個別スキルに基づく割り振りによるコードレビュー効率化の手法

1250286　稲田　慈英　【 コミュニケーション＆コラボレーション研究室 】

1　はじめに

コードレビューは，ソフトウェア開発において誤りを検出・修正することを目的とし，開発チーム内でソースコードの品質を検証する作業である．レビューの精度および時間効率の向上は重要な課題である．

本研究では，個人のスキルやバグの分類ごとに人員を最適に割り振ることで，コードレビューの効率化を図ることを目指す．このアプローチにより，レビューの品質向上と時間短縮を実現し，ソフトウェア開発プロセス全体の生産性向上に寄与することが期待される．

2　コードレビュー効率化手法

本研究では，評価式をもとにレビュースキルを算出し，その値に応じて，バグの分類ごとに人員を割り振ることで，効率化する手法を提案する．ソースコード中で出現頻度が高い 3 種類のバグを示す．本来の仕様と実装されている機能の相違に起因する「機能不良」，制御構造やアルゴリズムに起因する「構造不良」，データの使用や形式，種類，初期値に起因する「データ不良」を用意した [1]．個人の修正スキルの評価式を以下のように定義した．

$$E = \frac{2PR}{P+R}$$

P，R，E はそれぞれ精度，単位時間あたりの正しい修正数，評価値を意味する．評価値が大きいほど修正スキルが高いと判断する．

3　実験

本研究では，評価値をもとにバグの分類ごとに人員を割り振ることで，精度や時間的な効率を向上することが可能かを調査する．準備試行を含めた各 20 分の実験を 3 回行った．参加者は各試行において，3 種類のバグを 10 個ずつ含む Java コードをレビューした．試行中は，カーソル位置，スクロール位置，修正箇所，修正時間，修正内容を記録した．

4　結果

試行・分類ごとに平均化した処理時間に Friedman 検定を行った結果，$p = 0.02$ (< 0.05) となった．同様に，回答数に対しても検定を行った結果，$p = 0.0005$ (< 0.01) となった．2 試行の結果にそれぞれ評価式を適用し，それらの値に Wilcoxon の符号付き順位検定を使用し両側検定を行った結果，$p = 0.70$ となった．評価値と分類ごとの回答数・平均処理時間の相関係数は以下のようになった．

表 1　評価値との相関係数

	回答数	平均処理時間
機能不良	0.65	-0.93
構造不良	0.80	-0.79
データ不良	0.30	-0.76

5　考察

分類ごとの参加者全体の回答数の平均値を求めた結果，機能不良，構造不良，データ不良の順でそれぞれ 17，13.2，6.4 となった．この値と試行・分類ごとに平均化した処理時間に検定を行った結果を加味すると，難易度が高い順に機能不良，構造不良，データ不良であった．

2 試行間での評価値の検定結果に有意差が現れなかったことより，評価値が試行に左右されず参加者ごとに一定の評価値が算出されることが確認できた．

評価値と分類ごとの回答数・平均処理時間の相関係数より，平均処理時間はどの分類においても強い負の相関が見られた．そのため，評価値が高いほど処理時間が短くなることが確認できた．一方，回答数については機能不良，構造不良に強い正の相関が見られ，データ不良においては弱い正の相関が見られた．この結果より，ほとんどの場合において，評価値が高い参加者は正解率が高く，単位時間あたりの正解数が多くなっていることがわかる．今回の実験では，調べ物をしながらのレビューを制限していた．そのため，データ不良は回答数は変わらないが処理時間が短くなるため，評価値が高い人員を優先的に割り当てることで，更に詳しく分析する時間的余裕が生じることが期待される．同様に，機能不良，構造不良の順で評価値が高い人員から配置することで効率化が期待される．

6　まとめ

本研究では，評価値とバグの分類の関係について調査し，レビュースキルを定量的に評価した．結果，評価値が高いレビュアーから分類の難易度順に人員を割り振ることで組織全体でレビューを効率化できることが示唆された．また LLM の発展により，コード中のバグの種類を大まかに推測可能となった．今後，LLM とともに提案手法を用いることでコードレビューのさらなる効率化を目指す．

参考文献

[1] 應治沙織, 上野秀剛．コードレビュー時の読み方教示によるレビュー効率の変化．Technical Report 1, 奈良工業高等専門学校, 奈良工業高等専門学校, jul 2014.