

Aluno: José Teodoro da Silva

## 1. Tema

Avaliação dos métodos existentes na literatura para a identificação de desenvolvedores-chave em projetos de software livre.

## 2. Introdução

Alguns estudos caracterizam a divisão em camadas para representar os papéis existentes em comunidades de software livre. Entre eles, o modelo OnionPatch [5], descreve esses papéis como sendo “camadas de uma cebola”. As partes mais externas representam desenvolvedores menos experientes no projeto. À medida que esses desenvolvedores participam do projeto, eles podem migrar para papéis mais centrais no modelo (e no projeto). No entanto, estes limites entre as camadas do modelo OnionPatch não são intuitivos[2]. Não é prático utilizar os papéis existentes nos *sites* e na documentação do projeto para identificação dos desenvolvedores-chave. Em muitos casos, os desenvolvedores já possuem as características e atribuições antes mesmo de serem formalmente reconhecidos como tal [2]. A relação entre o grupo de desenvolvedores-chave e periferia (camadas mais externas) pode diagnosticar a maturidade do time que desenvolve o software [2]. Além disso, os desenvolvedores-chave são importantes para manter a comunidade unida [11] e são eles que tomam decisões importantes que podem definir o sucesso do projeto [12]. Terceiro (2010) encontrou evidências da influência dos desenvolvedores-chave para manutenção da integridade do projeto e destaca a importância de manter um grupo saudável para realizar essa tarefa [10].

Na literatura existem diferentes métodos para encontrar desenvolvedores-chave de um projeto. Entretanto, a escolha de um método ou a comparação entre eles nem sempre é discutida. Estudos são realizados sobre o conjunto dos desenvolvedores-chaves sem identificar ou apontar na literatura uma razão para escolha do método de identificação desse conjunto [3],[1],[8],[14],[13],[4]. Isto torna difícil a tarefa de escolha de um dos métodos já existentes. Crowston destaca a necessidade da realização de uma comparação entre os vários métodos de identificação de membros-chaves utilizando uma grande quantidade de projetos [2]. Lakhani aponta a dificuldade de generalizar o método de identificação de desenvolvedores devido às diferentes características das comunidades de cada projeto open source [7].

## 3. Objetivo

Facilitar a escolha de métodos para identificação de desenvolvedores-chave pela comparação entre os métodos existentes na literatura e as características dos projetos da Apache Software Foundation (ASF).

#### 4. Objetivos específicos

- a) Comparar os requisitos (dados necessários) e dificuldades de execução de cada método no contexto dos projetos Java da Apache Software Foundation.
- b) Identificar características dos projetos que podem inviabilizar o uso de determinados métodos.
- c) Avaliar a viabilidade de execução de cada método considerando as características do método e as características dos projetos da Apache Software Foundation.
- d) Comparar a similaridade de resultados entre os métodos utilizando a distância euclidiana sobre os conjuntos resultantes de cada um.

#### 5. Design da Pesquisa

Será realizado um experimento para avaliar os vários modos de identificação de desenvolvedores chaves. Esse experimento utilizará o ecossistema dos projetos de software livre da ASF. A ASF foi escolhida pela disponibilidade dos dados para acesso público e pelo número de projetos que possui. A metodologia baseada em experimento foi escolhida devido à natureza quantitativa da comparação entre os resultados dos métodos de identificação a serem estudados. Ao contrário de um benchmark em que são realizadas vários experimentos para quantificar qual o melhor método, esse experimento avalia as características específicas dos projetos e dos métodos para identificar métodos semelhantes que sejam mais eficazes e condizentes com o contexto estudado. Não procuramos o melhor método, procuramos o mais adequado para cada contexto dos projetos.

#### 6. Método da Pesquisa

- a) Levantar os requisitos necessários para cada método encontrado na revisão bibliográfica;
- b) Avaliar as características dos projetos da Apache que possam inviabilizar algum dos métodos; A viabilidade de cada método será medida levando em consideração os dados necessários, a linguagem utilizada (se o método for específico para uma dada linguagem) e ao acesso à ferramenta capaz de executar o método;
- c) Obter dos dados necessários para execução dos métodos passíveis de execução;
- d) Comparar a similaridade dos resultados de cada método utilizando a distância euclidiana;
- e) Avaliar a complexidade de execução entre métodos que obtenham resultados semelhantes. Essa complexidade será medida pela disponibilidade de recursos necessários à execução de cada método. A semelhança entre métodos será avaliada pela distância euclidiana entre os conjuntos de desenvolvedores resultantes de cada método.

## 7. Resultado esperados

Espera-se encontrar métodos que sejam mais adequados que outros para as diferentes características dos projetos.

## 8. Dificuldades e desafios

O levantamento dos pontos fracos e fortes de cada método empregado;

A obtenção dos requisitos necessários para a execução de cada método;

O esforço de execução dos métodos que exigirem mais recursos para que se alcance os resultados.

## 9. Revisão da Literatura

Foi conduzido um mapeamento sistemático para identificar os trabalhos relacionados à caracterização e identificação dos desenvolvedores-chaves em projetos de software livre. O processo para revisão da literatura foi baseado no processo sugerido por Kitchenham [6] para planejamento e realização de revisões sistemáticas.

### 9.1. Introdução

Existem vários trabalhos que realizam a caracterização de desenvolvedores-chaves em projetos de softwares livres. Esses estudos utilizam diversos modos para distinguir os desenvolvedores. De modo geral, a escolha do método empregado é realizada arbitrariamente. Esse mapeamento sistemático objetivou levantar os diferentes métodos empregados para caracterização dos desenvolvedores-chaves nestes estudos.

### 9.2. Planejamento

O planejamento foi baseado no modelo proposto por [6]. Nesta seção serão apresentados os principais pontos do plano de execução do mapeamento sistemático.

### 9.3. Busca e seleção de Estudos Primários

A busca e seleção de estudos primários está definida de acordo com as palavras-chaves de busca, as fontes escolhidas e critérios de inclusão/exclusão.

Na seleção das fontes foram escolhidas aquelas acessíveis via internet tanto para busca quanto para obtenção dos trabalhos, mas fontes muito genéricas foram evitadas devido ao excesso de ruído que possuem em seus resultados. Dentre as fontes disponíveis foram selecionadas: IEEE Xplore, ACM Digital Library e Scopus.

As palavras chaves utilizadas para as buscas foram: ( (core\*) OR (leader\*) OR (key) OR (peripheral) ) AND (developer).

### 9.3.1. Critérios para seleção dos resultados

Para serem incluídas as publicações devem: advir de Livros, Journals, Revistas, Conferências ou Congressos; realizar a identificação de desenvolvedores/membros chaves; ter como algo projetos de software livre; ser publicada entre o ano 2000 e junho de 2013.

Serão excluídas as publicações que: não deixarem explícitos os projetos analisados; não seja possível recuperar a publicação integralmente via internet.

### 9.4. Seleção de resultados

A seleção de resultados foi realizada em duas iterações:

- na primeira iteração as publicações foram selecionadas por título. Pela seleção baseada nos títulos foram excluídas as publicações que tratavam de assuntos tais como redes de computadores e hardware;
- na segunda iteração o critério de seleção foi o abstract. Foram excluídas as publicações que não tratavam de caracterização ou identificação de desenvolvedores chaves em projetos de software livre.

### 9.5. Adição de publicações pertinentes

Algumas das publicações selecionadas citavam métodos de outros trabalhos para a identificação dos desenvolvedores chaves. Essas citações foram adicionadas na revisão bibliográfica por apresentarem a descrição dos métodos utilizados nos trabalhos selecionados. Também foram adicionadas teses citadas que embasaram o método de identificação de desenvolvedores nos trabalhos selecionados nas buscas.

Este trabalho teve por motivação inicial o estudo exploratório foi executado por Oliva et al [9]. Deste modo, os trabalhos relacionados em [9] que motivaram o estudo sobre a caracterização e identificação dos desenvolvedores chaves foram adicionadas nesta revisão bibliográfica.

### 9.6. Resultados

Quarenta artigos na literatura utilizaram a identificação de membros chaves em projetos de software livre. Além destes foram adicionados:

- duas publicações que descrevem os métodos utilizados;
- uma tese de doutorado que utiliza a identificação de desenvolvedores chaves;
- uma tese de doutorado que consta como motivação do estudo exploratório [9] que originou essa pesquisa.

## 10.Referências:

- [1] Chintan Amrit and Jos Van Hillegersberg. Detecting coordination problems in collaborative software development environments. *Information Systems Management*, 25(1):57–70, 2008.
- [2] Kevin Crowston, Kangning Wei, Qing Li, and James Howison. Core and periphery in free/libre and open source software team communications. In *System Sciences, 2006. HICSS'06. Proceedings of the 39th Annual Hawaii International Conference on*, volume 6, pages 118a–118a. IEEE, 2006.
- [3] Anja Guzzi, Alberto Bacchelli, Michele Lanza, Martin Pinzger, and Arie van Deursen. Communication in open source software development mailing lists. In *Proceedings of the 10th Working Conference on Mining Software Repositories*, MSR '13, pages 277–286, Piscataway, NJ, USA, 2013. IEEE Press.
- [4] Chris Jensen and Walt Scacchi. Role migration and advancement processes in ossd projects: A comparative case study. In *Proceedings of the 29th international conference on Software Engineering*, ICSE '07, pages 364–374, Washington, DC, USA, 2007. IEEE Computer Society.
- [5] Corey Jergensen, Anita Sarma, and Patrick Wagstrom. The onion patch: migration in open source ecosystems. In *Proceedings of the 19th ACM SIGSOFT symposium and the 13th European conference on Foundations of software engineering*, ESEC/FSE '11, pages 70–80, New York, NY, USA, 2011. ACM.
- [6] Barbara Kitchenham. Procedures for performing systematic reviews. *Keele, UK, Keele University*, 33:2004, 2004.
- [7] Karim R Lakhani. *The core and the periphery in distributed and self-organizing innovation systems*. PhD thesis, Massachusetts Institute of Technology, 2006.
- [8] Kumiyo Nakakoji, Yasuhiro Yamamoto, Yoshiyuki Nishinaka, Kouichi Kishida, and Yunwen Ye. Evolution patterns of open-source software systems and communities. In *Proceedings of the international workshop on Principles of software evolution*, pages 76–85. ACM, 2002.
- [9] Gustavo A Oliva, Francisco W Santana, Kleverton CM de Oliveira, Cleidson RB de Souza, and Marco A Gerosa. Characterizing key developers: a case study with apache ant. In *Collaboration and Technology*, pages 97–112. Springer, 2012.
- [10] Antonio Terceiro, Luiz Romario Rios, and Christina Chavez. An empirical study on the structural complexity introduced by core and peripheral developers in free software projects. In *Proceedings of the 2010 Brazilian Symposium on Software Engineering*, SBES '10, pages 21–29, Washington, DC, USA, 2010. IEEE Computer Society.
- [11] Jin Xu, Yongqin Gao, Scott Christley, and Gregory Madey. A topological analysis of the open souce software development community. In *System Sciences*,

2005. *HICSS'05. Proceedings of the 38th Annual Hawaii International Conference on*, pages 198a–198a. IEEE, 2005.

[12] Wen Zhang, Ye Yang, and Qing Wang. An empirical study on identifying core developers using network analysis. In *Proceedings of the 2nd international workshop on Evidential assessment of software technologies*, EAST '12, pages 43–48, New York, NY, USA, 2012. ACM.

[13] Minghui Zhou and Audris Mockus. Growth of newcomer competence: challenges of globalization. In *Proceedings of the FSE/SDP workshop on Future of software engineering research*, FoSER '10, pages 443–448, New York, NY, USA, 2010. ACM.

[14] Minghui Zhou and Audris Mockus. What make long term contributors: willingness and opportunity in oss community. In *Proceedings of the 2012 International Conference on Software Engineering*, ICSE 2012, pages 518–528, Piscataway, NJ, USA, 2012. IEEE Press.