



**Universidade Federal de Pernambuco
Centro de Informática
Graduação em Ciência da Computação**

Uma comparação entre métodos para análise do risco de conflitos de merge em tarefas de desenvolvimento

Proposta de Trabalho de Graduação

Aluno: João Pedro de Medeiros Santos

Orientador: Paulo Henrique Monteiro Borba

Recife

Agosto de 2018

Sumário

Contexto	2
Objetivo	3
Cronograma	4
Possíveis avaliadores	5
Referências	6
Assinaturas	7

Contexto

A utilização de sistemas de controle, como o Git e Subversion, é amplamente difundida no ambiente de desenvolvimento de software. Essas ferramentas permitem a seus usuários editar o código fonte do programa de forma que exista um gerenciamento dessas edições. Ao publicar uma mudança é possível que haja conflito entre versões, ou seja, uma mesma linha de um documento foi alterada de forma diferente por dois ou mais desenvolvedores, tornando a mesclagem das edições impossível até que o impasse seja resolvido.

Conflitos de mesclagem, ou *merge*, são frequentes e consomem tempo para serem resolvidos [1]. Por esse motivo, abordagens que tentam diminuir o número de conflitos, como algumas práticas de desenvolvimento tais quais publicações frequentes e algumas ferramentas como o *merge* estruturado e semi-estruturado foram desenvolvidas. Sabendo que, entre 24% e 46% de todos os *merges* resultam em conflito [1], e que alguns desenvolvedores têm medo de encontrar um durante o desenvolvimento, aumentando as chances do mesmo acontecer [2], conhecer uma forma de evitar ou prevenir alguns desses conflitos é essencial para aumentar a produtividade no ambiente de trabalho.

Objetivo

O objetivo deste trabalho é comparar o desempenho de uma ferramenta que alerta previamente os possíveis riscos de conflito de *merge* atrelados a escolha de tarefas num ambiente de desenvolvimento de *software* ao desempenho dessa mesma ferramenta utilizando novos métodos.

Para que essa comparação seja possível é necessário desenvolver os novos métodos utilizados. A ferramenta de detecção de riscos atual abrange as tarefas de acordo com os testes previamente concebidos, seguindo a técnica de desenvolvimento guiado por testes(TDD), o primeiro método adicional utilizará não só os testes mas também as dependências dos arquivos que estes exercitam. A segunda mudança consistirá em processar as interfaces a nível de método ao invés da classe inteira, como é feito atualmente pela ferramenta.

Cronograma

Atividade	Fevereiro	Março	Abril	Maiο	Junho
Definição do escopo					
Análise da literatura					
Implementação da ferramenta					
Análise de resultados					
Desenvolvimento do relatório					
Apresentação					

Possíveis avaliadores

Hermano Perrelli de Moura
Leopoldo Motta Teixeira

Referências

- [1] T. Zimmermann. Mining Workspace Updates in CVS. In Proceedings of the International Workshop on Mining Software Repositories (MSR), page 11. IEEE CS, 2007.
- [2] DYuriy Brun, Reid Holmes, Michael D. Ernst, and David Notkin. Early detection of collaboration conflicts and risks. IEEE Transactions on Software Engineering, 39(10):1358–1375, October 2013.

Assinaturas

Aluno: **João Pedro de Medeiros Santos**

Orientador: **Paulo Henrique Monteiro Borba**