Visualização de Dados – Relatorio Trabalho Final

Danilo Ferreira, Guilherme Avelino, Hudson Borges

[1]Departamento de Ciência da Computação, UFMG

{danilofs,gaa,hsborges}@dcc.ufmg.br

Abstract. *Nesse trabalho desenvolvemos um conjunto de visualizações com o intuito de auxiliar na identificação de especialistas de código em sistemas de software. A indentificação do autor/especialista em artefato de código é uma atividade importante para auxilio na manutenção e evolução de software.*

1. Introdução

Nos últimos anos o GitHub se tornou uma poderosa ferramenta de colaboração para desenvolvimento de software. Atualmente é o maior repositório de código do mundo, com mais de 6.8 milhões de colaboradores e 15.2 milhões de repositórios armazenados. Muito de sua popularidade se deve a características como controle de versão distribuído e foco em *social coding* [1].

No trabalho aqui apresentado, buscamos desenvolver visualizações que auxiliem na identificação de especialistas em artefatos de software. As visualizações aqui apresentadas foram construídas com objetivo de analisar um conjunto de dados extraídas de sistemas Open Source armazenados no GitHub[1]. Nossas visualizações tem como foco destacar informações sobre os principais desenvolvedores dos sistemas, aqui denominados autores. Os autores são calculados baseados em informações de commits extraídas dos repositórios de dados e representam desenvolvedores com grande conhecimento sobre um determinado arquivo. A fórmula para cálculo do grau de autoria de um desenvolvedor sobre um arquivo é detalhada no trabalho de Fritz [2].

Nossa ferramenta de análise e visualização de dados está disponível através do link http://homepages.dcc.ufmg.br/ hsborges/2015/1/data-visualization/.

2. Interface da página

A primeira visualização a ser destacada neste trabalho é o *layout* adotado para a apresentação de todas as visualizações. O *layout* adotado por todas as páginas foi desenvolvido para se adaptar a diferentes tamanhos de telas (*responsive design*), possui uma interface limpa com conteúdos bem distribuidos e tem como base novas tecnologias *web* (e.g., as páginas não são recarregadas ao navegar no *website*). Mais especificamente, a interface da página possui três componentes principais:

Menu de Navegação: O menu de navegação encontra-se posicionado à esquerda na página e lista todas as páginas acessíveis. Cada opção do menu apresenta um título e um ícone indicativo, contudo os titulos podem ser escondidos para aumentar o espaço de apresentação

[1]https://github.com/

das visualizações. Essa funcionalidade é bastante útil em dispositivos móveis e/ou computadores que possuem telas menores, pois possibilita um melhor aproveitamento na área de apresentação das visualizações.

Menu superior: O menu superior consiste de uma pequena área localizada na parte superior da página que permite ao usuário se localizar e também apresenta a última data de atualização dos dados. Esse último detalhe é bastante importante pois as visualizações apresentadas são resultados de analises estáticas e que podem sofrer alterações durante o tempo, logo é importante deixar claro aos usuário à quando os resultados se referem.

Área de apresentação: Por fim, a área de apresentação dos dados é a maior região da página e apresenta o conteúdo da página visitada, sendo atualizada (sem necessidade de recarregamento completo) cada vez que o usuário navega pelas páginas disponíveis. Um detalhe importante está relacionada à biblioteca de desenho de gráficos utilizada, o D3.JS. Os gráficos desenhados por tal biblioteca não apresentam funcionalidades responsivas por padrão, logo, apesar da página possuir tais funcionalidades, quando um gráfico é desenhado na tela ele não é redimensionado automaticamente, para isso é necessário que o próprio usuário recarregue a página para que a biblioteca identifique e desenhe o gráfico de acordo com as novas dimensões.

3. Visualizações

Nas próximas subseções são apresentados detalhes sobre as principais vizualizações implementadas.

3.1. Dashboard

O dashboard é a primeira visualização apresentada quando os usuários visitam o *website*. Nessa interface são apresentadas informações de alto nível que tem por objetivo fazer com o que os usuários tenham uma visão geral do nosso objetivo e os dados que utilizamos como base em nossa pesquisa. Partindo da premissa que a leitura é feita da esquerda para a direita e de cima para baixo, inicialmente apresentamos os dados referentes ao número de repositórios, desenvolvedores, *commits* e arquivos que foram analisados. Em seguinda apresentamos uma tabela um pouco mais detalhada destes mesmos dados mas por linguagem. Por fim, apresentamos um gráfico de barras com a participação dos melhores autores por repositório.

3.2. Best Authors

A visualização *Best Authors* apresentam o percentual de arquivos que os principais autores de cada sistema dominam. Para não poluir excessivamente a tela, são mostrados no máximo 20 autores por sistema. Essa visualização tem como principal objetivo destacar a importância desses autores em cada sistema, sendo possível observar que em muitos dos sistemas analisados esses são responsáveis por 100% dos arquivos do sistema, demonstrando a concentração da responsabilidade em poucos desenvolvedores.

Os sistemas foram ordenados tendo como base o percentual de domínio do principal author de cada sistema. Dessa forma, destacamos do lado direito os sistemas onde o principal author tem domínio sobre a maior parte dos arquivos.

3.3. Dev/Authors

Aqui são apresentados boxplots para detalhar a distribuição da razão desenvolvedor author nos sistemas analizados. Essa análise permite verificar qual o percentual de desenvolvedores que contribuiu de forma significativa para o desenvolvimento, ganhando autoria sobre pelo menos um arquivo.

Para auxiliar na análise foram gerados boxplots para os valores agrupados por linguagem (primeiro) e por tamanho do sistema (segundo).

3.4. Workload

Essa visualização tem como objetivo analisar a distribuição da carga de trabalho entre os autores do sistema. Os dados são apresentados dividindo a carga de trabalho (número de arquivos) entre os autores agrupados de 10 em 10%. Destamos em amarelo os 10% mais ativos e em azul os seguintes 10% mais ativos.

A analize da visualização permite observar que para quase a totalidade dos sistemas estudados a carga de trabalho é má distribuída sendo a maior parte dessa de responsabilidade de apenas 10% dos autores.

3.5. Workload - Multi authors

Essa visualização é uma extensão da visualização anterior, porém nela é considerado que um arquivo pode ter mais que um author. Com isso a soma do percentual de arquivos dominado por cada classe de desenvolvedor acaba superando os 100%. Essa análise é importante, pois é relativamente comum a existência de arquivos onde mais de um desenvolvedor tem conhecimento similarmente grande sobre um arquivo, podendo ambos serem considerados autor desse.

3.6. Distribution

O *Distribution* apresenta como a autoria está distribuída pelo sistema. Seu objetivo é demonstrar de forma hierárquica como está espalhada a autoria. Ela é baseada em *Tree Map*, porém com uma abordagem diferente, na qual os agrupamentos são apresentados de forma circular. Foi desenvolvido para ser interativo, possibilitando navegar na hierarquia de pastas do sistema. Essa navegação é especialmente útil para sistemas grandes, permitindo, inicialmente, visualizar a distribuição da autoria a nível de módulos (diretórios), mas possibilitando navegar até o nível de arquivos.

Na visualização cada cor representa um author diferente, facilitando observar o domínio de cada autor sobre o sistema.

Referências

[1] L. Dabbish, C. Stuart, J. Tsay, and J. Herbsleb. Social Coding in GitHub: Transparency and Collaboration in an Open Software Repository. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, pages 1277–1286, 2012.

[2] T. Fritz, G. C. Murphy, E. Murphy-Hill, J. Ou, and E. Hill. Degree-of-knowledge: Modeling a developer's knowledge of code. *ACM Transactions on Software Engineering and Methodology*, 23(2):1–42, Mar. 2014.