



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

Extração de medida de modularidade em projetos Android para predição de conflitos de merge

Proposta de Trabalho de Graduação

Aluno: **Danilo Lima Ribeiro**

Orientador: **Leopoldo Motta Teixeira**

Recife

Agosto de 2018

Sumário

Sumário	1
Contexto	2
Objetivo	3
Metodologia	4
Cronograma	5
Possíveis avaliadores	6
Referências	7
Assinaturas	8

Contexto

O aumento da complexidade e tamanho das equipes em desenvolvimento de software tornou necessário a utilização de sistemas de controle de versão distribuídos, em que os desenvolvedores trabalham isoladamente nas suas *branches* e repositórios locais, e quando necessário, enviam sua contribuição para o repositório central.

Ao enviar as contribuições para uma outra *branch*, seja ela a central ou não, é possível que ocorra um conflito de *merge*. Conflitos de *merge* ocorrem quando o sistema de controle de versão tenta aplicar a contribuição de uma *branch* na *branch* alvo e ocorreram mudanças na mesma área textual ou no mesmo arquivo. Estes, então, devem ser resolvidos por algum desenvolvedor da equipe, que deve tomar decisões sobre como lidar com o conflito da melhor maneira, exigindo esforço e tempo dele sem contribuir para o resultado final do projeto, e a resolução podendo levar a novos erros (BRUN; HOLMES; ERNEST; NOTKIN, 2013).

Para reduzir ou evitar os custos decorridos dos conflitos de *merge*, pesquisas estão sendo feitas para construir métodos de predição de conflitos antes que tais problemas ocorram. Esta predição é feita com base em medidas temporais, escalares e de modularidade das contribuições, levando em consideração os *frameworks* e tecnologias adotados nos projetos. Preditores de conflito construídos com base nessas informações podem auxiliar na gerência de projetos e à organizar o trabalho em equipe.

Entretanto, não há estudos sobre predição de conflitos em projetos de software para dispositivos da plataforma Android, pois a medida de modularidade depende da tecnologia utilizada para desenvolver o projeto. Por exemplo, projetos construídos utilizando *frameworks* como Ruby on Rails ou Django, podem ter modularidade facilmente definida pela utilização do padrão Modelo-Visão-Controlador como as classes de modelo, mas para a plataforma Android, não há padrões claramente definidos que são exigidos para a sua utilização.

Objetivo

Esse trabalho tem por objetivo implementar um extrator de modularidade para projetos Android, usando análise estática de código, a fim de utilizar os dados extraídos para a construção de preditores de conflito de *merge* e construir uma ferramenta que dado uma coleção de projetos androids construa os preditores de automaticamente.

Metodologia

Será criada uma ferramenta capaz de construir preditores de conflitos de merge para projetos para a plataforma Android, utilizando modelos estatísticos desenvolvidos pela doutoranda Klissiomara Lopes Dias, e o extrator de modularidade desenvolvido por mim, a fim de facilitar o gerenciamento dos projetos. Ela será construída usando a linguagem de programação Java, e a ferramenta de build Gradle, usando a biblioteca JavaParser para análise dos códigos java.

Cronograma

Atividade	Agosto			Setembro				Outubro				Novembro				Dezembro			
Elaboração da proposta																			
Levantamento bibliográfico																			
Implementação do extrator																			
Execução de experimentos																			
Escrita da documentação e relatório																			
Preparação e apresentação da defesa																			
Correção e entrega da versão final																			

Possíveis avaliadores

Paulo Henrique Monteiro Borba

Fernando Henrique Castor

Referências

- [1] Christian Bird and Thomas Zimmermann. 2012. Assessing the value of branches with what-if analysis. In Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering. ACM, 45.
- [2] H Christian Estler, Martin Nordio, Carlo A Furia, and Bertrand Meyer. 2014. Awareness and merge conflicts in distributed software development. In Global Software Engineering (ICGSE), 2014 IEEE 9th International Conference on. IEEE, 26–35.
- [3] J. Garcia, I. Ivkovic and N. Medvidovic, "A comparative analysis of software architecture recovery techniques," *2013 28th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, Silicon Valley, CA, 2013, pp. 486-496.
- [4] M. Bauer and M. Trifu, "Architecture-aware adaptive clustering of OO systems," *Eighth European Conference on Software Maintenance and Reengineering, 2004. CSMR 2004. Proceedings.*, Tampere, Finland, 2004, pp. 3-14.
- [5] Yuriy Brun, Reid Holmes, Michael D Ernst, and David Notkin. 2013. Early detection of collaboration conflicts and risks. *IEEE Transactions on Software Engineering* 39, 10 (2013), 1358–1375.

Assinaturas

Aluno: **Danilo Lima Ribeiro**

Orientador: **Leopoldo Motta Teixeira**

