推薦論文● FOSE2011

ソースコード差分のレビューにおける所要時間とレビューアの経験が指摘レベルに与える影響の分析

森崎 修司　田口 雅裕　松本 健一

旧バージョンとの差分のレビューにおいてレビューアの経験とレビュー時間が欠陥指摘レベルに与える影響を分析することを目的として，ソフトウェア開発の実務者を対象とした実験を実施した．Java で記述した既存のソースコード，4 件の変更仕様，各々の変更仕様に対応するソースコード差分 4 件を対象とし，差分が変更仕様を満たすかどうかを被験者にレビューしてもらった．66 名の被験者のレビュー指摘を 3 段階の指摘レベルに分類し，指摘レベルと被験者のレビュー時間との関係を調べた．また，65 名の被験者の経験 (プログラミング経験，対象ソースコードで使われているフレームワークの経験，レビュー経験) が指摘レベルに与える影響を分析した．レビュー時間と指摘レベルには統計的に有意といえる差はなかった．また，レビュー経験が指摘レベルに影響を与えていることがわかった．

In order to investigate the impact of reviewers' experience and review time to comprehension level of source code in software evolution, we conducted an experiment with software development practitioners using source code written in Java along with four modification tasks. Each task had a change specification and corresponding modified source code. The subjects were asked to judge whether the modified source code meets the corresponding change specification. The subjects recorded the time to review and their reasons for the judgments. Each reason for the judgment was categorized into one of the three comprehension levels. The results of 66 subjects showed that there was no statistically significant difference among comprehension levels and time to review. The results of 65 subjects showed that experience with review had large impact on comprehension level.

1　はじめに

近年，既存ソフトウェアを改変，追加する保守開発がソフトウェア開発の多くの割合を占めている．保守開発を想定したコスト要因の特定や，保守開発時の開発者の行動分析などの研究が数多くある [1] [2] [4] [5] [6] 保守開発では，追加，変更する部分に欠陥を混入しないようにするだけでなく，既存部分との整合性の考慮，将来の拡張や改変といった多面的な理解と評価が必要になる．レビューでの欠陥指摘やレビュー対象の理解にもいくつかの段階 (レベル) があると考えられる．

レビューでの欠陥指摘を調査した論文を大別すると，レビューアの経験とレビュー時間の関係を調査したもの [1] [6] とレビューアの経験とレビュー方針や読解方針を調査したもの [2] [4] に分類できる．文献 [1] [6] をはじめとするレビュー時間を調査した研究では，レビューアの経験や知識がレビュー時間を短縮するか調査し，レビューアのプログラミング経験の有無，対象ソフトウェアに関する知識の有無によって，ソースコードレビュー時間が小さくなることを報告している．しかし，これらの研究では，レビューアの指摘レベルや被験者の理解度を加味した言及はなく，大まかに理解し詳細な部分は飛ばして全体的な指摘のみで短時間でレビューを終わらせたのか，短時間で詳細まで理解し，きめ細かな欠陥指摘ができたのかは不明で

An Empirical Evaluation on the Impact of Reviewers' Experience and Review Time to Defect Detection Level in Source Code Patch Review.

Shuji Morisaki, 静岡大学, Faculty of Informatics, Shizuoka University.

Masahiro Taguchi, Kenichi Matsumoto, 奈良先端科学技術大学院大学, Graduate School of Information Science, Nara Institute of Science and Technology.

コンピュータソフトウェア, Vol.29, No.4 (2012), pp.74–80.
[研究論文 (レター)] 2012 年 4 月 30 日受付.

ある．

文献 [2] [4] をはじめとしたレビューアの経験や知識によってソースコードレビューの方針に違いが生じるか調査した論文では，レビューアの経験や知識によってレビューや読解の方針に違いが生じることを報告している．これらの研究では経験や知識とレビュー時間の関係や，経験や知識とレビュー方針の関係については言及されているが，指摘レベルへの影響は示されていない．

Dunsmore らは，ソースコード理解の実験において，被験者の理解度の深さに応じて指摘できる欠陥の数も大きくなることを報告している [3] が，被験者の持つ経験や知識と理解度の関係やレビュー時間に関する言及はしていない．また，理解度は被験者の自己申告によるものである．

本研究の目的は，保守開発のソースコードレビューにおいて，レビュー時間と指摘レベルの関係，レビューアの経験と指摘レベルの関係を実験を通じて明らかにすることにある．経験が異なるソフトウェア開発の実務者を被験者とし，既存ソースコード (バージョン 1) とそのソースコードに対する変更仕様，変更仕様に対応するソースコード (バージョン 2) をレビューしてもらう．レビュー結果から指摘レベルを 3 段階に分類し，指摘レベルとレビュー時間，指摘レベルと被験者の経験の関係を分析する．

2 実験

2.1 概要

本実験の目的は，保守開発におけるソースコードレビューにおいて，レビューアの指摘レベルとレビュー時間の関係，レビューアの指摘レベルがレビューアの経験に影響を受けるかを調べることにある．被験者には，既存のソースコード (バージョン 1 のソースコード)，4 件の変更仕様，変更仕様に対応するソースコード (バージョン 2)，バージョン 1 と 2 の差分を渡し，差分が変更仕様を満たせているかレビューしてもらう．

表 1 に実験で用いた 4 つの差分の概要を示す．被験者には，レビュー結果として差分適用の可否，その理由，発見した問題・欠陥，可否判断までの所要時間を記録してもらった．被験者は API 仕様などの資料は参照できるが，実験中に他被験者に対してアドバイスを求めたり相談したりすることはできない．

本実験は，ソフトウェア開発に携わる実務者であれば誰でも参加できる技術者向けイベントの 1 セッションとして実施した．日本 IBM が主催するソフトウェア技術者向けメールマガジンやオンラインメディアのアットマーク・アイティ (@IT) での記事告知をはじめとして，参加者を一般公募した．

対象ソフトウェアは Java アプレットとして実装したペイントアプリケーション (10[Class]，1600[LOC]) であり，描く色の選択，線を描く，描いたものを消す，線や描画領域からなる閉路を塗りつぶすなどの基本的な機能を持つ．GUI クラスライブラリには，Swing クラスライブラリを利用し，一般的な GUI プログラミングのフレームワークに従って実装している．

被験者にはバージョン 1 のソースコードとバージョン 2 をそれぞれ渡す．また，変更仕様とソースコード差分を記述した資料を渡す (図 1)．変更仕様には自然言語で変更意図，変更内容，変更前と変更後の挙動が記述される．イラストやスクリーンショットを含むものもある．ソースコード差分はバージョン 1 と 2 の差分を diff 形式で示したものである．

2.2 実験の流れ

被験者はまずバージョン 1 のソースコードを読解し，被験者が十分理解できたと判断した時点で読解を終了する．バージョン 1 のソースコード読解の終了条件は “将来の機能変更に備えたソースコードの理解” とした．次にバージョン 2 のソースコードについて，4 件の変更仕様を満たすものかレビューしてもらう．4 件の変更仕様と差分をレビューする順番や 1 件あたりのレビュー時間の上限，下限は指定していない．レビューの結果として差分の適応可否とその判断理由，発見した問題・欠陥を差分ごとに記述し，レビューの所要時間を記録してもらう．また，表 2 に示す被験者の経験に関するアンケートに回答してもらった．

2.3 指摘レベル

被験者が記述した差分適応可否の判断とその理由

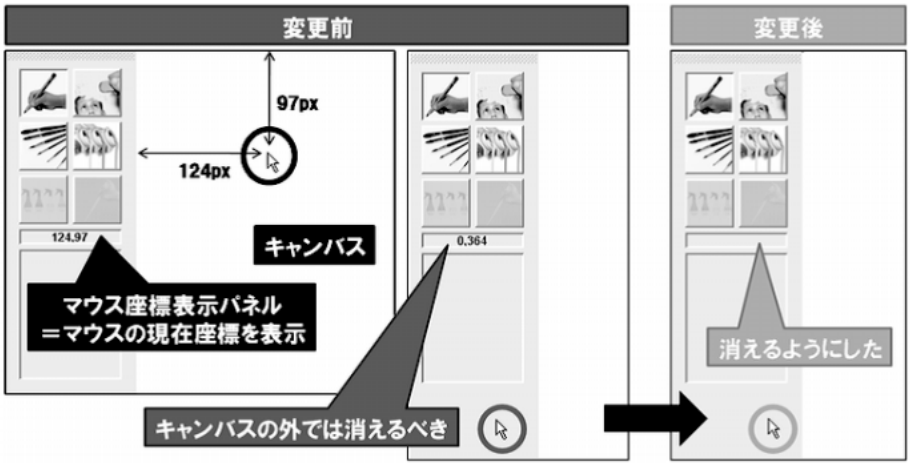
表 1　差分の概要

差分	追加	削除	GUI の変更	変更箇所数	説明
01	6	0	必要	1	マウス座標表示機能の修正
02	48	48	不要	11	クラス名の変更
03	1	58	必要	10	ブラシ描画機能の削除
04	1	1	必要	1	カラーパレットのモノクロ化

差分 **01**
「マウス座標表示パネル」の修正

変更

変更内容: マウスカーソルがキャンバス(絵の描画領域)の外に出たら「マウス座標表示パネル」に表示されているマウスの画面上の座標を消す。
変更前: マウスカーソルがキャンバス(絵の描画領域)の範囲外に移動した場合でも、マウスカーソルがキャンバス外に移動する直前の座標が表示されたままになる。
変更理由: マウスがキャンバス内にないことを明示的に示したいため



差分情報

パッチファイル（Unified diffs）

```
--- C:/Documents and Settings/yuichiro-y/workspace/reviewpaint/trunk/ReviewPaintMain.java   Mon Jun 01 05:12:12 2009
+++ C:/Documents and Settings/yuichiro-y/workspace/reviewpaint/branches/modify_cursor_position/ReviewPaintMain.java F
   01:09:36 2009
@@ -158,6 +158,12 @@

             jPanelCanvasPaper = new CanvasPanel(pt_pencil); // キャンバスクラスの生成、デフォルトのペイントツールの指定

+            // キャンバスからマウスが離れた時のイベントリスナ
+            jPanelCanvasPaper.addMouseListener(new java.awt.event.MouseAdapter() {
+                public void mouseExited(java.awt.event.MouseEvent e) {
+                    jLabelCursorPosition.setText(""); // 座標表示をオフにする
+                }
+            });
             // マウス移動時・マウスドラッグ時のイベントリスナ
             jPanelCanvasPaper.addMouseMotionListener(new java.awt.event.MouseMotionAdapter() {
                 public void mouseMoved(java.awt.event.MouseEvent e) {
```

図 1　被験者に渡した資料の例

表 2　被験者へのアンケート

質問	回答
GUI プログラミングの経験	“豊富な経験がある”，“経験がある”，“経験がない” のいずれか
レビューの経験	“豊富な経験がある”，“経験がある”，“経験がない” のいずれか
プログラミング経験	年数

表 3 指摘レベルの分類基準

差分番号	レベル	分類基準
01	1	座標表示消去用のイベントリスナの追加を確認している
01	2	追加したイベントリスナがマウスドラッグ中は無効であることを指摘している
02	1	リファクタリングで変更すべきソースコード (コンパイル対象) が全て変更されていることを確認している
02	2	コメント中にあるリファクタリング対象のクラス名の修正漏れを指摘している
03	1	ブラシ機能実装クラスの削除と機能選択ボタンの削除を確認している
03	2	必要がなくなったフィールド変数の削除漏れを指摘している
04	1	カラーパレットを内包するパネル (JPanel) のサイズ変更を確認している
04	2	OS・プラットフォームによって可視範囲に調整が必要になることを指摘している

から差分ごとに 3 つの指摘レベルに分類した．指摘レベル 0 は差分の適用可否と判断の間に矛盾がある，指摘レベル 1 は適用可否と判断の間に矛盾がなく，今後の保守において顕在化する可能性のある潜在的な問題，稀に起こる例外的問題を除いて，欠陥指摘ができている，指摘レベル 2 は指摘レベル 1 に加え、潜在的な問題，稀に起こる例外事象を含め，詳細に理解した上での問題・欠陥の指摘ができている，とした．それぞれの差分における指摘レベルの分類基準を表 3 に示す．

3 結果

分析の対象は差分 01〜04 の全てにおいてレビュー時間，差分適応可否の判断とその理由を記入していた被験者 66 名である．66 名のうち，経験に関するアンケート項目を記入していない被験者が 1 名いたため，経験による被験者分類において，被験者人数の合計は 65 名である．それぞれの指摘レベルの被験者人数を表 4 に示す．

3.1 指摘レベルとレビュー時間

帰無仮説を指摘レベルとレビュー時間の間に関係があるとはいえない，とした場合のクラスカル・ウォリス検定の結果 (p 値) を表 5 に示す．p 値が小さいほど指摘レベルとレビュー時間の間に関係がないことが否定される．4 つの差分とも 3 つの指摘レベルの間でレビュー時間に統計的に有意といえる差はなかった．

指摘レベルで分類したレビュー時間の分布を図 2 に示す．図中の箱の上端は，そのグループの被験者の 75 パーセンタイルを示す．箱の下端は 25 パーセンタイルを示し，箱の中の水平線は中央値を示す．箱から飛び出た線分の上端は 95 パーセンタイル，下端は 5 パーセンタイルを示す．全ての差分において，指摘レベル 1 の 5 パーセンタイルが指摘レベル 2 のそれよりも小さくなった．

表 4 被験者人数

指摘レベル	差分 1	差分 2	差分 3	差分 4
0	2	3	4	6
1	56	47	40	26
2	8	16	22	34

表 5 指摘レベルとレビュー時間の検定結果 (p 値)

	差分 1	差分 2	差分 3	差分 4
p 値	0.395	0.644	0.769	0.725

3.2 指摘レベルと経験

帰無仮説を指摘レベルと被験者の経験の間に関係があるとはいえない，とした場合のフィッシャーの正確確率検定の結果 (p 値) を表 6 に示す．表 5 と同様に p 値が小さいほど被験者の経験と指摘レベルの間に関係がないことが否定される．指摘レベルとレビュー経験との間には，差分 4 において，有意水準 5%で統計的有意差があった．また，差分 1, 2 の p 値が小さい値となった．プログラミング経験年数と GUI プログラミングの経験と指摘レベルの間には全ての差分に

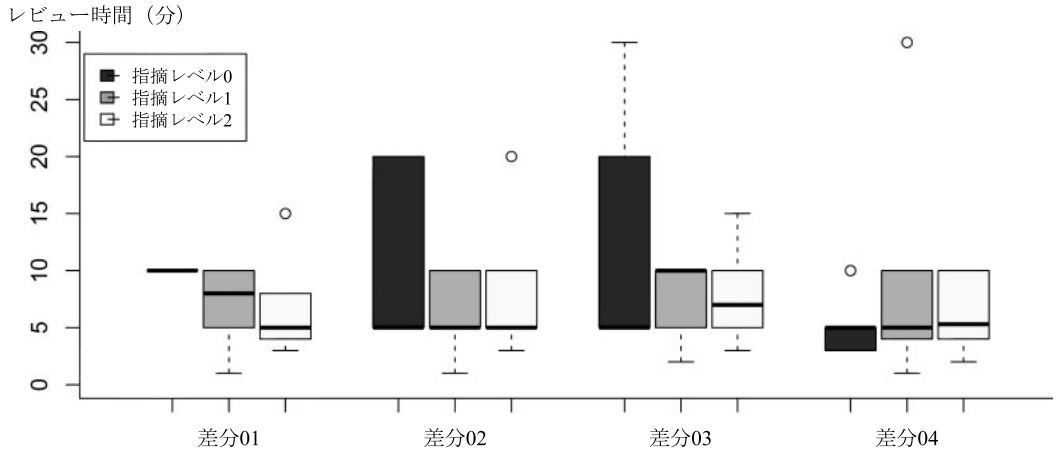


図 2　指摘レベルごとのレビュー時間の分布

表 6　指摘レベルと経験の検定結果 (p 値)

経験	差分 1	差分 2	差分 3	差分 4
主に使うプログラミング言語	0.673	0.953	0.873	0.690
プログラミング経験	0.824	0.760	0.558	1.000
GUI プログラミングの経験	0.708	0.630	0.342	0.592
レビュー経験	0.174	0.075	0.669	0.013

おいて統計的に有意といえる差はなかった．

4　考察

4.1　指摘レベルとレビュー時間

全ての差分において，指摘レベルの分類によるレビュー時間に統計的に有意といえる差はなかった．この結果は，ソフトウェア開発において一般的に収集されている単位時間あたりのレビュー進捗 (ドキュメントページ数やソースコード行数) の大小を単純に比較するだけでは，レビュー結果の妥当性を判断することが難しいことを示唆している．

全ての差分において，理解度 1 のレビュー時間の 5 パーセンタイルのほうが理解度 2 のレビュー時間のそれよりも小さかった．レビュー対象を十分に理解するためには，一定以上の時間が必要になると考えられる．レビュー時間が極端に短いときには，レビューに必要な最低限の時間が確保されているかを検討する必要がある．

4.2　指摘レベルと被験者の経験

3 つの差分において，レビュー経験が指摘レベルの分類に影響を与えていた．レビューアの育成においては，レビューの経験を優先的に積むことが効率的であることを示唆している．また，指摘レベルの高い欠陥指摘が必要な場合には，プログラミング経験をはじめとする開発経験の豊富なレビューアを割当てるのではなく，レビューに必要な着眼点等を整理し，そのような着眼点での指摘の経験があるメンバを優先して割当てることで，より効果的なレビューが期待できる．

今回の実験結果では，GUI プログラミングの経験が指摘レベルに影響を与えると推測される差分 1, 3, 4 において統計的に有意といえる差はなかった．理由の 1 つは，アンケートの質問項目 “GUI プログラミングの経験” だけでは，差分 1, 3, 4 に求められる個々の知識の有無を識別できないからであると考えている．たとえば，差分 1 では通常のマウス移動の際に発行されるイベントとマウスドラッグ中に発行されるマウス移動イベントが異なるという知識がなけれ

ば指摘レベル 2 の指摘はできない．アンケート項目 “GUI プログラミングの経験” は，通常のマウス移動のイベントとドラッグ中のマウス移動のイベントが異なるという知識を被験者が持っていることを必ずしも識別できない．

4.3 妥当性

実験で収集したレビュー時間は，被験者が計測したものである．実験を実施した部屋には掛け時計を設置し，その案内もしたため，時計を見ずに時間を計測した被験者はいないと推測されるが，その精度には限界がある．経験は被験者のアンケートへの回答によるものである．アンケートの質問には経験年数や具体的な活動を含める等，なるべく客観的な判断ができるよう工夫したが，被験者の主観を完全に排除できるものではない．

指摘レベルはレビュー結果の記述から分類したものであり，被験者が問題点・欠陥として認識していたが，その部分を記述しなかった可能性がある．しかしながら，実験終了後に自身のレビュー結果を他の被験者とグループで議論してもらうことを実験開始前に伝えており，理解できたことや発見した問題点や欠陥の多くはレビュー結果として記録されていることが期待される．

レビュー対象であるソースコードの規模は，商用開発やオープンソースソフトウェアで扱うソースコードの規模と比較すると小さい．しかし，変更規模や差分の規模は，現実の保守開発においても同等のものが多く存在すると考えられるため，その知見を活かすことができると考えている．また，本実験では，多くの実務者に同一対象をレビューしてもらうことを優先して，短時間で実験が終わるよう対象規模を小さいものとした．少人数で規模の大きな対象をレビューした場合の評価は重要な今後の課題の 1 つである．

5 まとめ

ソースコード差分のレビューにおいて指摘レベルとレビュー時間の関係，指摘レベルにレビューアの経験が与える影響を明らかにすることを目的として，ソフトウェア開発に携わる実務者を被験者として実験を行った．レビュー対象には Java アプレットとして実装したペイントアプリケーション (GUI アプリケーション) を用いた．バージョン 1 のソースコードとアプリケーションの説明，変更仕様 4 件，変更仕様に対応するソースコード差分 (バージョン 2 と 1 の差分) をレビューしてもらい，レビュー結果からレビュー時間と 3 段階の指摘レベルを得た．

66 名の被験者の結果を対象とし，指摘レベルで分類したレビュー時間の間に違いがあるかを検定したところ，統計的に有意な差はないことがわかった．単純にレビュー時間だけから欠陥指摘の質を推測するのは難しいことを示しているといえる．経験に関するアンケート項目を回答した 65 名の被験者の結果を対象とし，プログラミング経験年数，GUI プログラミングの経験，レビュー経験の 3 つの経験が指摘レベルに影響を与えているか検定した．4 件の差分のうち 3 件の差分においてレビュー経験が指摘レベルに影響を与えていることがわかった．そのうち 1 件は有意水準 5%としたときに統計的に有意な差があった．レビュー指摘のレベルを高めるためには，レビュー経験を優先的に積んでいくことが効率的であることを示しているといえる．

謝辞

実験にご協力いただいた被験者の方々に深く御礼申し上げる．実験の実施にあたりご協力いただいた日本 IBM 服部京子氏，春原秀仁氏に感謝する．本研究の一部は，文部科学省「次世代 IT 基盤構築のための研究開発」の委託に基づいて行われ，文部科学省科学研究補助費 (若手 B:課題番号 21700033, 基盤研究 B:課題番号 23300009) による助成を受けた．本論文の分析対象データの一部は IBM Academic Initiative Program (http://www.ibm.com/developerworks/university/academicinitiative/) の支援によって収集された．

参 考 文 献

[1] Benestad, H. C., Bente, A. and Erik, A.: Understanding cost drivers of software evolution: a quantitative and qualitative investigation of change effort in two evolving software systems, *Journal of Empir-*

ical Software Engineering, Vol. 15(2010), pp. 166–203.
[2] Corritore, C. L.: An exploratory study of program comprehension strategies of procedural and object-oriented programmers, *International Journal of Human-Computer Studies*, Vol. 54(2002), pp. 1–23.
[3] Dunsmore, A., Roper, M. and Wood, M.: The role of comprehension in software inspection, *Journal of Systems and Software*, Vol. 52(2000), pp. 121–129.
[4] Pennington, N.: Comprehension strategies in programming, in *Empirical Studies of Programmers: Second Workshop*, 1987, pp. 100–113.
[5] Robillard, M. P., Coelho, W. and Murphy, G. C.: How effective developers investigate source code: an exploratory study, *IEEE Transactions on Software Engineering*, Vol. 30, No. 12(2004), pp. 889–903.
[6] Soloway, E. and Ehrlich, K.: Empirical studies of programming knowledge, *IEEE Transactions on Software Engineering*, Vol. SE-10, No. 5(1984), pp. 595–609.

森崎修司

2001 年奈良先端科学技術大学院大学情報科学研究科博士課程修了．同年 (株) インターネットイニシアティブ入社．2007 年奈良先端科学技術大学院大学助教．2012 年静岡大学講師．ソフトウェアレビュー，エンピリカルソフトウェア工学の研究に従事．博士 (工学)．情報処理学会，電子情報通信学会，プロジェクトマネジメント学会，IEEE 各会員．

田口雅裕

2010 年神戸市立工業高等専門学校専攻科電気電子工学専攻修了．2012 年奈良先端科学技術大学院大学情報科学研究科博士前期課程修了．同年パナソニック株式会社入社．在学時，ソフトウェアレビューに関する研究に従事．

松本健一

1985 年大阪大学基礎工学部情報工学科卒業．1989 年同大学大学院博士課程中退．同年同大学基礎工学部情報工学科助手．1993 年奈良先端科学技術大学院大学情報科学研究科助教授．2001 年同大学教授．工学博士．エンピリカルソフトウェア工学，特に，プロジェクトデータ収集／利用支援の研究に従事．情報処理学会，電子情報通信学会，日本ソフトウェア科学会，ACM 各会員，IEEE Senior Member．