

アジャイル技術者育成のための開発方法論教育

Development Methodology Education for Agile Engineer Training

中鉢 欣秀^{*1}

Yoshihide CHUBACHI^{*1}

^{*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:アジャイル開発が注目される中で、それに必要な能力を備えたエンジニアの育成が急務である。アジャイル開発として代表的なスクラムにおいては短い期間（スプリント）でプロダクトをユーザにリリースする。このような迅速性を達成するために、最先端の仮想環境やクラウド環境を効果的に活用した開発環境に適応する必要がある。本研究では enPiT の一環として本学で実施しているアジャイル開発人材育成のための学習環境について発表する。

キーワード: アジャイル開発, アジャイル技術者育成, PBL, Scrum

1. はじめに

ソフトウェア開発の新しい方法論としてアジャイル開発が産業界に普及しつつある。従来手法であるウォーターフォール型の開発は、企業で利用する大規模な基幹系システム開発の方法としては未だに有効な側面はあるだろう。しかしながら、ユーザに対して新しい価値を提供する新規プロダクトの開発においては、アジャイル開発が適しているとの認識が形成され始めており、Web 開発のみならず IoT 分野での利用すら検討されている⁽¹⁾。

このような状況の中で、アジャイル開発に対応するスキルセットを備えた技術者の育成は急務といえる。そのためにはまず、「方法論」としてのアジャイル開発を効果的に学ぶ学習の場を提供する必要がある。加えて、アジャイル開発を円滑に進めるために有用な各種の「道具」の使い方についても学べるようにすることが大切だ。

ここでは、アジャイル開発の方法論と、開発に有用とされる各種のクラウド型サービスを活用したソフトウェア開発を学ぶ教育プログラムについて述べる。

2. アジャイル開発方法論を支援する道具

Kent Beck らが 2001 年に提唱したアジャイルソフトウェア開発宣言⁽²⁾には「プロセスやツールよりも個人と対話を」とある。プロセスやツールに価値があることを認めながらも、個人と対話することにより価値おくとされる。

ただし、この宣言が発表されたのは 15 年も前であり、状況は多少変化している。プロセスについては、軽量なアジャイル開発方法論である「スクラム」がよく知られるようになった。なお、スクラムの定義を記述したスクラムガイド⁽³⁾が作成されたのは 2011 年である。

また、Beck らがツールについての価値を低く見ていた背景には、CASE やモデリングツールなどの重

量級のツールを念頭にしていたからであろう。そうではない、ホワイトボードやポストイットのような軽量なツールは、アジャイル開発を進めるための道具として好んで用いられてきた。

ホワイトボード等は、確かに利用することが簡単ではあるが、不便な点もある。例えば、同じ場所で常に作業するチームでの開発には適しているものの、遠隔にいるメンバーとの分散開発を行う場合には不向きである。

このような理由もあり、近年はクラウド型サービス（SaaS）を、アジャイル開発を実施するための道具として活用する事例が多く見られるようになった。クラウド型であるので、情報を共有するために必ずしも同じ場を共有する必要がない。また、クラウド型のツールの一般的な特徴でもあるが、ツールが日々改良され使いやすくなってきている。

アジャイル開発において重要視されているのが「かんばん」である⁽⁴⁾。かんばんにはタスクを定義し、開発プロセスを視覚化する効果がある。この、かんばんを共有するためのツールとして、Pivotal Tracker, HuBoard, Trello 等が知られている⁽⁵⁾。

ここまで見てきた通りアジャイル開発においては、軽量で利用しやすいツールを利用し、近年ではクラウド型のツールも活用されている。そのため、これらのツールを上手に使いながら円滑に開発できるようになることが、アジャイル開発に対応するためのスキルセットには含まれることになる。

3. クラウドサービス・仮想環境の活用

一方で、アジャイル開発は 2012 年頃からのソーシャルコーディング⁽⁶⁾の潮流と融合している。ソーシャルコーディングの源流は OSS（オープンソースソフトウェア）の開発の流儀である。この開発スタイルを SaaS 型クラウドサービスとして実現したのが GitHub 等のサービスである。GitHub はソースコード管理や課題管理機能を提供する。これにより、OSS

のようにオープンかつ大規模な分散共同開発が容易になる。もっとも、アジャイル開発は必ずしもオープンソースではない。むしろ、GitHub を用いる場合の協同コーディングの流儀 (GitHub Flow⁽⁷⁾) の方が重要であり、この流儀に従いながらクローズドな開発を行うことも一般的である。

また、GitHub と他のクラウドサービスを連携させることでより多くの機能を利用することができる。例えば、継続的インテグレーション (CI) を提供する Travis CI などと連携させるケースは多く、テストの自動実行や完成したコードのリリースの自動化などができる。この他にも、Slack 等のコミュニケーションツール (メッセージャー) とも連携させ、各種の通知を送信することもできる。

仮想環境は、開発のための環境や、Web アプリケーションを運用するための環境として積極的に用いられる。物理的なサーバを構築する手間がらず、サーバの構成情報を電子化することもできるので、いつでも破棄・再構築できる。Heroku 等の PaaS 型のクラウドサービスではアプリケーションを運用するための仮想環境を提供しており、これを用いてインターネットに容易に公開できる。

4. 教育プログラムのコンセプト

以上を踏まえると、アジャイル開発に対応する技術者が備えるべきスキルセットとしては、1 つ目はアジャイル開発の「方法論」の理解であり、2 つ目は開発を円滑に進める「道具」の学習である。なお、プログラミングやデータベース、ネットワークや OS 等の基本知識も必要に応じて身に付ける必要があるが、ここでは触れない。以下、これらを学ぶために本学で実施している教育プログラムのコンセプトについて記す。

1 つ目のアジャイル開発の方法論に関し、「スクラム」の習得法について述べる。スクラムはその方法がスクラムガイドにより定義されていることは前述した。このガイドは日本語版で僅か 17 ページしかないので容易に習得できるのではないか、と思われるがそうではない。スクラムガイドには「スクラムとは軽量、理解が容易」であるにもかかわらず、「習得は困難」とある。ガイドに記述されているのはスクラムで用いる用語の定義であり、これを使用する戦略は様々だからである。

このため、スクラムを学習する方略としては、まず、スクラムガイドに基づき、スクラムで定義されている言葉についての共通理解を得る。次に、実際にスクラムによるソフトウェア開発プロジェクトを体験する、という流れが求められる。そこで、講義やワークショップ形式で言葉の共通理解を得、スクラムによる開発プロジェクトを PBL で実践する。

2 つ目の道具の学習については、各種のツールの有機的なインテグレーションをモデルとして提示し、これらを用いた開発のサイクルを学ぶことを目標と

する。ツールの選択肢は豊富であるが、学習効率の点から、代表的なインテグレーションの例を示し、それについて学ぶのが効果的だ。本学の例では、開発するソフトウェアは Web アプリケーションとし、プログラミング言語も Ruby に限定している。その上で、GitHub と Travis CI を連携させ、Heroku にリリースして動作させる環境をモデルとしている。

これらのツールを使いこなせるようになるには一定の訓練が必要だ。そこで、本学ではツールの使い方について演習中心の授業を実施し、その後、PBL に進むように授業設計した。

以上を整理すると、本教育プログラムでは、履修者はアジャイル開発の方法論と道具についての基礎知識を身につけてから、開発 PBL でそれらを総合的に学ぶという全体設計となっている。このように段階的に知識習得を進めるので、学習者にとっては体系的に理解しやすい内容となっている。

5. 本教育プログラムの実施と今後

本教育プログラムは、2013 年度より実施している本学における enPiT プログラムにおいて得られた知見を整理したものである。本学は専門職大学院であり、多くの社会人学生が受講している。彼らの中には今後、実務でアジャイル開発に取り組もうとしている者も多い。また、講師や外部評価者にはアジャイル開発の最先端で活躍されている方に協力して頂いている。このような環境の中で、様々な関与者から実務の観点からの多くの改善提案を頂くこともできた。大学における教育プログラムとして相当に実践的な内容となっていると考える。

今後、Web アプリケーションだけではなく、スマートフォンやタブレットで動作するアプリケーション、様々な IoT のプロダクトの開発にも対応できる教育プログラムへと発展させたい。

参考文献

- (1) 日経コンピュータ: “成功する IoT 製品開発——モノづくりにも Web のスピードを”, <http://itpro.nikkeibp.co.jp/atcl/column/16/020100027/020200003/>, (2016/02/10)
- (2) Beck, K 他.: “アジャイルソフトウェア開発宣言”, <http://www.agilemanifesto.org/iso/ja/>, (2001)
- (3) Beck, K 他.: “スクラムガイド - Scrum Guides”, <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-JA.pdf>, (2013)
- (4) Anderson, D. J. Kanban: “Successful evolutionary change in your technology business”, Blue Hole Press, Washington (2010)
- (5) SideCI: “アジャイルに効く「かんばん型」タスク管理ツール SaaS まとめ”, <http://blog-ja.sideci.com/entry/2014/10/23/171117> (2014)
- (6) Dabbish, L and Stuart, C, Tsay, J, and Herbsleb, J: “Social coding in GitHub: transparency and collaboration in an open software repository”, ACM, In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work (CSCW '12)*, pp. 1277-1286 (2012).
- (7) Chacon, S: “GitHub Flow”, <http://scottchacon.com/2011/08/31/github-flow.html> (2011)