

アジャイル開発手法の学生による実践実験

安澤 翔太[†] 松尾 和洋[‡]金沢工業大学工学研究科 BA 専攻[†] 金沢工業大学情報学部[‡]

1. はじめに

近年、情報化社会の進展に伴い、ソフトウェアの複雑化が進んでいる。複雑化が進むことで、ソフトウェア開発のコストや工数が肥大化し、開発者に大きな負担となっている。このような現状を打開するため、コストや工数を的確に管理し、負荷を軽減するソフトウェア開発の有効な手法の開発とその導入が必須となっている。

本発表では、最近注目されているアジャイル開発手法を採り上げ、大学における課外活動(プロジェクト)での実践を通して開発手法の実践実験を行った。このようなプロジェクトを実施するには、連携企業の協力が不可欠である。

2. アジャイル開発手法

アジャイル開発手法は、反復型開発を応用した開発手法である。反復型開発とは、ソフトウェア開発のマスタースケジュール内で、開発プロセスを反復して行い、常に顧客要求を反映することを意識した開発手法である。主なアジャイル開発手法には、XP(Extreme Programming)[1]をはじめとする様々な開発手法が存在する。

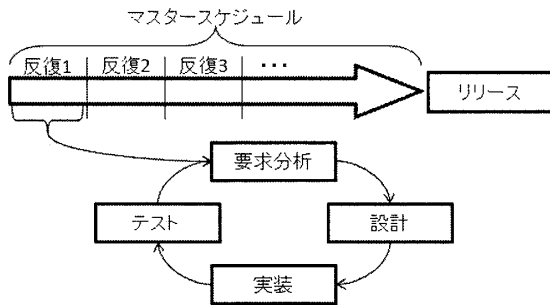


図1. アジャイル開発プロセス

大学において開発組織を形成し、ソフト開発手法を実践することは、企業等で実践経験を持つ指導者がいなければ難しい。今回の場合は、プロジェクトマネージャーとなる学生が企業でのインターンシップを通して開発手法を修得している。また、大学での実践に際しても当該企業より指導・評価を受けることができ、実践実験が可能となった。

Practices of Agile Software Development in School
[†]Shota Anzawa, Kanazawa Institute of Technology
[‡]Kazuhiro Matsuo, Kanazawa Institute of Technology

3. 実践プロジェクトの概要

プロジェクト実施環境：参加者は、プロジェクトマネージャーにインターンシップ経験のある院生、メンバーに学部2、3年生の7名で構成されている。

開発システム：大学の研究室活動を活性化するためのグループウェア KIT-Groupware

開発環境：Eclipse, MySQL, Mongrel

開発言語・フレームワーク：Ruby on Rails
 (Ruby: 1.8.6, Ruby on Rails: 2.0.2)

管理ツール：Redmine, Subversion

企業連携：連携企業には、プロジェクトのテーマ企画審査、イテレーション毎(2週間)の作業計画及び成果報告のレビューを依頼。

4. 実験の実施内容

この実験では、活動は3ヶ月=6イテレーションとし、実験の特徴は以下の通り。

イテレーション：開発の作業期間単位であるイテレーションを2週間と設定している。また、イテレーション初日を計画日、最終日をふりかえりとしている。ふりかえりでは、デモやKPTと呼ばれるプロセス改善等を行っている。

チケット駆動型開発(TiDD)：TiDDはRedmineやTracのような障害管理ツールを中心に開発ツールを駆使して、アジャイル開発手法をマネジメントする方法である。[2]プロジェクトでは、Redmineを利用し、開発作業や個人作業をチケットと呼ばれる作業単位に分割して管理している。

ペアプログラミング：メンバーはペアを形成し、開発作業を進める。ペア作業により、高品質なコード、知識伝達、強化学習、さらにはチームワークの向上[3]が期待できる。

テスト自動化：実装されたコードに対するテストコードの実行を自動化。毎日全てのコードが正常に動作するかを確認。テストサーバでテストを実行し、その結果をクライアントPCにレポートする仕組み。

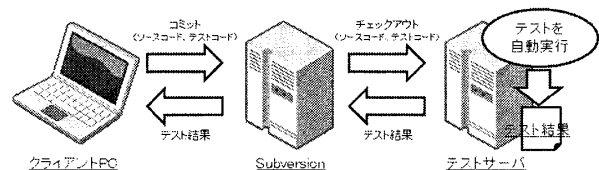


図2. テスト自動化の流れ

コード網羅率の測定：実装されたコードの命令や分岐がどれだけテストスクリプトで網羅されたかを示す指標。本プロジェクトでは、Ruby用のカバレッジツール rcov を利用。

ソフトウェアかんばん：ソフトウェアかんばんとは、かんばんツール。イテレーション内で何をやるのか、現在誰が何を作業しているか、レビューの負荷を把握するために用いる。

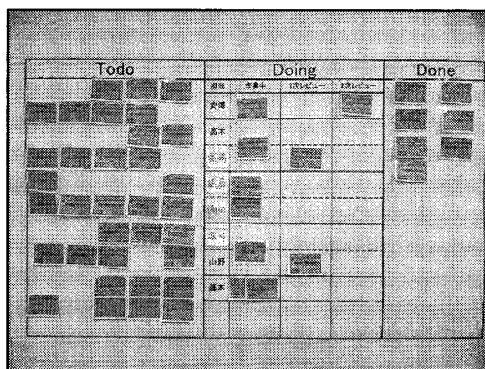


図 3. ソフトウェアかんばん

バーンダウンチャート：X 軸にイテレーション期間，Y 軸に総作業時間を設定し，毎日の作業の進捗管理を行うツール。作業の進捗を評価するために総予測作業時間から日別の予測作業時間を引いた線を理想線としている。また，リスク余地のために経験則から危険線を設定している。

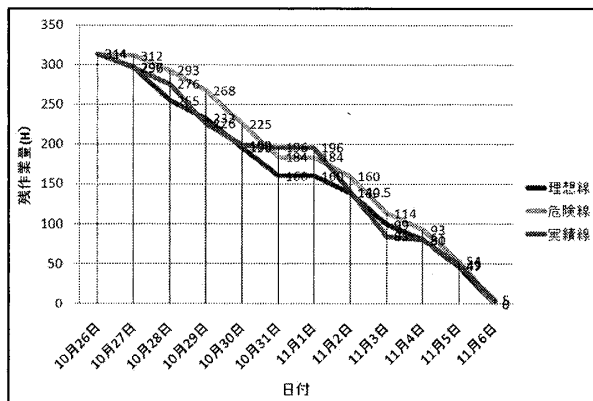


図 4. バーンダウンチャート

ふりかえり：各イテレーションの最終日にはふりかえりを行う。ふりかえりでは、KPT かんばんを利用し、イテレーションで良かった点、問題点、課題を検討する。課題については、誰が、何を、何時までにということを明確にし、Try かんばんで管理し、作業内容、開発プロセスの改善を行う。

5. 実験経過の考察

本プロジェクトを QCD(コード品質(Q), 作業コスト(C), 作業達成度(D))の面から分析する。

コード品質(Q)：単体テスト：96 件，機能テスト：742 件，コード網羅率：91.1%。

作業コスト(C)：学生によるプロジェクトのため，金銭的なコストの概念は存在しない。そのため，本プロジェクトでは予測作業時間＝コストと設定し，予実の結果で評価することとした。

1～6 イテレーションの総予測作業時間と総実績時間の結果は以下の通りである。

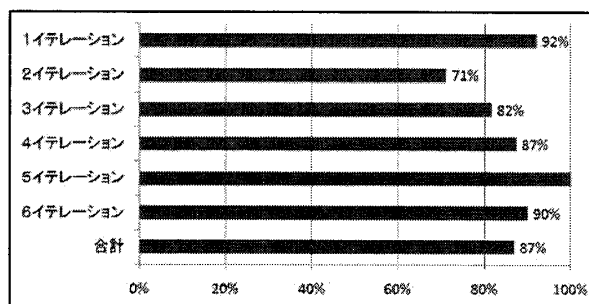


図 5. 総予測作業時間に対する実績時間の割合

以上の結果から，予定作業時間に対する実績作業時間の割合では，87%であった。予測作業時間より 13%の短い時間で達成出来た。

作業達成度(D)：各イテレーションの平均作業達成度は，96.8%であった。

6. まとめ

今回の実験では，学生によるアジャイル開発手法の実践を行った。実験の成果は，大学の講義では得られない，要求分析，設計，実装，テスト等の開発サイクルを学習出来たことである。参加学生からも同様の感想を得られた。

QCD の分析結果は，コード品質ではコード網羅率 91.1%，作業コストでは 13%の時間短縮，作業達成度では，高い値が示された。作業コストの分析は，見積りが作業者の自己申告のため，客観性に欠ける面がある。今後は，実績データから見積りの標準化を進め，客観性を高めたい。

本研究の一部は，大学 GP 大学教育・学生支援推進事業【テーマA】KIT ビジネスアーキテクトプロジェクトの活動として実施している。連携企業は富士通株式会社であり，プロジェクト推進への協力を感謝します。

参考文献

- [1] ケント・ベック, XP エクストリーム・プログラミング入門, ピアソンエデュケーション, 2005
- [2] 阪井 誠, TiDD: チケット駆動によるアジャイル開発法, ソフトウェア信頼性研究会 第 5 回 WS, 2009
- [3] Laurie Williams, Robert Kessler, ペアプログラミング, ピアソンエデュケーション, 2003