

Universidade Federal de Juiz de Fora  
Instituto de Ciências Exatas  
Programa de Pós-graduação em Ciência da Computação

**Vinicius Junqueira Schettino**

**Uma ferramenta para recomendação de revisores de código para apoiar a  
colaboração em Desenvolvimento Distribuído de Software**

Juiz de Fora

2018

**Vinicius Junqueira Schettino**

**Uma ferramenta para recomendação de revisores de código para apoiar a  
colaboração em Desenvolvimento Distribuído de Software**

Dissertação apresentada ao Programa de Pós-graduação em Ciência da Computação da Universidade Federal de Juiz de Fora, como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Orientador: Marco Antônio Pereira Araújo

Juiz de Fora

2018

Ficha catalográfica elaborada através do Modelo Latex do CDC da UFJF  
com os dados fornecidos pelo(a) autor(a)

Junqueira Schettino, Vinicius.

Uma ferramenta para recomendação de revisores de código para apoiar  
a colaboração em Desenvolvimento Distribuído de Software / Vinicius  
Junqueira Schettino. – 2018.

20 f.

Orientador: Marco Antônio Pereira Araújo

Dissertação (Mestrado) – Universidade Federal de Juiz de Fora, Instituto  
de Ciências Exatas. Programa de Pós-graduação em Ciência da Computação,  
2018.

1. Palavra-chave. 2. Palavra-chave. 3. Palavra-chave. I. Pereira Araújo,  
Marco Antônio, orient. II. Título.

**Vinicius Junqueira Schettino**

**Uma ferramenta para recomendação de revisores de código para apoiar a  
colaboração em Desenvolvimento Distribuído de Software**

Dissertação apresentada ao Programa de Pós-graduação em Ciência da Computação da Universidade Federal de Juiz de Fora, como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Aprovada em:

**BANCA EXAMINADORA**

---

Prof. Dr. Marco Antônio Pereira Araújo - Orientador  
Universidade Federal de Juiz de Fora

---

Professor Dr. ??  
Universidade ???

---

Professor Dr. ??  
Universidade ??

## **AGRADECIMENTOS**

De acordo com a Associação Brasileira de Normas Técnicas - 14724 (2011, p. 1) Agradecimentos é o “texto em que o autor faz agradecimentos dirigidos àqueles que contribuíram de maneira relevante à elaboração do trabalho.”

“Texto em que o autor apresenta uma citação, seguida de autoria, relacionada com a  
matéria tratada no corpo do trabalho”

(ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2011, p. 2)

A epígrafe elaborada conforme NBR 10520 (Epígrafe - Opcional)

## **RESUMO**

De acordo com a Associação Brasileira de Normas Técnicas - 6028 (2003, p. 2) “o resumo deve ressaltar o objetivo, método e as conclusões do documento (...) Deve ser composto de uma sequência de frases concisas, afirmativas e não de enumeração de tópicos. Recomenda-se o uso de parágrafo único.” O resumo deve ter de 150 a 500 palavras.

Palavras-chave: Palavra-chave. Palavra-chave. Palavra-chave.

## ABSTRACT

...

Key-words: ...



## LISTA DE ILUSTRAÇÕES

## **LISTA DE ABREVIATURAS E SIGLAS**

ABNT	Associação Brasileira de Normas Técnicas
UFJF	Universidade Federal de Juiz de Fora
IBGE	Instituto Brasileiro de Geografia e Estatística

## LISTA DE SÍMBOLOS

$\forall$  Para todo

$\in$  Pertence

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>11</b>
<b>2</b>	<b>TRABALHOS RELACIONADOS . . . . .</b>	<b>13</b>
2.1	Revisão sistemática da literatura . . . . .	13
2.2	Outros trabalhos relevantes . . . . .	13
<b>3</b>	<b>MÉTODOS E FERRAMENTAS . . . . .</b>	<b>14</b>
3.1	Ferramentas de <i>code review</i> . . . . .	14
3.2	Métricas relevantes para recomendação do revisor . . . . .	14
3.3	Métricas de avaliação dos resultados . . . . .	14
<b>4</b>	<b><i>CODE REVIEW</i> . . . . .</b>	<b>15</b>
4.1	Histórico . . . . .	15
4.2	Relevância . . . . .	15
4.3	Pull Based Method . . . . .	15
<b>5</b>	<b>SOLUÇÃO DESENVOLVIDA . . . . .</b>	<b>16</b>
<b>6</b>	<b>RESULTADOS . . . . .</b>	<b>17</b>
6.1	Apresentação dos resultados . . . . .	17
6.2	Discussão dos resultados . . . . .	17
<b>7</b>	<b>CONCLUSÃO . . . . .</b>	<b>18</b>
7.1	Ameaças . . . . .	18
7.2	Trabalhos futuros . . . . .	18
7.3	Considerações finais . . . . .	18
	<b>REFERÊNCIAS . . . . .</b>	<b>19</b>
	<b>APÊNDICE A – Artigo Mapeamento Sistemático . . . . .</b>	<b>20</b>

## 1 INTRODUÇÃO

O *code review* é considerado como uma das principais técnicas de diminuição de defeitos de software [1]. Nela, o autor de uma alteração na base de código de um projeto submete tal conteúdo ao crivo de um conjunto de pares técnicos, que irão revisar sua estrutura com base em uma lista de regras e convenções previamente definida. Diferentes aspectos relacionados ao autor, ao revisor e ao processo de revisão em si estão diretamente relacionados à eficiência da prática. Autores relatam relação da diminuição da incidência de *anti-patterns* [2] de acordo com o nível de participação dos envolvidos e cobertura do código revisado [3, 4, 5]. Reputação [6] e experiência [7] do revisor também parecem impactar nos efeitos do *code review*.

Intrinsecamente colaborativa, a atividade hoje é exercida com suporte de ferramentas computacionais específicas [6], principalmente no desenvolvimento distribuído. Dentro de workflows de trabalho descentralizados [8], a prática funciona como um *gateway* de qualidade que busca garantir que apenas alterações aderentes aos padrões de qualidade do projeto serão incorporados à codebase principal. Esta etapa do desenvolvimento se torna uma oportunidade para disseminação de conhecimento, embate de ideias e discussão de melhores práticas entre profissionais de experiência e visões diferentes.

Neste contexto, porém, os desafios à colaboração co-localizada são potencializados e as soluções tradicionais não são suficientes para fomentar este aspecto das atividades distribuídas [9]. Casey [10] mostra que, com a distribuição geográfica dos times, diversos outros desafios, antes considerados colaterais ou resolvidos, emergem de forma a ameaçar a colaboração entre os membros da equipe: barreiras culturais, temporais e geográficas; reengenharia dos processos de desenvolvimento; resistência em compartilhar informações e conhecimento com os pares distribuídos; entre outros desafios.

Estes desafios do desenvolvimento distribuído afetam o *code review* de duas formas distintas. Primeiro, o processo de revisão pode se tornar menos eficiente quando a colaboração é afetada, devido aos baixos níveis de participação e cobertura. O mesmo vale para a disseminação do conhecimento, que fica prejudicada. Outro desafio que se forma é a escolha do revisor adequado para aquele *patch*. Com um vasto número de opções e pouca informação disponível sobre seus aspectos técnicos e gerenciais (e.g. tempo disponível) já que não há contato co-localizado entre eles, a natureza distribuída deste tipo de desenvolvimento dificulta o processo de escolha do revisor, o que também pode impactar a eficiência do revisor.

Uma possível solução, visando amparar o desafio da colaboração e evitando o *overhead* da escolha do revisor, seria manter grupos bem testados e experientes exercendo as atividades de revisão. Ou ainda, fixar, dentro de cada equipe de desenvolvimento, quem são os responsáveis por revisão e pela submissão dos *patches*, evitando a diversificação das

relações de trabalho.

Contudo, estudos recentes demonstram que a fixação de grupos e responsabilidades pode não ser benéfica para o processo de desenvolvimento. Scott Page [11] argumenta que a diversidade de experiências, visões e especialidades fazem com que grupos sejam mais eficientes. Já Prikladnicki et al. [12] apontam indícios de que a formação de grupos temporários em detrimento ou em conjunto com permanentes é um fator de eficiência em projetos de software:

“Although old colleagues bring knowledge of the development process and prior norms from previous teams, new members bring fresh ideas that could promote project performance and creativity. Old colleagues might not do so and might not give new members a chance to implement their ideas.”

Expostas os desafios que o Desenvolvimento Distribuído de Software impõe sobre a escolha do revisor de código, a importância da escolha do revisor adequado do ponto de vista de colaboração e a motivação da formação de grupos heterogêneos e dinâmicos, expõe-se o objetivo deste trabalho. De acordo com a abordagem QGM (Goal/Question/Metric) proposta por Basili et al. [13]: **Desenvolver** um método de recomendação de revisores **com o objetivo de** potencializar a colaboração **em relação aos aspectos** de coordenação **do ponto de vista** de revisores e autores **no contexto de** desenvolvimento distribuído de software. A principal hipótese que norteia o andamento desta proposta, e que será revisitada e discutida nos capítulos derradeiros é:

- O método de recomendação apresentado pode potencializar a colaboração entre revisores e autores.

## **2 TRABALHOS RELACIONADOS**

2.1 Revisão sistemática da literatura

2.2 Outros trabalhos relevantes

### 3 MÉTODOS E FERRAMENTAS

#### 3.1 Ferramentas de *code review*

(justificar o porquê da escolha do GitHub)

#### 3.2 Métricas relevantes para recomendação do revisor

#### 3.3 Métricas de avaliação dos resultados



## **4 *CODE REVIEW***

### 4.1 Histórico

### 4.2 Relevância

### 4.3 Pull Based Method

(literalmente mostrar como funciona)

## **5 SOLUÇÃO DESENVOLVIDA**

(explicações técnicas, MER, tecnologias, etc)

## **6 RESULTADOS**

### 6.1 Apresentação dos resultados

### 6.2 Discussão dos resultados

## **7 CONCLUSÃO**

7.1 Ameaças

7.2 Trabalhos futuros

7.3 Considerações finais

## REFERÊNCIAS

- [1] BOEHM, B.; BASILI, V. R. Software defect reduction top 10 list. *Computer*, IEEE Computer Society Press, Los Alamitos, CA, USA, v. 34, n. 1, p. 135–137, 12 2001. ISSN 0018-9162.
- [2] KEMERER, C. F.; PAULK, M. C. The impact of design and code reviews on software quality: An empirical study based on psp data. *IEEE Transactions on Software Engineering*, v. 35, n. 4, p. 534–550, 07 2009. ISSN 0098-5589.
- [3] MENEELY, A. et al. An empirical investigation of socio-technical code review metrics and security vulnerabilities. In: . [S.l.: s.n.], 2014. p. 37–44.
- [4] MORALES, R.; MCINTOSH, S.; KHOMH, F. Do code review practices impact design quality? a case study of the qt, vtk, and itk projects. *2015 IEEE 22nd International Conference on Software Analysis, Evolution and Reengineering (SANER)*, IEEE Computer Society, Los Alamitos, CA, USA, v. 00, p. 171–180, 2015.
- [5] BAVOTA, G.; RUSSO, B. Four eyes are better than two: On the impact of code reviews on software quality. In: . [S.l.: s.n.], 2015. p. 81–90.
- [6] BAYSAL, O. et al. The influence of non-technical factors on code review. In: . [S.l.: s.n.], 2013. p. 122–131.
- [7] KONONENKO, O. et al. Investigating code review quality: Do people and participation matter? In: . [S.l.: s.n.], 2015. p. 111–120.
- [8] GOUSIOS, G.; STOREY, M.-A.; BACCHELLI, A. Work practices and challenges in pull-based development: The contributor’s perspective. In: IEEE. *Software Engineering (ICSE), 2016 IEEE/ACM 38th International Conference on*. [S.l.], 2016. p. 285–296.
- [9] COSTA, A. M. Nicolaci-da; PIMENTEL, M. Sistemas colaborativos para uma nova sociedade e um novo ser humano. *Sistemas colaborativos. PIMENTEL, M.; FUKS, H.(Orgs.). Rio de Janeiro: Elsevier*, 2011.
- [10] CASEY, V. Virtual software team project management. *Journal of the Brazilian Computer Society*, Springer, v. 16, n. 2, p. 83–96, 2010.
- [11] PAGE, S. E. *The difference: How the power of diversity creates better groups, firms, schools, and societies*. [S.l.]: Princeton University Press, 2008.
- [12] PRIKLADNICKI, R. et al. The best software development teams might be temporary. *IEEE Software*, v. 34, n. 2, p. 22–25, Mar 2017. ISSN 0740-7459.
- [13] BASILI, V. R.; WEISS, D. M. A methodology for collecting valid software engineering data. *IEEE Trans. Softw. Eng.*, SE-10, no. 6, p. 728–738, 1984.

## APÊNDICE A – Artigo Mapeamento Sistemático

Colocar aqui o artigo.