

# プログラミング演習におけるチームレビューが学生のプログラミング行動に与える影響の分析

堀口 諒人<sup>1</sup> 井垣 宏<sup>1</sup>

**概要:** プログラミング教育においてコードレビューは学生のコーディングスキルや授業満足度の向上に有用であることが知られている。本稿では、プログラミング演習におけるチームでのコードレビューが学生らのプログラミング行動にどのような影響を及ぼすか分析を行った。結果として、レビューを実施したクラスの学生のほうが、自身のコードをより積極的に改修する傾向があることが分かった。

## 1. はじめに

特定のプログラミング言語の習熟を目的としたプログラミング演習と呼ばれる形式の授業が数多くの大学で実施されている。プログラミング演習とは、学生に対して対象のプログラミング言語の実装を伴う課題が課され、教員やアシスタントから構成されるスタッフの指導のもとで実施される形式の授業である。著者らの大学では、2020年度コロナ禍の影響によるオンライン化に伴い、学生間交流の促進やコードリーディング能力の育成を目的とし、チームでのコードレビュー（以降チームレビューと呼ぶ）を演習内に取り入れた。プログラミング教育におけるコードレビューの有用性は良く知られており、プログラミング演習における学生の満足度やコード品質の改善に寄与するとの結果が示されている [1][2]。本研究では、課題提出締め切り後のコードに対してどのような振る舞いを行うか、すなわち学生自身の自習としてのプログラミング行動にどのような影響を与えるかを分析することで、チームレビューの有用性やその実施方法に対して検討を行う。

## 2. プログラミング演習とチームレビュー

2020年度の本学のプログラミング演習はA, B, C3クラスに分かれコロナ禍の影響もあり、オンライン形式で実施された。受講生は学部2年生330名（一部再履修生を含む）、言語はJava、開発環境としてJDKとVSCode、bashターミナルを学生自身のノートパソコン上にインストールさせた。演習では毎回10問近くの課題が出され、学生ら

はそれらの課題を開発環境上で実装し、ソースコードを締め切りまでに提出することが求められている。また、コンパイルやトラブル時の振り返り支援を目的として、開発環境におけるコードの編集履歴やコマンドの実行履歴を学生から許可を貰ったうえで提出してもらっている。

2020年度はオンラインでの実施ということもあり、レビューの体験及び学生同士の交流促進を目的として、チームレビューを毎回の授業の最初の30分～1時間程度実施した。なお、実施可否をクラス担当教員ごとに判断したため、今回はA, Bクラスのみの実施となった。

チームレビューでは課題の締め切り後に、各学生が提出したプログラムの入出力や仕様に準拠しているかといった項目とGoogle Java Style Guide<sup>\*1</sup>にもとづく簡易なチェックリストを教員が提示した。学生らには、課題ごとにチーム内（6名程度）で担当を決めてコードや入出力結果を他のメンバに提示し、他のメンバからのレビューを受けてその結果を記録するよう教員から指示した。

## 3. チームレビュー後に学生が行ったコード編集回数の分析

チームレビューは学生らに課された課題の締め切り直後の授業で行っている。そのため、チームレビュー以降に学生がレビュー対象のソースコードに対して編集を行った場合、チームレビューの影響を受けている可能性がある。そこで、チームレビューを実施したクラスA, Bはチームレビュー以降のタイミングでのコード編集内容、実施していないクラスCは締め切り直後の授業時間以降のタイミングでのコード編集内容をそれぞれ分析することで、学生のプログラミング行動への影響度合いを分析する。

<sup>1</sup> 大阪工業大学大学院 情報科学研究科  
Graduate School of Information Science and Technology, Osaka Institute of Technology 1-79-1 Kitayama, Hirakata City, Osaka, 573-0196 Japan

<sup>\*1</sup> <https://google.github.io/styleguide/javaguide.html>

### 3.1 チームレビュー実施後のクラス別ソースコード修正回数の分析

図1はチームレビュー以降にソースコードを編集した学生がどのくらいの割合存在するかをクラス別に示している。なお、チームレビューを実施していないクラスCについては、課題締め切り直後の授業以降にソースコードを編集した学生の割合を示している。チームレビューは全体で13回行われており、図の1回以上とは学生が少なくとも1回の授業の課題についてチームレビュー後に自身のコードを編集したことを示している（ここでは1回の授業に含まれる複数の課題についてコードを編集していても1回とみなしている）。また、赤、オレンジ、グレーの各棒グラフはそれぞれ課題締め切り直後の授業日当日、授業日翌日から一週間以内、授業日から8日以上経過後に、コードの編集を行った学生の割合を示している。

図1を見るとチームレビューを実施したクラスA, Bは実施していないクラスCに比べて、特に授業日にソースコードを編集した学生の割合が大きくなっている。実際に1度でも編集を行った学生を対象にクラスごとに何回ソースコードの編集を行っているか数えてみたところ、中央値がクラスA, Bの学生はほぼ4回、クラスCの学生は2回以下となった。

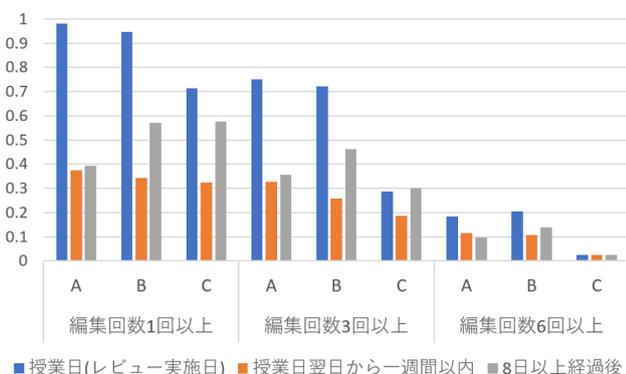


図1 課題締め切り直後の授業日（レビュー実施日）以降にコードを修正した学生の割合

## 4. 考察

本研究で収集したコード編集履歴にもとづいて、チームレビューを実施したクラスA, Bと実施していないクラスCで締め切り後のコード編集内容を比較した結果、特に締め切り直後の授業日にコードを編集した学生の割合において大きく差が付いた。また、3回以上の授業においてコードを編集した学生の割合は、すべての期間においてA, Bクラスのほうが大きくなった。

本研究で対象とするプログラミング演習では、提出された課題が仕様に準拠しているかを課題締め切り後に教員がテストしている。そこで締め切り後テストで仕様に準拠

```
1 for (int i = 0; i < array.length; i++) {  
2     - int index = i;  
3     - int min = array[i];  
4     + int min = i;  
5     for (int j = i + 1; j < array.length; j++) {  
6         - if (min > array[j]) {  
7             - index = j;  
8             + if (array[min] > array[j]) {  
9                 + min = j;  
10        }  
11    }  
12 }
```

図2 修正コードの例

していると判定されたソースコードを対象に、チームレビュー後のコード編集がどのように行われたかを分析した。

レビュー後の編集内容で特徴的だったものを図2に示す。赤字がレビュー後に削除されたコードであり、緑字がレビュー後に追加されたコードを示している。このコードはソートに関する処理を含む課題の一部を示しているが、レビュー後の編集内容を見ると、レビュー前に存在していた無駄な変数を削除するようリファクタリングが行われていることがわかる。

この例の他にも、コメントアウトによりコードの説明を追記するような修正や実行はできるが警告文が出されていたコードを修正している事例、warningの削除を目的として正常に動作しなくなるような修正を行ってしまっている事例など様々な種類の修正が確認できた。今後はどのような修正が行われているかより詳細に確認し、レビューの影響のより詳細な分析を進めていきたい。

## 5. おわりに

チームレビューを取り入れたクラスとそうでないクラスの間で、レビュー後の自発的なコード編集行動に差があることが分かった。今後は具体的な編集内容についてより詳細に分析し、さらに自発的なコード編集を促進する効果をもったチームレビューを授業に取り入れていくことなどを検討している。

### 謝辞

本研究の一部はJSPS 科研費17K00500の助成を受けた。

### 参考文献

- [1] Rong, G., Li, J., Xie, M. and Zheng, T.: The effect of checklist in code review for inexperienced students: An empirical study, *2012 IEEE 25th Conference on Software Engineering Education and Training*, IEEE, pp. 120–124 (2012).
- [2] Wang, Y., Li, H., Feng, Y., Jiang, Y. and Liu, Y.: Assessment of programming language learning based on peer code review model: Implementation and experience report, *Computers & Education*, Vol. 59, No. 2, pp. 412–422 (2012).