

GitHub を活用したソフトウェアインスペクション支援の試み

橋浦 弘明[†] 遠藤 亮河[†] 橋本 諒介[†] 樋山 淳雄[‡]

日本工業大学[†]

東京学芸大学[‡]

1. はじめに

近年のソフトウェア開発では GitHub[1]等のソースコードホスティングサービスが広く用いられている。このような開発におけるソフトウェアの欠陥の報告やディスカッションは Issue 管理機能によって行われていることが多い。

ソフトウェアの欠陥を検出するためにはソフトウェアインスペクション[2]が有効であることが知られている。インスペクションとは、ソフトウェアの中間成果物、成果物を対象として、対象の作成担当者以外が成果物を目視で確認することによって欠陥を発見、指摘する活動である[3]。一方で、このような作業を経験の浅い開発者が実施することは難しいという問題がある。

2. 研究目的

前述の問題が発生する原因として、GitHub 等に搭載されている Issue 管理機能は開発者が書き込みをするための入力欄はテキストによる自由記述となっている。このため、経験が浅い開発者がインスペクションを行う際に以下の2つの問題が発生する。

P1: インスペクション対象のどのような点に着目すれば良いかわからない

P2: 指摘にどのような内容を含めれば良いかわからない

これらの問題により、インスペクション時に指摘内容の記述に時間がかかったり、曖昧な記述をしてしまうなど、有効な指摘ができないことがある。本研究では、これらの問題を解決するために、インスペクション時に必要となる指摘項目をテンプレート化すると共に、GitHub に適した形で実装する方法を提案する。

3. 提案手法と実現方法

前節で述べた P1, P2 の問題を解決するために、本研究では IEEE 1028[4]からインスペクションを

行う上で必要となる項目の抽出を行った。さらに、ソフトウェア開発に GitHub が用いられていることを前提に、GitHub の機能を活用してインスペクションを行うためのフローを開発し、GitHub 上に実装を行った。

本研究が提案するソフトウェアインスペクションの流れを以下に示す。

1. 開発者が機能の追加や修正を行おうとする場合に、プルリクエストの機能を用いて変更提案を行う。この際に図1に示すプルリクエストのテンプレートを用いて、機能の追加や修正に関するシナリオを添付する。
2. インスペクション担当者は、プルリクエストに添付されたシナリオを参考にインスペクションを行い、図2に示す欠陥指摘フォームを用いて結果を報告する。

図1: プルリクエストのテンプレート

図2: 欠陥指摘フォーム

An Attempt to Support Software Inspection Utilizing GitHub

Hiroaki Hashiura[†] Ryoga Endo[†] Ryosuke Hashimoto[†] Atsuo Hazeiyama[‡]

Nippon Institute of Technology[†]
Tokyo Gakuhei University[‡]

4. 実験

提案手法の有効性を確認するために実験を行った。実験は実際に動作する簡単なシステムに対して被験者がインスペクションを行い、提案手法によってどの程度欠陥の指摘が行えるかどうかを確かめるものである。

インスペクション対象となるシステムは、著者らが開発した施設予約システムとした。まず、システムに関する要求仕様書を準備し、次にシステムに要求仕様書に反するような欠陥を組み込んでおく。欠陥の組み込みに際しては欠陥の偏りを防ぐために Yuepu と Sampath [5]が提案している Web 欠陥分類リストを用い、対象のシステムに合計 20 個の欠陥の組み込みを行った。実験の評価はこれらの欠陥に対して、適切に指摘できた数や誤った指摘の数を集計することによって行う。実験の具体的な流れは以下の通りである。

1. 事前学習 (15 分)

- ・ レビューの重要性と品質について基礎的な内容の説明

2. 実験説明 (15 分)

2.1. 要求仕様書を配布

2.2a. (ツールありの場合)

2.2a.1. ツールマニュアル配布

2.2a.2. ツールの使用方法を説明

2.2b. (ツールなしの場合)

2.2b.1. 欠陥指摘シートを配布

2.3. 説明内容に関する質疑応答

3. 実験 (60 分～90 分)

- ・ 対象のシステムに対するレビューを実施

実験の被験者は日本工業大学工学部情報工学科・情報メディア工学科に所属する 12 名とし、無作為に半数をツールあり（実験群）、残りの半数をツールなし（統制群）に割り振った。

5. 実験結果と考察

実験結果の概要を表 1 に示す。表中の値はそれぞれ 6 人の被験者の平均値である。ツールありの方が適切な欠陥指摘数が減少しているように見えるが、これらの結果に対してマンホイットニーの U 検定を行った結果、有意差は認められなかった ($\alpha = 0.05$, $p = 0.061$)。

一方で、不適切な欠陥指摘数はレビューに要する時間が増加しているにも関わらず、発生件数が半減している。このことに着目し、時間あたりの不適切な欠陥指摘数を求め（表 2）、これらに対してマンホイットニーの U 検定を行った結果、有意差が認められた ($\alpha = 0.05$, $p = 0.020$)。

このことから、本提案手法で用いた欠陥指摘のためのテンプレートがソフトウェアインスペクションに対して一定のガイドラインの役割を果たし、初学者の不適切な欠陥指摘を抑制する効果があることが明らかになった。

表 1： 実験結果の概要

#	評価項目	ツール	ツール
		あり	なし
1	適切な欠陥指摘数[件]	4.17	5.83
2	不適切な欠陥指摘数[件]	1.00	2.17
3	実験時間 [min]	88.3	71.7

表 2： 時間あたりの不適切な欠陥指摘数

#	評価項目	ツール	ツール
		あり	なし
1	時間あたりの不適切な欠陥指摘数 [件/min]	0.01	0.03

6. 結論

本研究では、GitHub を用いたソフトウェアインスペクション手法を提案し、実験によって本手法が不適切な指摘効率の減少に効果があることを明らかにした。今後は有効な指摘数を増やすための提案手法の改良および、ソフトウェア開発 PBL に対する適用などさらに実践的な環境での評価を実施していきたい。

謝辞

本研究の一部は JSPS 科研費 20K12089 の助成を受けた。

参考文献

- [1] GitHub, Inc., “GitHub,” <<https://github.com/>>, (Accessed 2021/1/7).
- [2] 森崎修司, “インスペクションの動向,” 情報処理, Vol.50, No.5, May, 2009.
- [3] K. E. Wiegers, “ピアレビュー 高品質ソフトウェア開発のために,” 日経 BP, Feb. 2004.
- [4] IEEE, “IEEE 1028-2008 - IEEE Standard for Software Reviews and Audits,” Jun. 2008.
- [5] GitHub, Inc., “テンプレートを使用して便利な Issue やプルリクエストを推進する,” <<https://docs.github.com/ja/free-pro-team@latest/github/building-a-strong-community/using-templates-to-encourage-useful-issues-and-pull-requests>>, (Accessed 2021/1/7).
- [6] G. Yuepu, S. Sampath, “Web Application Fault Classification-An Exploratory Study,” In Proceedings of the Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM 08). pp.303-305, Jan. 2008.