A Transparência do GitHub para uso de Artefatos como fontes de informação na Engenharia de Requisitos

Roxana Lisette Quintanilla Portugal, Julio Cesar Sampaio do Prado Leite.

Departamento de Informática – Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio)
R. Marquês de São Vicente, 225 - Gávea, Rio de Janeiro - RJ, 22451-900 – Brazil

rportugal@inf.puc-rio.br, julio@inf.puc-rio.br

Abstract. *This paper describes the research in progress: we are exploring the GitHub, an ecosystem of software developers, in order to achieve requirements reuse. This repository, which owns transparency attributes, is being explored mostly for its collaborative aspects and from different approaches. Therefore, the aim of this work is finding what is valuable for a requirements engineer when eliciting information from GitHub. In other words, what is the role of the document reading technique when the source of information it is vast, as GitHub. For achievement of this objective, this work it is exploring different data mining strategies.*

Resumo. *Este artigo descreve os avanços da pesquisa que vem sendo realizada sobre os artefatos produzidos no ecossistema de desenvolvedores de software GitHub, com o intuito de reusar requisitos. Este repositório de projetos possui características de transparência, e vem sendo estudado principalmente por seus aspectos de colaboração por diversas abordagens. O objetivo de nosso trabalho é explorar aquilo que é insumo para um engenheiro de requisitos quando trabalhando na elicitação, ou seja, utilizar a técnica de leitura de documentos utilizando o GitHub como fonte de informação. Para alcançar esse objetivo estamos explorando diversas estratégias de mineração de dados.*

1. Introdução

Fontes de informação na engenharia requisitos são parte do processo de elicitação, elas são importante para delinear o contexto e obter os requisitos, que depois serão transformados em software. A pesquisa de fontes de informação (Leite, 2007) é vital dentro dos métodos ágeis de desenvolvimento, tanto pelo fator tempo, como pela distribuição geográfica dos desenvolvedores. Este trabalho propõe explorar o repositório GitHub[1] como fonte de informação, pois o fato de ter as informações acessíveis e com publicidade permite o reuso dessas informações a partir da perspectiva de um engenheiro de requisitos. Trabalhos anteriores (Dabbish, 2012; Dabbish, 2013) mostraram o uso desta transparência, de modo que é possível avaliar, por exemplo, se um desenvolvedor é um perito de acordo com as suas atividades de desenvolvimento de software, registradas no GitHub e que são de acesso publico. Outra qualidade que ajuda à transparência no GitHub é a sua natureza web, que permite a visualização das ações de

[1] GitHub: É uma plataforma de desenvolvimento colaborativo para projetos que utilizam o sistema de controle de versão Git. O código é armazenado em público.

desenvolvedores no decorrer da construção software e também através do compartilhamento de artefatos. Uma das perguntas que (Dabbish, 2012) propõe é: que inferências as pessoas fazem quando a transparecia é integrada em um ambiente de trabalho web? Uma delas poderia ser, o fato de aprender a partir de projetos existentes, o reusa-lo caso seja interessante. Nesse sentido, concordamos com (Dabbish, 2012) que argumenta que a transparência do Github apoia a aprendizagem a partir das ações de outros desenvolvedores.

GitHub também permite a coordenação e a comunicação, pois fornece um ambiente onde é possível criar conhecimento distribuído. Além disso, as atividades públicas, que seguem o formato de uma rede social, podem levar a uma melhor tomada de decisões, redução de conflitos e aumento da produtividade (Carlile, 2002; Bechky, 2006; Dabbish, 2013). Consequentemente, esse ambiente colaborativo permite uma abertura para a adoção das boas práticas, novas ideias e conhecimento compartilhado (Tortoriello & Krackhardt, 2010; Dabbish 2012). No entanto, embora estes trabalhos mostrem características de transparência no GitHub que podem ser aproveitadas, pouco tem sido explorado sobre a natureza das discussões ou avaliações feitas sobre essas informações publicas (Tsay, 2014). Um desses poucos trabalhos expõe a ferramenta GILA (Izquiero 2015; Cabot 2015), que usa informações do GitHub para inferir informações a partir das tags atribuídas às issues de cada projeto. Assim, através das tags que categorizam cada issue, sejam tags que pertencem aos meta-dados do GitHub ou àquelas definidas pelo próprio usuário, é possível chegar a interpretar o problema que esta se tentando resolver. Outra inferência é feita a partir da participação dos desenvolvedores e colaboradores nas issues, a partir destas informações é possível determinar, quais são os usuários especialistas num determinado tópico.

Finalmente, o trabalho mais próximo da nossa abordagem, que é localizar artefatos que servem como uma fonte de informação para o engenheiro de requisitos, está na investigação de (Salo, 2014). Nesse trabalho, se indica que uma issue no Github pode ser qualquer coisa, desde notas simples, assuntos ou até mesmo requisitos.

2. Objetivos da Pesquisa

Utilizar as características de transparência do repositório GitHub para identificar os possíveis artefatos que possam ser utilizados para realizar mineração de textos, de acordo a uma cadeia de busca.

Analisar a estrutura do GitHub para identificar artefatos, em linguagem natural, factíveis de extração e que possam ser uteis para o engenheiro de requisitos. Desta forma, a tarefa de leitura pode ser facilitada usando técnicas de mineração de texto.

Apresentar as informações categorizadas por termos que fazem parte do léxico do engenheiro de requisitos, como: os interessados, metas, funcionalidades, qualidades, tarefas; por exemplo.

Usar os meta-dados do github para fazer inferências a partir das informações obtidas na mineração. Tais como filtros de issues comuns que são categorizados como questões de tipo bug, e que poderiam ajudar a antecipar situações ao elicitar ou criar requisitos.

3. Contribuições esperadas

Os textos resultantes da mineração terão a seguinte formatação (Fig.1) com o intuito de que o engenheiro de requisitos possa mergulhar só na informação que é de interesse no momento de elicitar, sem perder o traço às fontes. Isso permitiria menos esforço do que se teria ao fazer: uma busca convencional no GitHub, selecionar os repositórios de interesse, para depois fazer uma revisão manual dos conteúdos.

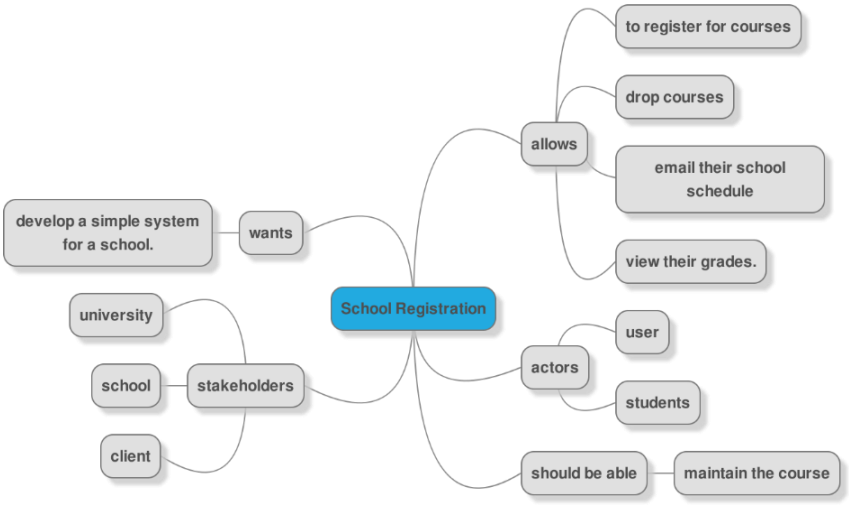

Figura 1. Apresentação das informações para o E.R

Automatizar a formatação proposta para otimizar o bestmatch do GitHub, segundo a perspectiva de um engenheiro de requisitos.

Servir como uma fonte de informação, não só para os engenheiros de requisitos, mas também para ajudar a qualquer usuário na definição de um produto, principalmente, se a característica de *Time to market*[2] for critica.

4. Resultados já alcançados

Um primeiro mapeamento das características de transparência identificadas no GitHub com as características do catálogo de transparência [Capelli, 2009] apresenta o que registra a Figura 2, com as características comuns realçadas.

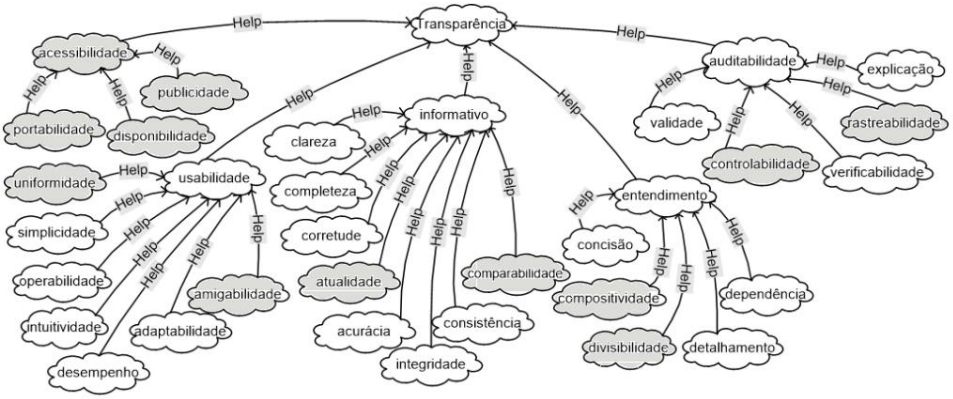

Figura 2. Carácteristicas de transparência do GitHub

[2] Time to market: É o período de tempo que leva concebir um produto até que seja disponível para venda.

Além disso, já foi feita uma revisão de literatura (Meyer, 2008) para delinear uma estratégia em mineração de textos, e sobre técnicas usadas em várias abordagens de mineração de textos em linguagem natural focadas na descoberta requisitos. (Sayão, 2007; Meth, 2013), também criou-se um algoritmo para a mineração de textos. Estamos explorando o banco de dados do GitHub através de sua API[3]. Tal algoritmo permite a extração de artefatos em linguagem natural, tais como os readme de cada projeto, a partir de uma cadeia de busca.

Em particular estamos utilizando a ferramenta de estatística computacional R[4], experimentando os algoritmos que implementam as técnicas de mineração de texto (Meyer, 2008; Murhaf, 2013) e vem sendo utilizados para as tarefas de mineração dado um corpus de documentos. No caso de este estudo, os readmes entraidos do GitHub.

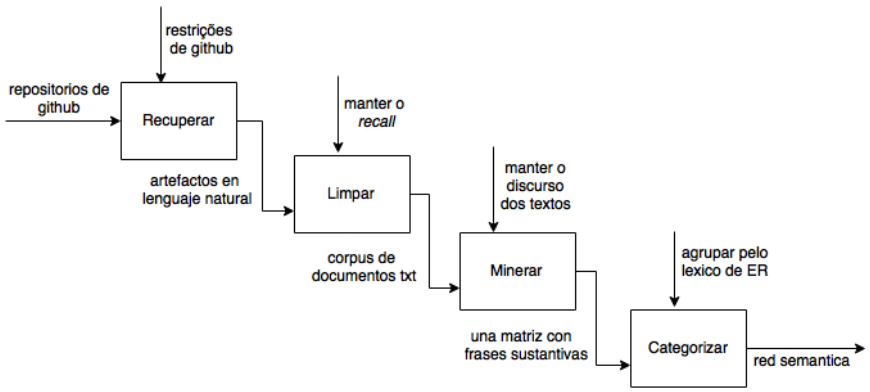

Figura 3. Atividades para a mineração de textos

A Figura 3 fornece uma visão geral do processo que estamos implementando para dar apoio a identificação de fontes de informação visando o reuso.

5. Conclusão

Vamos continuar neste caminho para realizar a mineração de textos, tomando cuidado para não perder informações e salvando os traços que marcam onde a informação foi encontrada. Nesse sentido vem sendo estudado trabalhos anteriores (Goldin, 1997; Berry, 2005; Berry, 2007) que nos mostram os perigos no tratamento de textos em linguagem natural, e ainda mais que em nosso caso são textos não estruturados. Finalmente, no decorrer desta pesquisa se esta tendo cuidado de manter o *recall* (Berry, 2012) da fonte a ser minerado, o que é uma questão muito importante ao tentar fazer uma ferramenta de apoio na engenharia de requisitos.

6. References

Do Prado Leite, J. C. S., de Moraes, E. A., & de Castro, C. E. P. S. (2007). A Strategy for Information Source Identification. In WER (pp. 25-34).

Laura Dabbish, Colleen Stuart, Jason Tsay, and Jim Herbsleb. 2012. Social coding in GitHub: transparency and collaboration in an open software repository. In

[3] GitHub-API: A API do GitHub que dá acesso a ler e escrever objetos no seu banco de dados.

[4] R-project: R é uma linguagem e um ambiente de desenvolvimento integrado, para cálculos estatísticos.

Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work (CSCW '12). ACM, New York, NY, USA, 1277-1286. DOI=10.1145/2145204.2145396 http://doi.acm.org/10.1145/2145204.2145396

Dabbish, L.; Stuart, C.; Tsay, J.; Herbsleb, J., "Leveraging Transparency," Software, IEEE , vol.30, no.1, pp.37,43, Jan.-Feb. 2013 doi: 10.1109/MS.2012.172

Jason Tsay, Laura Dabbish, and James Herbsleb. 2014. Let's talk about it: evaluating contributions through discussion in GitHub. In Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE 2014). ACM, New York, NY, USA, 144-154. DOI=10.1145/2635868.2635882 http://doi.acm.org/10.1145/2635868.2635882

Canovas Izquierdo, J.L.; Cosentino, V.; Rolandi, B.; Bergel, A.; Cabot, J., "GiLA: GitHub label analyzer," Software Analysis, Evolution and Reengineering (SANER), 2015 IEEE 22nd International Conference on , vol., no., pp.479,483, 2-6 March 2015 doi: 10.1109/SANER.2015.7081860

Cabot, J.; Canovas Izquierdo, J.L.; Cosentino, V.; Rolandi, B., "Exploring the use of labels to categorize issues in Open-Source Software projects," Software Analysis, Evolution and Reengineering (SANER), 2015 IEEE 22nd International Conference on , vol., no., pp.550,554, 2-6 March 2015 doi: 10.1109/SANER.2015.7081875

Salo, R. (2014). A guideline for requirements management in GitHub with lean approach.

Aló, Claudia Cappelli. “Uma abordagem para transparência em processos organizacionais utilizando aspectos”. Diss. PUC-Rio, 2009.

Meyer, D., Hornik, K., & Feinerer, I. (2008). Text mining infrastructure in R. Journal of Statistical Software, 25(5), 1-54.

Sayão, M. (2007). Verificação e validação em requisitos: Processamento da linguagem natural e agentes (Doctoral dissertation, PUC-Rio).

Meth, H. (2013). A Design Theory for Requirements Mining Systems.

Murhaf, F. (2013). ERG Tokenization and Lexical Categorization: A sequence labeling approach.

Goldin, L., & Berry, D. M. (1997). AbstFinder, a prototype natural language text abstraction finder for use in requirements elicitation. Automated Software Engineering, 4(4), 375-412.

Berry, D., Gacitua, R., Sawyer, P., & Tjong, S. F. (2012). The case for dumb requirements engineering tools. In Requirements Engineering: Foundation for Software Quality (pp. 211-217). Springer Berlin Heidelberg

Berry, D. M., & Kamsties, E. (2005). The syntactically dangerous all and plural in specifications. Software, IEEE, 22(1), 55-57.

Berry, D. M., Kamsties, E., & Krieger, M. M. (2003). From contract drafting to software specification: Linguistic sources of ambiguity. Technical Report, School of Computer Science, University of Waterloo, Waterloo, ON, Canada.