

Coordenação de colaboradores em desenvolvimento de software *crowdsourced*

Vitor Queiroz de Campos¹, José Maria N. David¹

¹Departamento de Ciência da Computação – Universidade Federal de Juiz de Fora (UFJF)
Juiz de Fora – MG – Brasil

vitor.queiroz@ice.ufjf.br, jose.david@ufjf.edu.br

Abstract. *Coordination support of crowdsourced teams is a challenge in Global Software Development environments. Understanding the already proposed barriers and solutions for coordination of crowdsourced teams, it turns out that a service that uses the project repository itself for helping to coordinate a crowdsourced team is needed. We propose a tool that helps the analysis of three main aspects of the collaborators by project managers: 1) collaborators most related to issues, 2) the commits evolution over time, and 3) the contribution factor generated by the codehosting platform.*

Resumo. *O suporte à coordenação de equipes "crowdsourced" é um desafio em ambientes de Desenvolvimento Global de Software. Compreendendo as barreiras e soluções já propostas para a coordenação de equipes "crowdsourced", verifica-se a necessidade de um serviço que utiliza o repositório do próprio projeto para ajudar a coordenar um time "crowdsourced". Propomos uma ferramenta que auxilia a análise de três aspectos principais dos colaboradores por gerentes de projetos: 1) colaboradores mais relacionados a "issue", 2) a evolução dos "commits" ao longo do tempo e 3) o fator de contribuição gerado pela plataforma de "codehosting".*

1. Introdução

No contexto de Engenharia de Software, o desenvolvimento global de software trata da construção de software considerando o suporte de especialistas em diferentes países. É empregado quando uma instituição deseja utilizar o conhecimento de outros especialistas visando a melhoria da qualidade do produto final.

Crowdsourcing é utilizado na produção de software de maneira a solucionar problemas que instituições ou pessoas físicas apresentam, através de plataformas específicas. Neste contexto, o *crowdsourcer* - nome dado a quem cria a tarefa nas plataformas - abre um chamado público e pessoas geograficamente distribuídas podem dar soluções para o problema no universo do desenvolvimento de Software. A melhor solução pode ser recompensada e empregada no projeto. [Stol et al. 2017]

Pesquisas realizadas ([Stol and Fitzgerald 2014] e [Mao et al. 2017]), constataram a necessidade de uma análise mais profunda sobre a coordenação dos envolvidos em um projeto que utiliza *crowdsourcing*. Uma vez que os chamados são globais, uma multidão reage e tarefas de um mesmo projeto podem envolver diferentes colaboradores, o que gera uma necessidade de coordenar os mesmos. Sendo assim, o objetivo principal desta pesquisa em andamento é desenvolver um serviço que possa auxiliar a coordenação

de colaboradores em situações que utilizam *crowdsourcing*. Dados de repositórios online, que possam posicionar o *crowdsourcer* sobre fatores, como produtividade e assiduidade dos colaboradores de um projeto, podem ser obtidos. Como resultado, informações sobre o desenvolvimento dos especialistas por ele contratado podem apoiar o projeto.

Inicialmente, para compreender melhor o que tem sido proposto e pesquisado na área, desenvolvemos um Mapeamento Sistemático que tinha como objetivo compreender os desafios e soluções propostas à coordenação de colaboradores em um ambiente de desenvolvimento global de software que utilizasse *crowdsourcing*. Este mapeamento foi conduzido levando em consideração as atividades propostas por Kitchenham: planejamento, condução e publicação do estudo [Kitchenham and Charters 2007].

De um universo de 532 estudos primários retornados de bases bibliográficas e passando por etapas de exclusão de estudos não relacionados ao tema em pesquisa, foram identificados 19 estudos principais que abordam as principais barreiras enfrentadas na coordenação de desenvolvimento *crowdsourced* e também as soluções às barreiras já propostas por pesquisadores da área. Como resultado do Mapeamento, algumas barreiras foram identificadas, as quais apoiaram a construção da proposta de solução tais como a complexidade de coordenação, a coordenação de micro tarefas e o gerenciamento de motivação de colaboradores. Por limitações de espaço, este mapeamento não está no escopo deste artigo.

Deste mapeamento sistemático, foi observada a ausência de um serviço que auxiliasse a coordenação de especialistas de um projeto que utilizasse de *crowdsourcing* para a solução de problemas específicos e que também utilizasse diretamente repositórios abertos para a análise de fatores que auxiliassem na tarefa de coordenação dos colaboradores deste projeto. Ao compreender as barreiras mais frequentes, como as citadas anteriormente, através do mapeamento, tornou-se mais claro que a empregabilidade de um serviço como o descrito poderia apoiar desenvolvedores que utilizam de *crowdsourcing* em seus projetos.

O presente documento está organizado da seguinte forma: na seção 2 apresentamos trabalhos relacionados, na seção 3 comentamos como o serviço foi desenvolvido e mostramos seu funcionamento atual, na seção 4 descrevemos alguns dos desafios que tivemos ao longo do projeto e na seção 5 concluímos e comentamos sobre os trabalhos futuros do projeto.

2. Trabalhos Relacionados

Após a condução do Mapeamento Sistemático, foi possível a compreensão de trabalhos relacionados. [Zhou et al. 2017] correlacionam a performance de times *crowdsourced* com fatores de formação de equipes. Entretanto, apesar de ressaltar a importância da coordenação na atividade de *crowdsourcing* no desenvolvimento de software, não são propostas soluções para este aspecto.

[Won Lee et al. 2017] propõem uma abordagem de auto-coordenação através de elementos de design de ambientes de programação compartilhados. Contudo, não considera a análise de estratégias para o gerenciamento de times por um coordenador de maneira sistêmica.

[Li et al. 2018] avaliam a qualidade do software produzido por meio de *crowd-*

sourcing através de uma perspectiva de classificação e esforço em projetos. Indicadores de classificação presentes no trabalho dos desenvolvedores podem ajudar um gerente de projeto à tomada de decisões. Porém, não é levado em consideração a análise de produtividade do desenvolvimento de um software utilizando repositórios.

Sendo assim, o presente trabalho propõe, a análise de fatores de contribuição de colaboradores em desenvolvimento de software *crowdsourced* através de dados encontrados em repositórios abertos. Dados esses que podem auxiliar coordenadores de projetos que utilizem de *crowdsourcing* no desenvolvimento de software.

3. O serviço para apoiar a coordenação

Para desenvolver e definir o serviço, requisitos funcionais foram definidos, tais como: (RF1) Efetuar buscas por repositórios desejados no banco de dados de uma plataforma de *codehosting* online; (RF2) Listar repositórios encontrados na pesquisa; (RF3) Apresentar dados gerais dos repositórios encontrados na pesquisa para uma boa compreensão de aspectos dos repositórios; (RF4) Explorar um repositório à interesse do usuário analisando 3 fatores: (a) Colaboradores mais relacionados às *issues*; (b) A evolução dos *commits* ao longo do tempo; (c) O fator de contribuição gerado pela plataforma de *codehosting*; (RF5) Gerar gráficos com os 10 melhores contribuidores automaticamente a partir de cada fator em análise; (RF6) Demonstrar os 10 melhores contribuidores de cada fator em lista decrescente. Como requisitos não funcionais, podemos citar: (RNF1) Tempo de processamento dos dados, e (RNF2) Escalabilidade das bases de busca.

Foi escolhida a plataforma Github como fonte de dados, uma vez que esta possui ampla usabilidade em projetos de várias esferas e uma API bem documentada para que o serviço fosse desenvolvido. Foi utilizada a API¹ que traz informações de repositórios do Github e que utiliza a linguagem Java para sua comunicação e programação.

A relação dos colaboradores com as *issues*, primeiro fator do RF4, é essencial para o entendimento de quais colaboradores têm mais se envolvido com os problemas abertos pelo *crowdsourcer* no Github. O segundo fator do RF4, a evolução dos *commits* dos usuários, é importante para analisar quais colaboradores atuaram em um determinado ponto do projeto, e quais têm atuado mais ativamente no momento. O terceiro fator do RF4, a contribuição dos colaboradores, é um dado informado pelo próprio Github que gera um número que expressa a contribuição de cada participante no projeto. Esses três fatores foram escolhidos com o objetivo de demonstrar características dos participantes que possam apresentar ao coordenador do projeto uma lista com os mais atuantes colaboradores e como eles têm produzido e reagido à suas demandas com o tempo.

Para que fosse viável analisar projetos com muitos colaboradores, foram gerados gráficos programados que mostram dados dos 10 melhores participantes de acordo com o fator em observação pois projetos com muitos contribuidores poderiam gerar enormes listas e gráficos. Sendo assim, o serviço proposto tem como função principal: a análise conjunta de fatores específicos de repositórios que remonta a qualidade de sua produção. Este tipo de análise conjunta torna o serviço diferente de outros trabalhos ou até mesmo da *dashboard* do próprio Github.

¹<http://github-api.kohsuke.org>

3.1. Implementação do serviço

Para explorar um repositório, primeiramente devemos entrar com os termos desejados na barra de busca e clicar no botão ao lado. Ao clicar em um dos repositórios da lista, o sistema carrega informações para que o usuário identifique seu repositório ou de terceiros. A Figura 1 representa a tela de pesquisa. Na ocasião, buscávamos pelo repositório da própria API utilizada para desenvolver o serviço. Como pode-se observar, ao clicar em seu nome, informações sobre o repositório aparecem ao lado para que possamos compreender aspectos gerais do mesmo. O próximo passo é explorar o repositório.

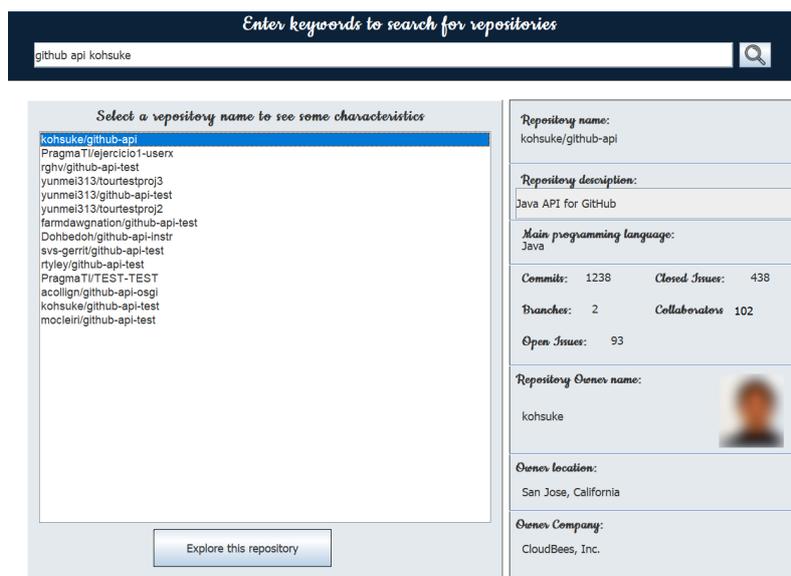


Figura 1. Tela de busca dos repositórios desejados pela entrada de termos na barra de busca

Ao selecionar o repositório desejado, o usuário poderá clicar em “*Explore this repository*” para que uma janela de exploração seja aberta, como mostra a Figura 2. Na mesma, o usuário encontrará filtros para que sua pesquisa seja efetuada. Esses filtros são importantes, uma vez que o tempo de busca e processamento de informações em repositórios grandes pode ser significativo. Informações básicas do repositório aparecerão ao lado direito da tela. Ao clicar em “*Get repository data*” todas os dados referentes aos três fatores serão processadas de acordo com os filtros selecionados. Após o preenchimento das listas de colaboradores em cada fator e seus gráficos gerados, o usuário poderá clicar nas imagens do colaborador ao qual deseja obter informações como login, nome, localidade, empresa e seu e-mail.

A Figura 3 ilustra como o serviço informa ao usuário dados do primeiro fator, os colaboradores mais relacionados às *issues*. É possível observar quais são os 10 melhores colaboradores deste projeto, que contém 102 colaboradores. Ao selecionar um determinado colaborador de interesse, podemos visualizar suas informações de contato.

As informações dos fatores 2 e 3 são expressas de maneira semelhante às do fator 1, com a diferença que, para o fator 2, a evolução dos *commits* dos colaboradores, teremos um gráfico de linhas, com linhas para cada usuário, mostrando o volume de *commits* de cada um com o passar do tempo. Por limitações de espaço não foi possível demonstrar



Figura 2. Tela de exploração dos repositórios

as telas dos fatores 2 e 3 neste artigo, entretanto, as mesmas possuem similaridade com a Figura 3.

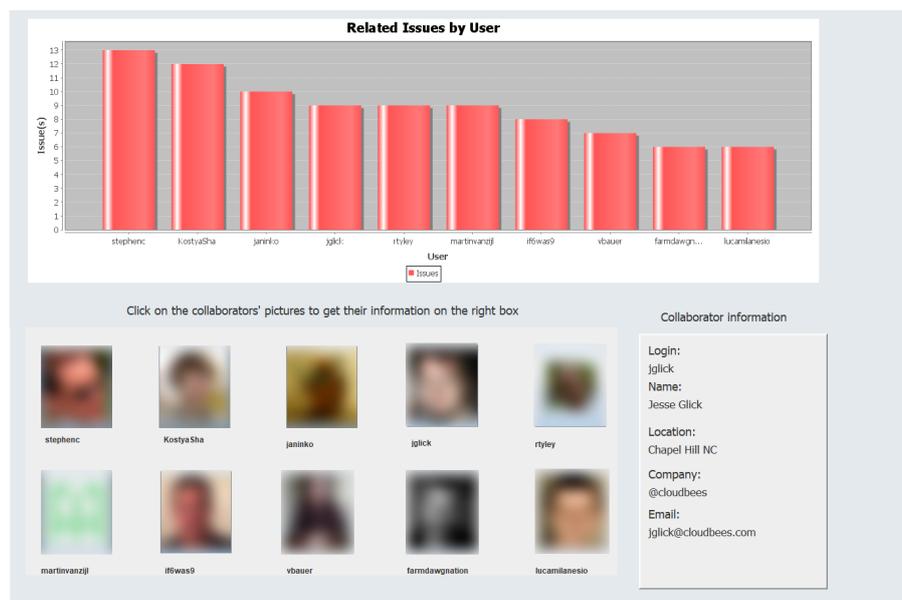


Figura 3. Resultado da primeiro fator apresentado pela aplicação ao usuário

4. Desafios

Durante o desenvolvimento do serviço descrito, vários desafios surgiram em sua implementação. A utilização de bibliotecas da linguagem JAVA ainda não conhecidas, o uso e adaptação da API, a melhor maneira para poder estruturar e apresentar os dados ou ainda a maneira como a pesquisa de repositórios poderia ser efetuada.

Outro aspecto, ainda em desenvolvimento, diz respeito à integração do serviço a uma plataforma de Ecosistema de Software. Com isso, esperamos tratar de outros

desafios a este contexto. Dentre eles, cabe mencionar a análise de reputação dos desenvolvedores associados a um projeto inserido na plataforma.

5. Conclusão

Após a compreensão de desafios e soluções identificados na área, através de um Mapeamento Sistemático, foi observada a ausência um serviço que auxiliasse o gestor de um projeto de desenvolvimento de software *crowdsourced* por meio da análise de aspectos existentes em repositórios de projetos. Sendo assim, desenvolveu-se um serviço através do qual pode-se pesquisar e explorar, um repositório do Github. Ao analisar seus fatores de produtividade, sabe-se quais especialistas podem apoiar a coordenação de um projeto no contexto de Desenvolvimento Global de Software. Tal aplicação traz inovação para a área pois, como foi constatado, ainda não haviam sido propostas soluções que trabalhassem os aspectos mencionados.

Sendo assim, como trabalhos futuros cabe validar o serviço com gestores de projetos *crowdsourced*, aprimorar a aplicação em suas funcionalidades já presentes, e também analisar mais fatores dos colaboradores de um repositório para que o usuário possa obter informações sobre o repositório que está sendo explorado. Além disso, cabe integrar a solução a uma plataforma de ecossistemas de software com o objetivo de apoiar a construção de softwares em domínios de aplicação específicos, por exemplo, *e-Science*.

Referências

- Kitchenham, B. and Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering.
- Li, B., Wu, W., and Hu, Z. (2018). Evaluation of software quality for competition-based software crowdsourcing projects. pages 102–109.
- Mao, K., Capra, L., Harman, M., and Jia, Y. (2017). A survey of the use of crowdsourcing in software engineering. *Journal of Systems and Software*, 126:57 – 84.
- Stol, K., LaToza, T. D., and Bird, C. (2017). Crowdsourcing for software engineering. *IEEE Software*, 34(2):30–36.
- Stol, K.-J. and Fitzgerald, B. (2014). Two’s company, three’s a crowd: A case study of crowdsourcing software development. In *Proceedings of the 36th International Conference on Software Engineering, ICSE 2014*, pages 187–198, New York, NY, USA. ACM.
- Won Lee, S., Chen, Y., Klugman, N., R. Gouravajhala, S., Chen, A., and S. Lasecki, W. (2017). Exploring coordination models for ad hoc programming teams. pages 2738–2745.
- Zhou, W., Yan, W., and Zhang, X. (2017). Collaboration for success in crowdsourced innovation projects: Knowledge creation, team diversity, and tacit coordination.