}, 小山 裕司^{*1}

Yoshihide CHUBACHI^{*1}, Hiroshi KOYAMA¹

^{*1}産業技術大学院大学

^{*1}Advanced Institute of Industrial Technology

Email: yc@aiit.ac.jp

あらまし：2013 年度から開始された enPiT において、産業技術大学院大学ではアジャイル開発人材育成のための教育プログラムを開発した。アジャイル開発で必要となる能力を、マインドセットとスキルセットの観点から学習する体系的な教育パッケージとなっている。本学では、2017 年度より修士課程の授業科目にこれを導入し、継続的に実施している。本発表ではこの取り組みについて述べる。

キーワード：アジャイル開発、アジャイル開発人材育成、PBL、enPiT

1. はじめに

ソフトウェア開発方法論としてのアジャイル開発は、近年産業界への普及が著しく、この新しい方法論をマスターした人材（アジャイル開発人材）が求められている。大学においても、ソフトウェア開発技術者教育の一貫としてアジャイル開発人材の育成が急務である。

2013 年度より開始した「分野・地域を越えた実践的教育協働ネットワーク（通称：enPiT エンピット）」では全国の 15 大学が中心となり、修士課程在籍学生を主たる対象として実践的 IT 人材の育成プログラムを実施した。また、2017 からは主に学部 3~4 年生を対象にしたプログラムとしてこれを継続している（enPiT2）。

産業技術大学院大学（以下、AIIT）では enPiT 開始当初よりアジャイル開発に注目し、大学における PBL 型教育においてアジャイル開発人材育成のための教育手法について検討し、実施してきた⁽¹⁾。毎年、教育内容の改善を行い、よりよい教育パッケージを目指して開発してきた。

本発表では、AIIT においてこの 5 年余り開発に取り組んできた、大学におけるアジャイル開発人材育成のための教育パッケージについて紹介し、修士課程の授業科目として継続的に実施していることについて述べる。

2. アジャイル開発人材育成の教育

従来のウォーターフォール型開発では、上流工程から下流工程へ至る定められたプロセスが定義されている。これに対して、アジャイル開発には、そのような定式化されたプロセスはない。アジャイル開発の一手法（フレームワーク）であるスクラムでは、役割やイベントなどの用語についてある程度定義されているものの、実際にどのように実施すればよいかについては言及されていない。

アジャイル開発には、開発チーム自らがより良い

手法を考えて振る舞い（自己組織化）、常に課題とその解決策を発見し（継続的な改善）、互いを尊重して協調する（心理的安全）といった概念が組み入れられている。これらを通して、チームとしての生産性の向上を図り、より上手にソフトウェア開発ができるようになることが、アジャイル開発人材には求められる。

これらの能力は、単に教科書を学習しただけで得られるものではない。実際に開発プロジェクトを実施し、その場において学習者が得る経験によって深く身につくものである。

アジャイル開発人材が実践的な能力を獲得するには、このようなアジャイルの概念を学んだ上で、実際にアジャイルに振る舞えるようになる必要がある。そのための教育方法として、従来型の講義や、シナリオのある演習では難しい。

そこで、AIIT はアジャイル開発を実施する PBL（Project Based Learning）と、その準備としての事前学習からなるアジャイル開発人材育成パッケージを開発した。このパッケージでは、アジャイル開発で学ぶべき内容を「マインドセット」と「スキルセット」に大別し、それらを体系的に学ぶことのできる内容となっている。

以下、これらの概要を述べる

2.1 アジャイル開発マインドセットの学習

アジャイル開発では、チームの生産性を向上させるために、常にチーム自体を改善する。改善は一度限りで終わるものではなく、繰り返し継続的に行う。このためには定期的かつ頻繁な振り返りの機会を設け、改善点を見出し、改善策を考えられるようにならない。

そこで、学習者に対しては振り返りの方法と、振り返りの結果から実際にチームの活動を改善する一連の流れを経験させ、改善を実施するためのマインドセットが身に着くようにする。

これを実施するには、短いタイムボックス（5 分

～1 時間) を設け、タイムボックスの間で何らかのチーム作業を行い、タイムボックスが経過したら振り返りを実施して、次のタイムボックスでのチーム作業を行うというワークショップが効果的である。

このようなワークショップの例として、「紙飛行機を作成する」ワークショップがある⁽²⁾。これは、課せられた制約(ルール)のもとで、チームメンバーが協力してできるだけ多くの、よく飛ぶ紙飛行機を作成するというものである。

紙飛行機を作成するタイムボックスは任意であるが、例えば5分で当初2～3機しか成功しなかったチームが、繰り返し(5回程度)改善を行うことで20機以上も成功できるようになる、といったことが発生する。このようなことを実際に体験することで、振り返りの意義や、具体的な改善方法について体得できる。

また、振り返りの具体的なやり方として「KPT(Keep Problem Try)」を紹介し、演習において実際に実施してもらう。これも複数回繰り返すことによって、課題を発見し、解決に導くための能力の向上を図る。

アジャイル開発はソフトウェア開発の方法論ではあるものの、人が構成するチームが協調し、チームとしてより上手にものづくりができるようになることが本質である。これには、単に技術的知識を知るのではなく、ある種の「心構え」を身につけることが重要だ。そのために、ワークショップを中心にアジャイル開発のマインドセットを学ぶ科目として構成している。

2.2 アジャイル開発のスキルセット

アジャイル開発において重要なマインドセットを学ぶと共に、実際にチームで円滑に開発作業が実施できるようになるためのスキルセット(技術)についても学習する必要がある。

ソフトウェア開発のためのプログラミング言語の習得はもちろん必要であるが、一方で、チーム開発を円滑にするためのツールについても学ぶ必要がある。複数人からなるチームがソースコードを共同で編集する場合、編集内容のマージや、同一箇所を変更した場合のコンフリクトの解消などが必要となる。

そこで、近年、ソースコードのバージョン管理システムとしてデファクトスタンダードとなっているGitと、クラウド型GitサーバであるGitHubを用いたコラボレイティブなソフトウェア開発のためのスキルを身につけておくと、チーム開発が円滑になる。

これらを身につけるため、チームで実際に小規模なソフトウェア(簡単な静的Webページでよい)を作成する演習を用意した。チーム開発は一人で学習することはできないため、このような場を提供することで、チーム開発におけるスキルセットを身につけることができる。

2.3 アジャイル開発 PBL

以上のマインドセット及びスキルセットの学習を踏まえ、PBLを実施する。PBLはチームごとに開発するアプリケーションを企画し、スクラムによって実際に実装する。

このPBLで最も重要なのは、定期的なレビューと振り返りである。レビューでは顧客の視点で、開発したプロダクトについてコメントする。また、振り返りの結果としての課題や改善策についてクラスルームで共有する。

これを繰り返すことで、チーム自体が改善され、より良い成果物が生まれるようになる。

3. 修士課程科目としての導入

ここまで述べたenPiTで開発した教育プログラムを、AIIITでは2017年度より修士課程の授業科目として設置している。PBLに関しては、本学は開学以来修士課程2年生の必修科目として実施しているので、PBLの事前学習科目として以下の3科目を用意した。

「アジャイル開発特論」では、アジャイル開発のマインドセットについて取り扱う。講義と演習(ワークショップ)を通して、アジャイル開発のチームメンバーとしてどのように振る舞えばよいかを学ぶ。

「コラボレイティブ開発特論」では、チームによる共同作業を実施するためにGit/GitHubの使い方を学び、クラウド環境で実行できる簡単なWebアプリケーションをチームで開発する。

「フレームワーク開発特論」では、本格的なWebアプリケーション開発に利用できるアプリケーションフレームワークについて取り扱う。Ruby on Railsを題材に、実用的なフレームワーク開発について深く学習する。

4. おわりに

本学において2013年度より開発してきたenPiTの教育プログラムは2017年度より修士課程の科目として導入し、アジャイル開発技術者育成の取り組みを勧めている。

enPiTのような期限のある教育プロジェクトは、予算措置が終了した後に同様の教育を継続することが難しい。本学では、これを修士課程の授業科目として設置し、継続的に実施できるようにした。今後とも、更により教育プログラムとなるよう改善をしていく。

参考文献

- (1) <http://enpit.aiit.ac.jp/>
- (2) <https://agilefaq.wordpress.com/activities-and-games-for-learning-agile/paper-plane-game/>
- (3) <https://github.com/>