

# Educação em Engenharia de Software no contexto do Projeto do Novo Portal do Software Público

Camila Ferreira<sup>1</sup>, Paulo Meirelles<sup>1</sup>, Aline Gonçalves<sup>1</sup>, Hilmer Neri<sup>1</sup>

<sup>1</sup>Faculdade UnB Gama

Universidade de Brasília, Brasil

camilaferreira251@gmail.com, paulo@softwarelivre.org, alinegsantoss@gmail.com, hilmer@unb.br

**Resumo**—Source code metrics are not new, but they have not yet been fully explored in software development. For example, most metrics do not have known thresholds to guide their use by non-experts. Most metrics tools show isolated numeric values, which are not easy to understand because the interpretation of these values depends on the implementation context. Kalibro Metrics goal is to improve the readability of source code metrics. It allows a source code metric user to create a configuration of thresholds associated with qualitative evaluation, including comments and recommendations. Using these configurations, Kalibro shows metric results in a friendly way, helping software engineers to spot design flaws to refactor, project managers to control source code quality, and software adopters and researchers to compare specific source code characteristics across free software projects. These configurations are shared among its users via the Kalibro Web Service and a source code tracking network to enhance metric results interpretation.

**Index Terms**—Software metrics, source code analysis, thresholds configuration, Web Service integration, source code tracking network, Free Software.

## I. INTRODUCTION

## II. PROJETO

O Portal do Software Público Brasileiro (SPB), inaugurado em 2007, na prática, é um sistema web que se consolidou como um ambiente de compartilhamento de projetos de software. Oferece um espaço (comunidade) para cada software. A comunidade é composta por fórum, notícias, chat, armazenamento de arquivos, downloads, wiki, lista de prestadores de serviços, usuários, coordenadores, entre outros recursos. Teve um crescimento expressivo contando, hoje, com mais de 60 comunidades de desenvolvimento e mais de 200.000 usuários cadastrados. O SPB abrange também o 4CMBR, que é um grupo de interesse voltado para soluções de tecnologia para municípios, o 5CQualiBr, que é um grupo que trabalha para evoluir a qualidade do SPB, o 4CTecBr, que é um portal destinado a colaboração no desenvolvimento de Tecnologias Livres, o Mercado Público Virtual, que é um grupo de empresas e pessoas que prestam serviços nos softwares ofertados no Portal, e o Ava- liaSPB, que avalia a entrada dos softwares candidatos à software público. O ambiente do SPB não proporciona a integração com ambientes colaborativos externos, especialmente com redes sociais. A plataforma escolhida na ocasião da criação foi o framework OpenACS, que continua sendo utilizada na versão atual. A evolução do SPB foi

comprometida desde 2009, quando framework OpenACS foi descontinuado. Com isso, não tendo versões lançadas a partir daquele ano. Por isso, hoje, é necessária a evolução para novas tecnologias que tenham maior suporte das comunidades de desenvolvimento, que utilize linguagens de programação com maior rapidez de aprendizagem e de desenvolvimento e que permita a integração com ambientes colaborativos externos, em especial, as redes sociais. Além disso, é preciso realizar a manutenção evolutiva das funcionalidades existentes e também o desenvolvimento de novas funcionalidades para o Portal do SPB. Um dos passos para a concretização de uma nova geração do Portal SPB é a integração de novas tecnologias, desde uma plataforma colaborativa até sistemas de controle de versão e de monitoramento da qualidade do código fonte, gerenciadas e apresentadas em uma plataforma integrada no back-end <sup>1</sup> e, em especial, no front-end <sup>2</sup> para que os usuários e as comunidades tenham um conjunto de recursos para encontrarem os projetos, bem como colaborarem em torno de um software público. Mesmo com as limitações citadas, o Portal do Software Público Brasileiro teve em 2013 mais de 600 mil visitantes únicos, com mais de 1 milhão de visitas/acessos, gerando mais de 16 milhões de visitas nas páginas, com um total de mais de 49 milhões de hits <sup>2</sup> no Portal SPB. Avaliando apenas as comunidades dos projetos I3Geo, CAU, CACIC e Geplanes, houve mais de 15 mil downloads e 4 mil mensagens nos fóruns. Essa amostra ilustra bem o potencial do Software Público Brasileiro, bem como as expectativas de seus usuários e colaboradores para a evolução do Portal e do modelo em si.

## III. ARQUITETURA

O sistema que irá atender a evolução e reformulação do Portal do Software Público Brasileiro é composto por diversos módulos, os quais irão comunicar-se entre si de forma organizada e integrada para suprir as necessidades do projeto.

As ferramentas que serão utilizadas para suprir essas necessidades serão:

<sup>1</sup>O front-end é responsável por coletar a entrada em várias formas do usuário e processá-la para adequá-la a uma especificação em que o back-end possa utilizar. O front-end é uma espécie de interface entre o usuário e o back-end.

<sup>2</sup>O número de vezes que os visitantes visualizaram uma determinada página no site.

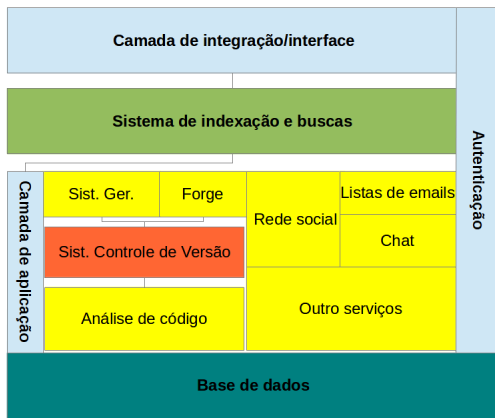


Figura 1. Proposta de arquitetura do Novo Portal do Software Público

- Para lista de e-mail estamos utilizando o Mailman na versão 2, que é um software gratuito para gerenciamento de discussão eletrônica de e-mail e listas e- newsletter;
- Para Chat estamos utilizando Punjab BOSH (XMPP), que é uma interface HTTP cliente jabber. É um gerenciador de conexão BOSH que permite conexões de clientes persistentes para um servidor XMPP (protocolo de comunicação para mensagens orientadas a middleware baseado em XML);
- Para Plataforma de Buscas estamos utilizando Apache Solr, que é uma plataforma de busca open source da Apache Lucene escrita em Java;
- Para rede social estamos utilizando o Noosfero que é uma plataforma web livre para criação de redes sociais com blog, e-Portifólios, CMS, RSS, discussão temática, agenda de eventos, galeria de imagens, chat, entre outros. Ele foi desenvolvido pela Cooperativa de Tecnologias Livres – Colivre 3 em 2007, sob a licença AGPL v.3, com a proposta de permitir ao usuário criar sua própria rede social personalizada, livre e autônoma;
- Para Forge para SVN estamos utilizando o Trac, que é uma ferramenta open source para controle de mudanças em projetos de desenvolvimento;
- Para sistemas de controle de versão estamos utilizando SVN e Git, que são ferramentas open-source para controle de mudanças em projetos de desenvolvimento;
- Para Forge para Git estamos utilizando o GitLab, que é um software livre de colaboração de código online que utiliza a ferramenta de gerência de código fonte Git;
- Para sistema de gerenciamento estamos utilizando o Redmine, que é uma aplicação web de gerenciamento de projetos que disponibiliza diversas ferramentas para auxiliar a gestão e manutenção de um projeto;
- Para suporte a Single Sign On estamos utilizando o Mozilla Persona, que foi desenvolvido pela Mozilla e permite o suporte a single sign on;
- Para Sistema de Integração contínua estamos utilizando o Jenkins, que é uma aplicação web de integração contínua

de construção de projetos.

Para integrar todas estas ferramentas estamos utilizando o Colab, que é uma plataforma de integração de ferramentas. Nele, são também integradas as interfaces das ferramentas para que, ao navegar, o usuário tenha a sensação de estar navegando em uma única ferramenta.

Nesse projeto, chegamos ao entendimento que utilizar elementos já oferecidos por softwares livres existentes seria o ideal, pois eles já atendem às nossas necessidades, evitando o retrabalho e customizando os elementos necessários. No início do projeto realizamos estudos de possíveis ferramentas que pudessem ser utilizadas para atender as nossas necessidades, chegando então na lista apresentada anteriormente.

#### IV. METODOLOGIA DE DESENVOLVIMENTO

A Engenharia de Software tem evoluído suas práticas e metodologias em busca de padrões que regem o desenvolvimento de software de qualidade dentro dos escopos, custos e prazos desejados. O modelo dito tradicional tem como característica um conjunto grande e detalhado de documentação que deve, supostamente, ser utilizada ao longo de todo o ciclo de desenvolvimento. Entretanto, percebendo que os objetivos principais do desenvolvimento de software não estavam sendo alcançados, alguns líderes da indústria e academia começaram a adotar métodos mais simples de trabalho que apresentaram melhores resultados em projetos de software. Em 2001, líderes que estavam desenvolvendo projetos fora dos padrões industriais se reuniram para trocar experiências e trabalhos. Este grupo se tornou a Aliança de Desenvolvimento Ágil e escreveram o Manifesto Ágil que apresenta os princípios e valores que o grupo considera ser determinante para o desenvolvimento de software. Neste sentido, os métodos de desenvolvimento ágeis de software são métodos que implementam os seguintes valores:

- Indivíduos e interações acima de processos e ferramentas;
- Software operante acima de documentações grandes e completas;
- Colaboração do cliente acima de negociações contratuais;
- Responder à mudanças acima de seguir a um planejamento.

Esse conjunto de valores não descartam a importância dos elementos citados à direita das setas, mas evidenciam que estes são menos importantes diante dos primeiros elementos citados. Em outras palavras, apesar da documentação ser importante, o foco principal deve estar na entrega de software de valor para o cliente e na interação e a consequente comunicação entre as pessoas. Além disso, os métodos ágeis exaltam a simplicidade, feedback contínuo e adaptação à mudanças que podem ser obtidos a partir de comunicação face à face, qualidade de código e entrega contínua de software.

Dada a oportunidade de adoção de métodos ágeis no desenvolvimento do presente trabalho, a metodologia utilizada é baseada em uma combinação das metodologias Scrum e Extreme Programming. Destacam que XP e Scrum complementam um ao outro bem, com o XP provendo suporte para aspectos

mais técnicos enquanto o Scrum provê práticas e técnicas para gerenciamento, planejamento e acompanhamento. Assim, com base nas experiências destacadas em [1] e [2], na motivação de adoção de métodos ágeis no desenvolvimento de software moderno, serão apresentadas os métodos XP e Scrum, suas principais características e práticas que serão utilizadas no desenvolvimento do presente projeto.

## V. ESCOLHA DA EQUIPE

A equipe é formada, majoritariamente, por alunos de graduação do curso de Engenharia de Software da Universidade de Brasília, conta ainda com dois ex-alunos formados, alunos de mestrado, que trabalham exclusivamente com o design, e professores orientadores.

Devido a esta formação, a equipe não consegue estar sempre trabalhando fisicamente junta, pois cada membro da equipe possui horários particulares de aula e, com isso, o horário dedicado a contribuição do projeto pelos mesmos depende do horário dessas aulas. Dessa forma, não conseguimos utilizar integralmente as práticas das metodologias ágeis.

A equipe do projeto está dividida em duas equipes: equipe Noosfero e equipe Colab, a qual também atua na integração com outras ferramentas. A equipe do Noosfero está desenvolvendo um plugin com novas funcionalidades para o Portal do Software Público. A equipe do Colab está, no momento, trabalhando com configurações das ferramentas, pois está sendo realizada a integração do Redmine e do Gitlab com o Colab, o que exige maiores esforços de infra-estrutura.

Para manter o controle das atividades do projeto e evitar os ruídos de comunicação temos uma lista de e-mail com todos os integrantes do projeto, onde foi criado o hábito de enviarmos um e-mail ao final do dia com as atividades que desenvolvemos, e a ferramenta de gerenciamento Redmine, onde são escritas as histórias de usuários, as histórias técnicas e suas respectivas tarefas. Realizamos ainda um stand-up em todos os dias de trabalho, em que todos estão ou no laboratório ou presente virtualmente para alinharmos a situação das equipes.

Estamos trabalhando em sprints/ciclos de duas semanas, em média, e releases de quatro meses, sendo que o projeto tem duração programada de três anos, sendo ao total sete releases e cinquenta e oito sprints.

## VI. RELATO DE EXPERIÊNCIA

Foi aplicado um questionário na equipe de desenvolvimento do Novo Portal do Software Público, onde os mesmos poderiam expressar o quanto estão aprendendo com o projeto e o quanto o projeto está ajudando na sua formação.

O questionário foi respondido por 17 integrantes do projeto que têm entre 20 e 26 anos, os quais estão entre o 5º e o 10º semestre do curso de Engenharia de Software.

A primeira questão pretendia averiguar o nível de conhecimento dos entrevistados, em uma escala de 1 a 5, em relação às ferramentas que estão sendo utilizadas no projeto. Como resultados tivemos que 63% responderam nível 3, 19% responderam nível 2 e 19% responderam nível 4.

A segunda questão pretendia averiguar o quanto o projeto está contribuindo para a formação do entrevistado. Em uma escala de 0 a 5, 69% dos entrevistados responderam "5 - Contribui muito, mais que determinadas disciplinas", 19% dos entrevistados responderam "4 - Contribui, no mesmo nível de muitas disciplinas" e 13% dos entrevistados responderam "3 - Contribui, da mesma forma que um estágio convencional".

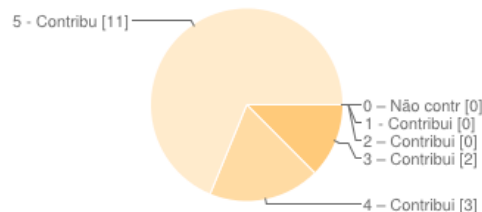


Figura 2. Respostas do nível de contribuição do projeto na sua formação

A terceira questão está relacionada, também em uma escala de 0 a 5, a quanto o projeto atrapalha a sua performance nas disciplinas da graduação. Como resultados, 19% dos entrevistados responderam "3 - Atrapalha minha graduação da mesma forma que um estágio", 50% dos entrevistados responderam "2 - Atrapalha moderadamente", 19% dos entrevistados responderam "1 - Atrapalha pouco, menos que um estágio" e 13% dos entrevistados responderam "0 Não atrapalha em nada na minha graduação".

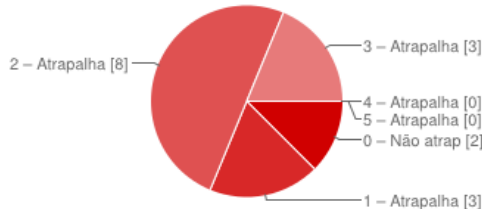


Figura 3. Respostas do nível de performance nas disciplinas de graduação

A quarta questão averigua o quanto, em uma escala de 0 a 5, os conhecimentos adquiridos durante a execução do projeto ajudam os entrevistados nas disciplinas da graduação. Como resultados, 7% dos entrevistados responderam "5 - Ajuda muito, sempre utilizo os conhecimentos adquiridos nas disciplinas da graduação", 47% respondeu "4 Ajuda muito, utilizo os conhecimentos com frequência nas disciplinas da graduação", 27% respondeu "3 - Ajuda, utilizo os conhecimentos nas disciplinas da graduação", 13% respondeu "2 - Ajuda moderadamente, as vezes utilizo os conhecimentos adquiridos no projeto" e 7% respondeu "1 - Ajuda um pouco, utilizo pouco os conhecimentos do projeto".

E a quinta questão perguntou o quanto, em uma escala de 0 a 5, o entrevistado acredita que o projeto do Novo SPB vai contribuir em experiência para o mercado de trabalho. Como resultados, 38% responderam "5 O projeto me tornará experiente para o mercado de trabalho", 38% responderam "4 - O projeto me dará muita experiência para o mercado de

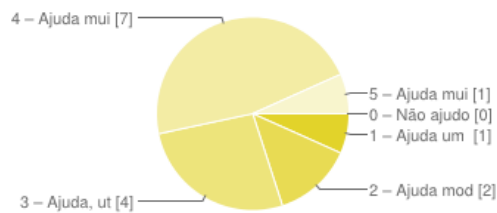


Figura 4. Respostas do nível de conhecimento adquirido

trabalho" e 25% responderam "3 - O projeto me dará uma boa experiência para o mercado de trabalho".

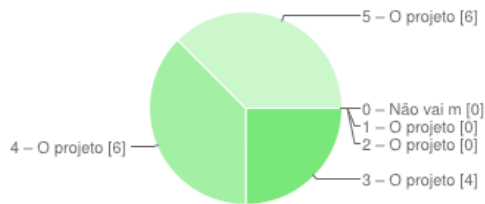


Figura 5. Respostas do nível de experiência para o mercado de trabalho

## VII. CONCLUSÃO

A partir do relato de experiência podemos concluir que os integrantes do projeto não tinham muita experiência nas ferramentas a serem utilizadas no projeto e ainda assim os entrevistados conseguiram contribuir com o projeto.

Foi possível concluir também que o projeto está contribuindo com a construção da formação dos alunos e ainda ajuda no desempenho dos alunos nas disciplinas da graduação devido ao conhecimento prévio adquirido. Mesmo que o tempo gasto trabalhando no projeto atrapalhe moderadamente o desempenho na graduação dos integrantes do projeto, o esforço será recompensado com uma boa experiência para o mercado de trabalho.

De maneira geral o projeto tem ajudado os seus integrantes a aprender a lidar com dificuldades encontradas no desenvolvimento de software, está trazendo conhecimento técnico e gerencial para os membros da equipe além de auxiliar na relações interpessoais.

## REFERÊNCIAS

- [1] K. Schwaber and M. Beedle, *Agile Software Development with Scrum*, 1st ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.
- [2] B. Fitzgerald, G. Hartnett, and K. Conboy, "Customising agile methods to software practices at intel shannon," *Eur. J. Inf. Syst.*, pp. 200–213, 2006.