Survey do Processo de Revisão de Código

* Required

TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO (TCLE)

Esta pesquisa faz parte de um trabalho científico que está sendo desenvolvido na disciplina de Engenharia de Software durante a realização do Mestrado Profissional em Computação Aplicada na Universidade Tecnológica Federal do Paraná (UTFPR).

O objetivo desta pesquisa é descobrir problemas que são gerados durante o desenvolvimento de software e são encontrados durante o processo de revisão de código. Não é objetivo da pesquisa identificar nomes ou equipes que geram problemas durante o desenvolvimento de software.

Revisão de código (ou do inglês "code review") é o ato de conscientemente e sistematicamente verificar o código fonte de outros desenvolvedores em busca de erros. Tal prática tem se mostrado eficiente para acelerar e agilizar o processo de desenvolvimento de software como poucas outras práticas podem.

Respondendo a opção "Concordo" estou ciente de que:
01) O questionário possui 9 questões.
02) Minha participação é totalmente voluntária.
03) Não haverá divulgação de dados pessoais.
04) Esta pesquisa não tem vínculo empresarial.
05) Ao finalizar o questionário poderei ver todas as minhas respostas.
06) O tempo esperado para responder esta pesquisa varia de 5 a 10 minutos.
07) As respostas desta pesquisa serão utilizados apenas para fins acadêmicos.
08) A obrigatoriedade de uso da Conta Google serve apenas para evitar respostas duplicadas.
09) Posso me retirar da pesquisa a qualquer momento e sem qualquer tipo de prejuízo a minha pessoa.
10) A resposta a estes instrumentos/procedimentos não causam riscos conhecidos à minha saúde física e mental.
11) Este Termo de Consentimento, apesar de ser respondido virtualmente, poderá ser solicitado ao pesquisador, que
enviará uma cópia assinada do mesmo por e-mail.
12) Poderei entrar em contato com o responsável pelo estudo, Altieres de Matos, sempre que julgar necessário pelo telefone (44) 99931 8326 ou pelo e-mail altitdb@gmail.com.

1. **Posterior a leitura do TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO declaro estar? ***
   Mark only one oval.

   - ◯ Ciente, concordo e seguirei para as próximas etapas
   - ◯ Ciente, porém não concordo com os termos de participação do mesmo e não seguirei para as próximas etapas *Stop filling out this form.*

2. **1. Qual sua função? ***
   Mark only one oval.

   - ◯ Desenvolvedor
   - ◯ Arquiteto
   - ◯ Other: _____

3. **2. Você faz revisão de código?** *
   Mark only one oval.
   - Sim
   - Não
   - Talvez

3. Quais problemas são frequentemente encontrados durante a revisão de código?

1 - Nunca
2 - Raramente
3 - Algumas vezes
4 - Frequentemente
5 - Sempre

4. **Código fonte com defeitos** *
   O termo defeito refere-se geralmente a algum problema com o software, quer com o seu comportamento externo quer com as suas características internas. Exemplo: Tentativa de login com usuário e senha corretos porém acesso não é permitido por erro no banco de dados. Contra-exemplo: Tentativa de login com usuário e senha corretos e com acesso permitido.
   Mark only one oval.

	1	2	3	4	5	
Nunca	◯	◯	◯	◯	◯	Sempre

5. **Código fonte com code smells** *
   Code smells geralmente não são defeitos. Eles não são tecnicamente incorretos e não impedem o funcionamento do software. Em vez disso, eles indicam fraquezas no projeto que podem estar atrasando o desenvolvimento ou aumentando o risco de erros no futuro. Code smell pode ser um indicador de fatores que contribuem para a dívida técnica. Exemplo: Classe com um método privado não utilizado e software funcionando corretamente. Contra-exemplo: Classe com todos os métodos privados sendo utilizados e software funcionando corretamente.
   Mark only one oval.

	1	2	3	4	5	
Nunca	◯	◯	◯	◯	◯	Sempre

6. **Código fonte com complexidade ciclomática alta** *
   A complexidade ciclomática é uma métrica de software responsável por medir a quantidade de caminhos de execução independentes a partir de um código fonte. Exemplo: Código fonte com muitos if, switch, for, while. Contra-exemplo: Código com poucos ou sem if, switch, for, while.
   Mark only one oval.

	1	2	3	4	5	
Nunca	◯	◯	◯	◯	◯	Sempre

7. **Código fonte com design ruim** *
   O design geralmente é ruim quando seus objetos não são bem definidos, seus métodos ou procedimentos são extensos e complexos, e o código fonte não favorece a manutenibilidade. Exemplo: Objetos com mais de uma responsabilidade e métodos/procedimentos com muitas instruções. Contra-exemplo: Objetos somente com uma responsabilidade e métodos/procedimentos simples.
   Mark only one oval.

	1	2	3	4	5	
Nunca	◯	◯	◯	◯	◯	Sempre

8. **Código fonte com duplicação de código** *
   Código fonte duplicado significa dois ou mais fragmentos de código multi-linha que são idênticos ou semelhantes, particularmente em sua estrutura. Exemplo: Código fonte com várias instruções iguais duplicadas em vários métodos/procedimentos. Contra-exemplo: Código fonte sem instruções duplicadas.
   Mark only one oval.

	1	2	3	4	5	
Nunca	◯	◯	◯	◯	◯	Sempre

9. **Código fonte fora do padrão da empresa** *
   Requisito empregado para estabelecer uma abordagem disciplinada e uniforme para fornecer consistência em um produto de software, ou seja, um padrão ou forma uniforme para organização. Exemplo: Nomenclatura de objetos diferente do documento de padronização da empresa. Contra-exemplo: Nomenclatura de objetos de acordo com o documento de padronização da empresa.
   Mark only one oval.

	1	2	3	4	5	
Nunca	◯	◯	◯	◯	◯	Sempre

10. **Código fonte com layout errado** *
    A organização do código fonte incluindo o uso de espaços em branco, agrupamento de código, linhas em branco, alinhamento, indentação, entre outros. Exemplo: Código fonte sem indentação e sem agrupamento. Contra-exemplo: Código fonte com indentação e com agrupamento.
    Mark only one oval.

	1	2	3	4	5	
Nunca	◯	◯	◯	◯	◯	Sempre

11. **Código fonte não atende o requisito** *
    Um requisito é uma condição ou capacidade que deve ser cumprida ou possuída por um sistema, componente do sistema, produto ou serviço para satisfazer um acordo, padrão, especificação ou outros documentos formalmente impostos. Exemplo: Código fonte que multiplica dois valores e requisito solicita a soma de dois valores. Contra-exemplo: Código fonte que soma dois valores e requisito solicita a soma de dois valores.
    Mark only one oval.

	1	2	3	4	5	
Nunca	◯	◯	◯	◯	◯	Sempre

12. **Código fonte sem testes automáticos** *

Quando um teste exercita um pedaço de código, dizemos que tal pedaço de código é coberto de testes. No entanto existem cenários que nenhum pedaço de código é exercitado pelos testes. Exemplo: Dado um objeto com três métodos/procedimentos e não existe classe com testes automáticos para realizar a validação dos métodos/procedimentos. Contra-exemplo: Dado um objeto com três métodos/procedimentos e uma classe com pelo menos um teste automático para realizar a validação dos métodos/procedimentos.
Mark only one oval.

	1	2	3	4	5	
Nunca	◯	◯	◯	◯	◯	Sempre

13. **Código fonte com testes automáticos errados** *

Compreende-se como teste automático errado um teste executado em condições especificadas e os resultados que são validados não são iguais aos resultados definidos nos requisitos do software. Exemplo: Teste automático que realiza a validação da multiplicação entre X e Y, e requisito solicita a soma entre X e Y. Contra-exemplo: Teste automático que realiza a validação da soma entre X e Y, e requisito solicita a soma entre X e Y.
Mark only one oval.

	1	2	3	4	5	
Nunca	◯	◯	◯	◯	◯	Sempre

14. **Código fonte com baixa cobertura de testes automáticos** *

A cobertura de código também é usada como uma medida de qualidade do produto. Uma equipe pode estabelecer um limite de cobertura de 80% como seu mínimo, o que significa que eles não liberarão um produto a menos que o tenham um mínimo de 80% de cobertura. Exemplo: Teste automático somente do cenário feliz, e cobrindo 10% da funcionalidade. Contra-exemplo: Testes automáticos para cenários de erro e de sucesso, e cobrindo 90% da funcionalidade.
Mark only one oval.

	1	2	3	4	5	
Nunca	◯	◯	◯	◯	◯	Sempre

15. **4. Existem problemas que são frequentemente encontrados durante a revisão de código e não foram citados? Cite esses problemas, descrevendo-o brevemente e informando quão frequentemente eles são encontrados.**

16. **5. Quais são os problemas mais graves encontrados durante a revisão de código?** *

Você pode escolher de 1 a 5 alternativas.
Check all that apply.

- [ ] Código fonte com baixa cobertura de testes automáticos
- [ ] Código fonte com code smells
- [ ] Código fonte com complexidade ciclomática alta
- [ ] Código fonte com defeitos
- [ ] Código fonte com design ruim
- [ ] Código fonte com duplicação de código
- [ ] Código fonte com layout errado
- [ ] Código fonte com testes automáticos errados
- [ ] Código fonte fora do padrão da empresa
- [ ] Código fonte não atende o requisito
- [ ] Código fonte sem testes automáticos
- [ ] Other: _____

17. **6. Quais os problemas levam mais tempo para serem encontrados na revisão de código?** *

Você pode escolher de 1 a 5 alternativas.
Check all that apply.

- [ ] Código fonte com baixa cobertura de testes automáticos
- [ ] Código fonte com code smells
- [ ] Código fonte com complexidade ciclomática alta
- [ ] Código fonte com defeitos
- [ ] Código fonte com design ruim
- [ ] Código fonte com duplicação de código
- [ ] Código fonte com layout errado
- [ ] Código fonte com testes automáticos errados
- [ ] Código fonte fora do padrão da empresa
- [ ] Código fonte não atende o requisito
- [ ] Código fonte sem testes automáticos
- [ ] Other: _____

18. **7. Quais itens reportados na revisão de código demandam mais esforço para correção?** *

Você pode escolher de 1 a 5 alternativas.
Check all that apply.

- [ ] Código fonte com baixa cobertura de testes automáticos
- [ ] Código fonte com code smells
- [ ] Código fonte com complexidade ciclomática alta
- [ ] Código fonte com defeitos
- [ ] Código fonte com design ruim
- [ ] Código fonte com duplicação de código
- [ ] Código fonte com layout errado
- [ ] Código fonte com testes automáticos errados
- [ ] Código fonte fora do padrão da empresa
- [ ] Código fonte não atende o requisito
- [ ] Código fonte sem testes automáticos
- [ ] Other: _____

19. **8. Quais técnicas, práticas, plugins e/ou ferramentas você utiliza para reduzir problemas na revisão de código?**

_____

_____

_____

_____

_____

20. **9. Caso você tenha alguma sugestão ou queira complementar suas respostas com algo que o questionário não abordou fique a vontade para descrever neste espaço.**

_____

_____

_____

_____

_____