M-046

アジャイルソフトウェア開発における保守への引継ぎ支援のための開発履歴提供手法の提案

A Presentation Method of Software Development History to Support Knowledge Transfer for Maintenance in Agile Software Development

山中 啓祐 †
Keisuke Yamanaka

高田 秀志 ‡
Hideyuki Takada

1. はじめに

システム開発における顧客からの要求は多様化しており，また要求の変化が起こることも増加している．このような状況に対して，ソフトウェア開発手法の一つであるアジャイルソフトウェア開発は有効とされており，近年用いられるようになっている．アジャイルソフトウェア開発は，包括的なドキュメントの作成よりも動作するソフトウェアの開発を重視する．包括的なドキュメントの作成を行わない分の開発の細かい取り決めは，顧客と開発者や開発者同士のコミュニケーションの機会を増やし，会話で決定する．ドキュメントの作成・変更に注力しない分，要求の変更が発生した場合に迅速な適応が可能である．

アジャイルソフトウェア開発において，開発のメンバーと保守のメンバーが異なり，開発から保守への引継ぎが行われる場合，包括的なドキュメントを作成しないために，引継ぎ時は開発中に会話で決定した内容を伝えることが多い．このとき開発者がもれなく情報を伝えられるかどうかは，保守時の作業の効率に大きく関わる．開発者がもれなく情報を伝えるには，開発時のどのような作業の時に，会話によって開発の取り決めをしていたのかを開発者が把握しておく必要がある．

本稿では，アジャイルソフトウェア開発時において，開発者の作業対象の遷移の情報，開発中の会話量，および，作成されたドキュメントを用いた開発履歴情報を提供する手法を提案する．本手法によって，保守メンバーがドキュメントを見ただけでは理解することが難しいような，会話によって取り決められた箇所を，開発メンバーが把握して，保守メンバーに対して効率よく伝えることが可能となる．

2. アジャイルソフトウェア開発における保守への引き継ぎ

包括的なドキュメントを作成する開発手法では，ドキュメントを保守のメンバーに渡すことによって引継ぎを行うことが可能である．しかし，アジャイルソフトウェア開発では，簡潔に記述されたドキュメントが作成されるために，引継ぎ時はドキュメントを保守のメンバーに提供するだけでは成り立たない．また，包括的なドキュメントを作らない分，引き継ぎ方法も異なる．本節では，アジャイル開発におけるドキュメントの特徴と，保守への引継ぎにおける問題について述べる．

2.1 アジャイルソフトウェア開発におけるドキュメント

アジャイルソフトウェア開発は，開発における内容を優先順位と共にメモ程度に記述する．例えば，アジャイルソフトウェア開発の代表的な手法のエクストリーム・プログラミングでは，要求定義をまとめたものとして，図 1 のようなカード形式のドキュメントが作成される．このようなドキュメントは，開発者にとって，開発時のことを思い出すときに有用ではあるが，第三者からすると詳しいことは理解しにくい．

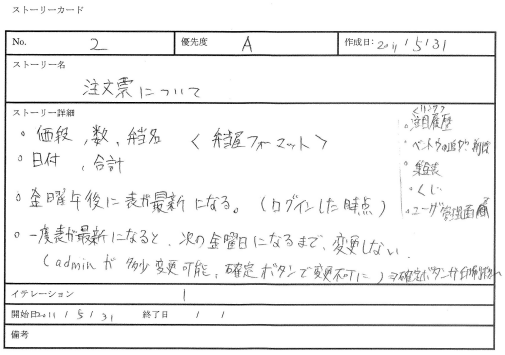

図 1: アジャイルソフトウェア開発におけるドキュメントの例

2.2 アジャイルソフトウェア開発における保守への引継ぎ時の問題

アジャイルソフトウェア開発において，開発のメンバーと保守のメンバーが異なる場合，保守への引き継ぎ方法が 2 種類存在する [1]．一つは，開発の終盤時に開発に保守のメンバーが加わり，一緒に作業を行い，設計やその背景といった開発情報を共有する方法である．もう一つは，開発後に開発のメンバーがドキュメントなどを基に開発情報をまとめ，保守のメンバーに伝達して引き継ぐ方法である．開発終盤に保守のメンバーを入れる方法は，保守のメンバーも開発に携わりつつ，開発情報を得ていくことができる．しかし，開発時に必要以上の人員を動かすことになるために，人的コストが高い．開発後に開発のメンバーから保守のメンバーに情報を伝達して引き継ぐ方法は，開発終盤に保守のメンバーを入れる方法より人的コストは低いが，伝達するときに活用できる情報源は少なく，ソースコードとそれをリバースエンジニアリングすることで得られる UML 図などのみとなり，それらだけでは保守のメンバーが設計の背景やシステム全体のことを理解することは難しい [2]．また，ドキュメントには記述されていない設計の背景やシステム全体のことを伝える場合，開発中に会話によって決定したことやその決定の背景にあったことは，開発者が記憶しているだけなので，曖昧になってしまう箇所が発生し，引継ぎ時に伝えられない場合がある．そのため，開発者が会話によって内容が決定された箇所を把握することが必要であると考えられる．また，会話によって内容が決定された箇所と共に，話す内容を思い出すためのものがあると，より円滑に保守への引継ぎ時の伝達を行うことができると考えられる．しかし，そのためには，引継ぎ時に必要な情報は開発中に会話によって決定した箇所であるという前提が必要となる．

†立命館大学大学院 理工学研究科
‡立命館大学 情報理工学部

3. アジャイルソフトウェア開発中の会話の調査

引継ぎ時に必要な情報が開発中に会話で決定した箇所であるという前提が正しいかどうかを確認するために，実際にアジャイルソフトウェア開発を行ってもらい調査を行った．

3.1 調査概要

アジャイルソフトウェア開発手法の 1 つであるエクストリーム・プログラミングを実践してもらい調査を行った．開発者役は 4 人，顧客役は 1 人であり，全員プログラミング経験のある学生である．今回行ったエクストリーム・プログラミングの実践は，テストファースト，計画ゲーム，短期リリースおよびコーディング規約である．イテレーションは 1 回が 10 日程度のものを 3 回行った．作成したものは，研究室で行っている弁当の注文を容易にする WEB アプリケーションである．開発者には，開発を行ってもらった後に，作成したクラスやメソッドについて開発中に行った会話の内容，保守への引継ぎを想定した場合にそれぞれのクラスやメソッドについて伝えるであろう内容，および，カードが開発中にどのようなを役割をしたかをアンケートで尋ねた．

各イテレーションでは，最初に顧客と会話し，顧客の要求を記述するストーリーカード，開発者が行うことを記述するタスクカードおよびクラスに関することを記述する CRC カードを作成した．その後，それぞれの担当箇所を決定し，コーディングを行った．加えて，実装途中に開発者のタスクや実装すべきメソッドなどができた場合は，必要なカードを逐一作成した．コーディングは各人の席で行った．不定期ではあるが，何日かに 1 回集まってそれぞれが作成した部品の結合を行った．このときにバグがあれば，1 つの机に 4 人でノートパソコンを持ち寄りあい，コーディングとデバッグ作業を行った．

3.2 調査結果

開発中は，各人の席で作業をしているときに，メンバー同士が会話をすることが見られた．

アンケートからは，開発中には，クラスの呼び出し関係の確認，実装するメソッドの仕様やアルゴリズム，および必要となったメソッドの作成依頼について話したという回答が得られた．保守への引継ぎをする場合何を話すかという項目には，各クラスの説明，どのクラスと関係を持つか（どういうクラスを呼び出すか）および，細かい設計の背景について話したと記述されていた．このことから，開発中に話すことと保守への引継ぎ時に話そうとすることは一致する箇所が多くあることが判明した．開発者それぞれの意見をみると，同じ箇所を何人か複数の開発者が担当していた場合，それぞれが開発中に話したことは同じでも，保守の時に伝えたい内容は異なる場合があった．

作成されたカードについては，他の開発者の作業に関わるときに，カードは他の開発者の作業内容を理解するときに役に立った，またカードは自分の作業内容が曖昧になったときにリマインダとして使える部分があったという回答を得られた．

4. 引継ぎ支援のための開発履歴提供手法

今回の調査から，開発中に話される内容は，保守への引継ぎ時に話さなければならないことであることが判明した．そのため，保守への引継ぎには，開発中の会話に関する情報を利用することが有効だと考えられる．本節では，保守への引継ぎ時に開発メンバーから保守メンバーへ効率よく情報を伝達するために必要な情報を開発中に記録し，引継ぎ時に参考となる情報を提供する手法について述べる．本手法では，まず，記録された会話の音量と，開発者の作業対象の遷移を利用し，会話が終了した時刻と，その時点で編集していたファイルを抽出する．会話の音量の変化から，開発者が，いつ，どれぐらいの時間の会話を行ったかを判別する．作業対象の遷移からは，開発者がいつ，どのソースコードを編集したかを取得する．図 2 のように 2 つの情報を組み合わせることによって，会話が行われたあとに，編集が行われている箇所を判別する．次に，ソースコードと対応するカードを関連付けする．この関連付けにより，会話で開発内容が決定されたソースコードと，それらに関連するカードの提供が可能となるため，保守のメンバーは，どの点が会話によって内容が決定したかということと，その部分についてのリマインダとなる情報を得て，作業を円滑に進めることが可能となる．

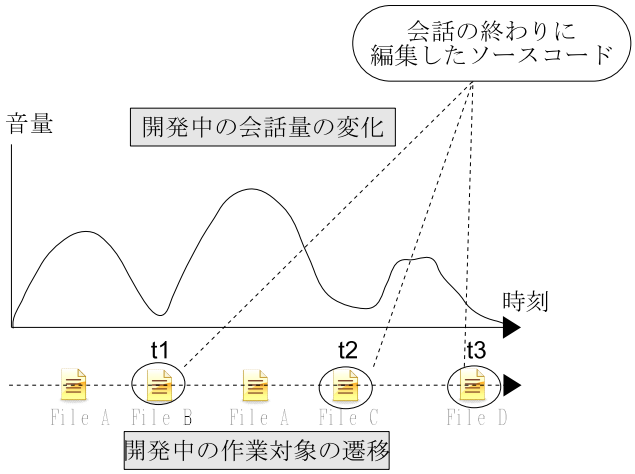

図 2: 会話によって内容が決定されたソースコードの選出方法

5. おわりに

本稿では，口頭のコミュニケーションが多いアジャイルソフトウェア開発では，開発から保守への引継ぎ時の伝達に必要な情報は，開発中にあると仮定し，実際にアジャイルソフトウェア開発を行うことによって得られた知見を報告した．報告内容から保守への引継ぎ時に情報の伝達を効率化するための情報提供手法について述べた．

今後は，提案した手法を統合開発環境 Eclipse のプラグインとして実装し，実際のソフトウェア開発で用いてもらい，保守への引継ぎ時の有効性を検証する．

参考文献

[1] ピート・マクブリーン，XP エクストリーム・プログラミング懐疑編，ピアソンエデュケーション，pp153-154，2002.

[2] 伏田 享平，大前 勇輝，名倉 正剛，川口 真司，大蔵 君治，飯田 元，バグ管理システムを対象としたアジャイルソフトウェア開発における保守プロセスの観察，電子情報通信学会技術研究報告. SS, ソフトウェアサイエンス，vol.108，No.362，pp1-6，2008