

ソフトウェア開発演習における成果物の編集履歴を活用した振り返りによる教訓抽出手法の提案

A Method to Extract Lessons Learned by Reflection Using Artifacts Editing History in Software Engineering Education

中浜 彩¹ 田中 昂文² 今野 翔太郎² 櫛山 淳雄¹
Aya Nakahama¹ Takafumi Tanaka² Shotaro Konno² Atsuo Hazeyama¹

¹東京学芸大学 ²東京学芸大学大学院
¹Tokyo Gakugei University ²Graduate School of Education Tokyo Gakugei University

概要: ソフトウェア開発演習における振り返りで、学習者が以後の開発に役立つ具体的な教訓を抽出することは重要である。しかし、教訓はしばしば抽象的な内容にとどまり、活用が難しい。そこで本研究では、学習者が成果物の編集履歴と、その背景にあるコミュニケーションのログを参照しながら振り返りを行うことで、具体的な教訓を抽出する手法を提案する。

Abstract: It is important for learners to extract concrete lessons that are useful to their subsequent development in their reflection process of software engineering education. However, lessons extracted often remain abstract, therefore difficult to reuse. This study proposes a method to extract concrete lessons by reflecting over referring to both editing histories in the artifact creation process and communication logs behind the background.

1. はじめに

教育機関において、ソフトウェア技術者を育成するためには、ソフトウェア工学の知識を活用して様々な場面における問題解決能力を養うための実践的なソフトウェア開発を経験する必要があるといわれている[5].

この要求に応え、教育機関等では実践的なソフトウェア開発演習を実施している。このような開発演習では、学習者は、開発の流れ、成果物の作り方、チームメンバおよび教授者とのコミュニケーションの取り方、開発を進めていく上でのコツなどを、開発に取り組みながら実践的に学ぶことができる。このようにして得られた学習項目は、学習者が以降の開発に活用できる重要なものである。本研究では、このような学習項目を教訓と呼ぶ。

学習者に教訓を意識させ、抽出させることは重要であり、そのためには開発プロセスの中で振り返りを行うことが有効だと言われている[1]。一般に、ソフトウェア開発における振り返りでは、一般化された汎用性の高い教訓を抽出して共有することに注力した研究や実践が盛んに行われている(例えば, [8])。しかし、従来の手法で得られる教訓は、その教訓が

いつ、どのような状況で生まれ、どのように活用できるのか、といったコンテキスト情報を重視していないため、教訓が抽象的すぎたり、当たり前のものであったりしがちである[4]。また、開発経験の少ない学習者は、一般化された教訓を実践に落とし込むための基盤となる経験や知識が不十分であり、教訓を十分に活用できない可能性があると考えられる。よって、ソフトウェア開発演習において振り返りを行う際には、具体的なコンテキストを含むなるべく詳細な教訓(以降、具体的な教訓)[4]を意識させ、以降の開発に適用するイメージを持ちやすくすることが重要であると考えられる。

具体的な教訓を抽出するには、開発プロセスに密着して振り返りを行い、詳細なコンテキスト情報を逃さずに抽出することが重要となる。しかし、開発プロセスで常に振り返りと教訓抽出を行い続けるのは学習者にとって大きな負担となる。これに対し、本研究では成果物の編集履歴(過去の成果物のスナップショット)を活用することを提案する。その理由は、多くの場合、教訓は成果物の編集、変更と深く関係しており、編集履歴は学習者が成果物の編集を行っていたときのコンテキスト情報を想起する重要な手がかりとなるからである。

これらの背景から、本研究ではソフトウェア開発演習における振り返りで学習者が具体的な教訓を抽出することを目的とし、成果物の編集履歴を活用して振り返りを行うことで具体的な教訓を抽出する手法を提案する。本稿では、要求仕様書作成工程を対象として提案手法を適用し、有効性を確認する。

2. 関連研究

本節では、関連研究として、ソフトウェア開発における振り返りと、教訓抽出に関する研究を紹介する。

Hazzan と Tomayko[3]はソフトウェア開発、特にソフトウェア工学教育において振り返りの重要性を述べている。彼らの講義において 12 のトピックに対して振り返りを行うことを受講者に求めている。しかしながら、過去の自身の経験に基づいて振り返りを行うことを求めているのみで、初学者が何らかの支援なしに的確な振り返りを行うことは困難であると考えられる。

Sharif らは様々な場面での教訓を抽出し活用するために、暗黙知と形式知からなる知識について SECI モデル[7]に着目した。共同化に教訓の収集、表出化に教訓の検証と蓄積、連結化に教訓の普及、内面化に教訓の再利用を対応させ、それに準ずる形で機能を策定し、教訓活用システム (LLS) を構築した[6]。しかしながら、本研究が目指す成果物編集過程と連携した形で教訓を抽出するものではない。

3. 教訓抽出手法

本節では、ソフトウェア開発演習において具体的な教訓を抽出する手法について述べる。

3.1 対象とする開発工程

本稿では要求仕様書作成工程を対象とする。要求仕様書の作成演習は、プログラミング等に比べて行われる機会が少ないが、ソフトウェア開発において重要な工程である。そのため、貴重な演習の機会として、得られる具体的な教訓を抽出することは学習者にとって重要であると考ええる。

3.2 教訓

Sharif らは、「教訓」とは、一般的によく似た状況において将来適用可能性のある、経験の短い記述であると述べている[6]。本研究では、ソフトウェア開発において経験から得ることができ、将来の開発で適用できる可能性のある学習項目を教訓と定義し、学習者自身にそれを記述させることで抽出する。また、教訓の記述の中にコンテキスト情報（いつ、だ

れが、どこで、なにを、なぜ、どのように）が多く含まれている教訓を具体的な教訓と定義する。

3.3 想定するソフトウェア開発演習

本研究では、ウォーターフォールモデルを開発プロセスとし、3~5 名程度のチームで数か月程度の期間にわたって行う開発演習を想定している[2]。各開発工程ではチームメンバが作成した成果物に対して教授者、ティーチングアシスタントがインスペクションを行う機会を数回設けるようにする。また、学習者は、各開発工程の終了時に開発活動の振り返りを行う。

3.4 対象とする振り返り

天野は、振り返りについて、プロジェクト中あるいは終了後にプロジェクトメンバ同士により行われるプロジェクト改善のための自主的な活動で、それまでの進め方に関して、良い点や問題点を自ら認識し、問題点に対する解決策を検討するものであるとしている[1]。しかし、本手法では、対象となるプロジェクトメンバが開発経験の少ない学習者であることから、プロジェクト全体としての活動よりもむしろ個々の学習者が自身の活動を個人で振り返る活動に注目する。

3.5 成果物の編集履歴の活用

ソフトウェア開発演習は全体の期間が数か月間であり、各工程の期間は 2~3 週間程度である。この期間に学習者が得る教訓は多く、学習者は教訓に関する詳細なコンテキスト情報を忘れてしまう可能性がある。このため、成果物の完成版など限られた情報をもとに振り返りを行っても、以後の開発に活用しにくい抽象的な教訓が抽出されてしまう場合が多く、詳細なコンテキスト情報を含む具体的な教訓は抽出されにくい。

そこで本手法では、次の方法（図 1）で振り返りを行うことを提案する。

開発工程において、学習者が成果物を作成する過程の編集履歴（成果物のスナップショット）をあらかじめ収集しておく（図 1 中の Change logs with snapshots of the artifact）。開発工程終了後、学習者が成果物の完成版を見ながら振り返りを行い、教訓を記述する（図 1 中の①）。その後、学習者は各教訓と関連する成果物の箇所注目し、その箇所の編集履歴を閲覧することで、その箇所がどのような変遷を経て完成版に至ったかを確認する（図 1 中の②）。さらに、その教訓や成果物の編集と関連するインスペクションコメントや関連する議論のログをあわせて閲覧する（図 1 中の②'）。図 1 中の②および②' の

活動により、教訓と関連する編集を行った理由や目的、編集のきっかけとなった会話など、教訓獲得時のコンテキスト情報を想起する手がかりとなる情報を得ることができ、具体的な教訓を抽出しやすくなると考える（図1中の③）。

本稿では要求仕様書作成工程を対象として、次節で提案手法の具体的な流れを述べる。

ここで、本手法における編集履歴の収集の粒度を定義する。本手法では、この粒度についての要件として、教訓と成果物の編集履歴の関連が明らかにできる粒度で編集履歴を収集しておく必要がある。本稿で対象とする要求仕様書作成においては、単語の追加、削除、変更の単位で編集履歴を収集することで、この要件を満たすと考える。

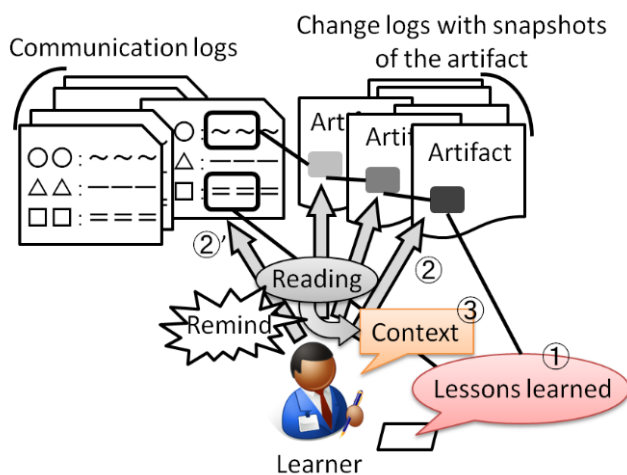


図1 提案する教訓抽出手順

3.6 要求仕様書作成工程での教訓抽出手順

要求仕様書作成工程の振り返りによる教訓抽出の手順を以下に述べる。

(1) 学習者が要求仕様書作成工程を、要求仕様書の完成版を見ながら振り返り、得られた教訓を「編集履歴閲覧前コメント」として記述する。

(2) 学習者は、それぞれの編集履歴閲覧前コメントの内容に関連する、要求仕様書の完成版の箇所を指定し、その箇所周辺の編集履歴を閲覧する。また、それらに関連するインスペクションコメントと議事録も同時に閲覧する。

(3) ある編集履歴閲覧前コメントに対して、(2)の手順を行ったことで新たに得られたコンテキスト情報や教訓を「編集履歴閲覧後コメント」として記述する。

編集履歴閲覧前コメントに記述された教訓がコンテキスト情報を多く含まない抽象的な内容であった

としても (2)でその教訓に関連するコンテキスト情報を想起し、新たな情報を (3)で記述することで、具体的な教訓が抽出できると期待される。

4. 実験

本節では、前節で述べた教訓抽出手法について、具体的な教訓を抽出するために編集履歴を閲覧することの有効性を確認するため、東京学芸大学情報教育専攻3年生に開講されている「システムプログラミング」の2014年度受講者6名を対象に実験を行った。実験では、要求仕様書作成工程終了時の振り返りに3.6項で述べた手順を導入し、教訓抽出を行った。ただし、手順の(2)に関しては、データ収集の制約から以下のように疑似的な方法を採用した。(2)' 学習者がそれぞれの編集履歴閲覧前コメントに対し、要求仕様書の完成版の該当する箇所に印をつける。印をつけた箇所を含む機能の記述に関して、筆者が版ごとの編集履歴を学習者に提供する。

5. 実験結果

本節では実験の結果について述べる。実験で得られた編集履歴閲覧前コメントおよび編集履歴閲覧後コメントの件数を表1に示す。また、得られたコメントの一部を表2に示す。表1より、編集履歴閲覧前コメントが28件あり、そのうち17件に編集履歴閲覧後コメントが追加されたことがわかる。

得られた28件の編集履歴閲覧前コメントおよび17件の編集履歴閲覧後コメントから、教訓に関するコンテキスト情報(What, Where, Who, Why, When, How)を抽出し、個数を比較した。表2中に下線と番号で示した部分はコンテキスト情報として抽出した記述である。また、番号の後の括弧内の「What」などは、コンテキスト情報の種類に対応している。比較の結果を表3に示す。

表1 コメントの件数

編集履歴閲覧前 コメント(件)	編集履歴閲覧後 コメント(件)
28	17

表2 得られたコメントの例

No	編集履歴参照前コメント	編集履歴参照後コメント
1	① (What) 言葉の定義はしっかりした方がいい.	① (How) その言葉は存在するものかどうか, 考えたほうがいい. ② (Why) 自分であると思ひ込んで書いても, 通じないときもあると感じた. ③ (How) 自分1人ではなく, 共有して気づいてもらうことも大事.
2	① (What) 機能同士 の矛盾をなくす	① (What) 1つの機能で決めたもの (パスワード) が② (Where) 他の機能で使おうと思った時に, ③ (Why) 例外系列となって④ (How) 使えなかった. ⑤ (What) 全体の機能の統合性を見なければいけないと感じた.
3	グループとしてではなく, ① (How) 個人として仕様書を作りそれを統合して成果物を作成したが, ② (What) 書式から③ (What) システムのイメージに多くの④ (How) 食い違いが生じた. ⑤ (How) グループ内でイメージを共有する前から仕様書を書き始めたという点が⑥ (When) その後のことにも影響し, ⑦ (What) 労力を増やしたので, その点を次に生かしたい.	なし
4	① (What) 主語はできるだけ省略しない.	なし

表3 コンテキスト情報の個数

編集履歴閲覧前コメントに含まれる 個数	編集履歴閲覧後で 追加された個数	合計
84	75	159

6. 考察

(1) 教訓抽出のための編集履歴閲覧の有効性

表1から, 編集履歴の閲覧は教訓に関する新たな情報の想起に有効であると考えられる.

表2のNo.1とNo.2より, 編集履歴閲覧前コメントではコンテキスト情報が少なく, 抽象的な内容だった教訓が, 編集履歴閲覧後コメントでコンテキスト情報が追加され, 具体的になっていることがわかる. No.3は編集履歴閲覧後コメントが記述されていないが, 編集履歴閲覧前コメントに7個のコンテキスト情報を含んでいる. 編集履歴閲覧前コメントに含まれるコンテキスト情報の個数の平均は $84/28=3$ 個であり, これと比較するとNo.3は編集履歴閲覧前コメントの段階ですでに7個と多数のコンテキスト情報を含んでいたため, 編集履歴閲覧後コメントが記述されなかったと推測される.

また, No.4のように, 内容が単純で様々な場面に容易に適用できる教訓については編集履歴閲覧後コメントが記述されない例が散見された.

さらに, 表3よりコンテキスト情報の個数は編集履歴閲覧後コメントにおいては編集履歴閲覧前コメントより2倍近くに増加していることがわかる. これらの事実から, 編集履歴閲覧後コメントではコンテキスト情報が増え, 教訓の記述が具体的になっているといえる. よって, 振り返りにおいて編集履歴を閲覧することが具体的な教訓の抽出に有効であると考えられ, 提案手法の有効性が示唆されたといえる.

(2) 明らかになった問題点

実験によって以下の問題点が明らかになった.

- (a) 編集履歴を閲覧する際に手間がかかる.
- (b) 記録されている編集履歴 (スナップショット) が少ない.

問題点 (a)について, 今回の実験で参照した各版の要求仕様書は10ページ以上にわたるものも見られたため, 編集履歴を閲覧する際に多大な労力と時間がかかった. 今回の実験で行った編集履歴閲覧の手順を実際の開発演習で開発工程ごとに行うのは現実的でないため, 支援ツールを構築するなどしてスムーズに編集履歴を閲覧できるようにする必要がある.

問題点 (b)について, 今回の実験ではデータの収集方法の制約から版ごとの編集履歴を用いた. しかし, 版ごとでは変更箇所が多いため, 編集履歴を閲覧する際に注目すべき箇所が限定しにくく, 具体的なコンテキスト情報の想起につながりにくいと考えられる. このことから, 適切な粒度で編集履歴を収集しておくことが重要であると考えている.

7. おわりに

本研究では, ソフトウェア開発演習において学習者が具体的な教訓を抽出するための方法として, 成果物の編集履歴を活用して振り返りを行う手法を提

案した。また、具体的な教訓抽出のために成果物の編集履歴を閲覧することの有効性を確認するため、実際にソフトウェア開発演習に取り組む学習者に対して実験を行った。その結果、編集履歴を閲覧することで、より具体的な教訓を得られることが分かった。

今後の課題としては、実験の結果から得られた問題点である編集履歴を閲覧する際に手間がかかること、記録されている編集履歴 (スナップショット) が少ないことについて、編集履歴を収集する適切な粒度を設定した上で、手法のシステム化による支援を検討していきたい。

参考文献

- [1] 天野 勝, プロジェクトファシリテーション実践編 ふりかえりガイド 第 20 版, <http://objectclub.jp/download/files/pf/RetrospectiveMeetingGuide.pdf> (参照 2014/3/27).
- [2] Atsuo, Hazeyama, Kazuyuki, Shimada, and Yuusuke Kobayashi, A Collaborative Problem Resolution Support System for Group-based Software Engineering Project Course and its Application, Proceedings of the Seventh International Conference on Creating, Connecting and Collaborating through Computing (C5'09), IEEE Computer Society, pp. 29-34, 2009.
- [3] Orit Hazzan, and James E. Tomayko, Reflection and Abstraction in Learning Software Engineering's Human Aspects, IEEE Computer, Vol.38, No.6, pp.39-45, 2005.
- [4] 独立行政法人情報処理推進機構, 情報処理システム高信頼化教訓集 (IT サービス編) 障害対策手法・事例集障害分析手法・事例集, 2014.
- [5] 松浦 佐江子, 実践的ソフトウェア開発演習によるソフトウェア工学教育, 情報処理学会論文誌, Vol.48, No.8, pp.2578-2595, 2007.
- [6] Mohd Nazir Ahmad Sharif, Kamaruddin Malik Mohamad, Rose Alinda Alias, and Shamsul Shahibudin Nor Hidayati Zakaria : Knowledge Management (KM) Framework for Representing Lessons Learned System for Communities of Practice in Institutions of Higher Learning, Malaysian Journal of Computer Science, Vol.17, No.1, pp.1-12, 2004.
- [7] 野中 郁次郎, 竹内 弘高, 梅本 勝博, 知識創造企業, 東洋経済新報社, 1996.
- [8] 岡田 公治, 小池 太, 内田 吉宣, 事業部門横断的なプロジェクト教訓事例の共有と教訓抽出のための分析手法, プロジェクトマネジメント学会研究発表大会予稿集 2010 (春季), pp.162-167, 2010.