

Refactoring-Aware Code Review: A Quasi-Experiment Approach

Q1 2019

In this empirical study, our hypothesis under investigation is stated as follows:

H₀. There is no significant difference between refactoring-aware and no refactoring-aware code review.

H_a. There is significant difference between a refactoring-aware and a no refactoring-aware code review.

In order to characterize refactoring-aware code review process at ePol project, we carried out the steps described in the next sections.

1. OBSERVATION OF REVIEWERS IN ACTION

Date	Session time	Change type	#files	#add	#del	#discussions
March 14th	78 minutes	improvement	30	1,787	236	13
March 15th	29 minutes	bug-fix	10	69	85	8

The change code may be either an improvement or a bug-fix. Both reviewers (respectively, 9 and 31 months of experience time at ePol project) conferred a change related-description during the code review session.

During the first session, in case of doubts, the reviewer personally contacted the code's submitter. The second reviewer checked the code related-tests before doing suggestions. Both reviewers accessed system's codes for figuring out the change impacts. As an additional comment, the reviewers emphasized the difficulty in understanding, sometimes, the intention of some code changes.

At ePol project, a code reviewing occurs in rounds, being performed by different reviewers until all discussions (anomalies found) are resolved.

2. PRE-TEST

2.1. PLANNING

2.1.1. *RefactoringMiner* RUNNING FOR ePOL REPOSITORIES

On March 18th, we ran *RefactoringMiner* to commits of ePol repositories under regular maintenance (epol-web, epol-rest, correicao-web, and correicao-rest), obtaining these results.

Output file	#commits	#refactorings
epol-web-Refactorings	18,713	3,298
epol-rest-Refactorings	12,233	8,102
correicao-web-Refactorings	565	0
correicao-rest-Refactorings	238	73

Table 2.1: Commits' description

2.1.2. SELECTION OF CHANGESETS

On March 19th, we selected these changesets considering the performing time for code reviewing and the number of multiple refactorings implemented in the same commit. All refactorings were manually validated.

Repository	SHA (last 4 digits)	#refactorings	Refactoring types
epol-rest	936e 8 files 151 additions 57 deletions	6	Rename Method Extract Class Move Method
epol-rest	b8b8 5 files 75 additions 47 deletions	5	Extract Method Rename Variable Move Method

Table 2.2: Commits' description

2.1.3. SETTING OF THE PRE-TEST ENVIRONMENT

On March 20th, we set up a machine with *SimpleScreenRecorder* tool to record the code review sessions and Internet access.

2.1.4. PROPOSAL FOR METRICS AND CODE REVIEW RELATED-QUESTIONS

Initially, a few metrics and questions are proposed for observation, as follows.

- session time;

- number of discussions (anomalies/issues);
- Is there any code change understanding difference between a refactoring-aware and a no refactoring-aware code review session?
- Have reviewers understood the code changes' intention?
- Have reviewers used the ePol project checklist for code review?
- Are reviewers familiar with the changeset's refactorings?

2.2. EXECUTION

Participants: ePol developers. To clarify, developers at ePol have been working as submitters and code reviewers. Next, the reviewers' experience description (Table 2.3). Sessions were executed on March 25th (#1 and #2), and March 28th, 2019 (#3 and #4).

Experience	Reviewer #1	Reviewer #2
Experience time at ePol (in months)	31	14
Experience time in Java programming (in months)	48	27
Experience time in performing code refactoring (in months)	48	14
Familiarity with the refactoring types have applied at session's commit	Yes	Yes
Frequency of performing code reviews	on average, 3 times/week	at least, 1 time/day
Frequency of performing refactorings	on demand	on demand
Manner of performing refactorings	manually	manually

Table 2.3: Reviewers' experience description

During the code review sessions (summarized in Table 2.4), in addition to the screen/audio recording, the reviewers have registered their work through the thinking aloud method. At the ending of sessions, each reviewer was interviewed considering the code review practice related-questions (Subsection 2.1.4).

The reviewer #1 performed sessions #1 and #2 and consulted the list of refactorings applied once, at the start of session #1.

A. PRE-TEST APPENDICES

A.1. GENERAL COMMENT

Nesse pre-test, não simulamos um ambiente real de code review no ePol, pois os reviewers efetuaram a revisão de commits ao invés de merge requests (que, por sua vez, acompanham uma descrição da mudança).

Session	Refactoring-aware	Commit	#files	#add	#del	Duration (in minutes)	#disc	#doubts
#1	Yes	936e	8	151	57	23	7	1
#2	No	b8b8	5	75	47	9	1	0
#3	No	b8b8	5	75	47	5	0	0
#4	Yes	936e	8	151	57	9	5	0

Table 2.4: Code review sessions results

A.2. CODE REVIEW SESSIONS #1 AND #2

A prática do reviewer #1 revisa: a documentação Javadoc, notação dos métodos de teste, comportamento. Inicialmente, o reviewer alertou que a descrição do commit (bastante resumida) já dá indícios das alterações e, portanto, da aplicação de refatoramentos.

Na sessão #2, dos 5 refatoramentos aplicados, o reviewer conseguiu detectar 2 *Rename Variable* e 1 *Move Method*.

- **#discussions**

Na session #1: 7 discussões foram levantadas pelo reviewer #1. 2 relacionadas à lógica, 1 sugerindo a inclusão da descrição de um parâmetro na documentação Javadoc, e 4 indicando a necessidade de alterar os testes desenvolvidos (ao invés de @Test, investigar o retorno do try-catch).

Na session #2: 1 discussão relacionada à inconsistência entre a assinatura e o comportamento de um método (inclusive, proveniente de um *Extract Method*).

- **#doubts**

Na session #1: 1 dúvida surgiu a respeito do *Extract Class* efetuado, pois o reviewer gostaria de verificar outras classes de validação para entender porque a classe não tem documentação Javadoc.

- **Is there any code change understanding difference between a refactoring-aware and a no refactoring-aware code review session?**

Sobre a session #1: "foi bom ter pego o commit (*lista de refactorings aplicados no commit*) antes, ter visto os refatoramentos anteriores, porque me fez assim me situar bem no contexto que eu 'tava' revisando, então isso é bom porque se eu já sei do que se trata, fica melhor de lembrar o que foi feito e saber quais são as necessidades do código para essa melhoria. Mas pontualmente, para cada um dos refactorings feitos é que, não sei se pelo número deles ou porque são similares, eu não parei para lembrar: Ah, esse nome eu já vi antes, mas já era mais confortável ter visto antes; já me deixa mais confortável de saber. Se eu vi um rename aqui, ah eu já estava esperando por isso."

Sobre a session #2: "como eu não tinha visto antes, eu não sabia muito bem o que esperar da revisão, então me deixou um pouco mais atenta sim, de verificar se era uma coisa pontual; talvez também tenha influenciado seja o fato de, apesar de saber do que

se trata a atividade em geral, eu não sabia do que se tratava esse commit específico. Então, isso também influenciou pra eu saber se: ah, tanto de eu saber do que tratava total, mas saber as particularidades do código, que algumas coisas eu não sabia. Então, isso me deixou um pouco mais atenta porque eu não tinha visto o registro de commit (*lista de refactorings aplicados no commit*) e não tinha muita ideia de toda a parte do código. A primeira revisão foi um pouco mais fluída."

- **Have reviewers understood the code changes' intention?**

Sim, o reviewer associou *Rename* a mudanças de padrão e ajustes na ortografia.

- **Have reviewers used the ePol project checklist for code review?**

Para o reviewer, o checklist é importante principalmente para os iniciantes em code review. Dada a sua experiência, o reviewer já incorporou o checklist à sua prática de code review.

- **Are reviewers familiar with the changeset's refactorings?**

Sim.

A.3. CODE REVIEW SESSIONS #3 AND #4

Na sessão #3, dos 5 refatoramentos aplicados, o reviewer verbalizou a detecção de 2 *Extract Method*.

- **#discussions**

Na session #4: 2 sugestões de melhoria para o código, 1 discussão relacionada à ortografia, 1 sugestão de remoção de um parâmetro e respectivo ajuste na documentação (inclusive, relacionado a um *Rename Method*) e 1 indicação de inclusão de documentação em um teste.

- **Is there any code change understanding difference between a refactoring-aware and a no refactoring-aware code review session?**

Sobre a session #4: "eu não abriria nenhuma issue relacionada a esses refatoramentos, porque foi mais mudança de nome de método e a questão da extração da classe ... eu não perguntaria porque ele fez essa nova classe... e mover o método foi consequência dele ter extraído a classe."

- **Have reviewers understood the code changes' intention?**

Sim, especialmente na session 4, porque o reviewer já conhecia a funcionalidade diretamente relacionada.

- **Have reviewers used the ePol project checklist for code review?**

Não.

- **Are reviewers familiar with the changeset's refactorings?**

Sim.