

A FERRAMENTA SONARQUBE NA GESTÃO DA QUALIDADE DE SOFTWARE

Lael Santos Francisco¹

Fernando de Sousa Faria²

RESUMO

A gestão da qualidade deve ser uma preocupação recorrente das empresas. E são decorrências da qualidade, a produtividade e a competitividade. No que tange ao desenvolvimento de software, é imperioso que tais virtudes sejam visadas. Há normas e certificações nacionais e internacionais que padronizam processos com vistas à qualidade. Questiona-se neste artigo se a ferramenta SonarQube auxiliaria na gestão da qualidade nos processos relacionados à construção de sistemas. Este artigo pretende responder a questão por meio do método de pesquisa qualitativo e quantitativo e tendo como instrumento de pesquisa a análise documental. O trabalho está organizado basicamente em duas partes: revisão de literatura e relato de experiência. Nesta última buscou-se responder argumentativamente ao problema. Trata-se de um assunto relevante e prático, sendo de interesse profissional e acadêmico, portanto.

Palavras-chave: SonarQube. Gestão. Qualidade. Produtividade. Competitividade.

ABSTRACT

Quality management must be a reoccurring concern of businesses. There are consequences on quality, on productivity, and on competitiveness. Regarding the development of software, it is imperative that such virtues are concerned. There are national and international standards and certifications that standardize processes aimed at quality. It is questioned, in this article, if the tool SonarQube would assist in the quality management of processes related to development systems. This article aims to answer that question by means of showing qualitative and quantitative data, as well as having a research tool to document analysis. The paper is organized basically into two parts: literature review and report of results. The latter half attempts to answer the problem. It is a relevant and practical subject, and has both professional and academic interest.

Keywords: SonarQube. Management. Quality. Productivity. Competitiveness

Introdução

O presente artigo trata da gestão da qualidade no processo de desenvolvimento de software. Partindo da experiência profissional do autor, percebeu-se que poderia haver uma ferramenta que atendesse as necessidades no gerenciamento da qualidade em códigos implementados.

Pensou-se, inicialmente, que a ferramenta SonarQube pudesse ser uma resposta possível a esta busca. Assim sendo, deparou-se diante da seguinte questão motriz deste artigo: Pode-se realizar o gerenciamento da qualidade do

¹ Analista de Sistemas, pós-graduando em MBA de Gestão de Projetos na Faculdade G&P. E-mail: lael.san@gmail.com.

² Orientador, professor da Graduação e Pós-graduação da Faculdade G&P. E-mail: fersfaria@gmail.com

código desenvolvido por meio de verificação de análises e monitoramento mediante a ferramenta SonarQube?

É a meta deste trabalho, com vistas a responder a problematização suscitada, apresentar instrumentos e estratégias para realizar o gerenciamento da qualidade no desenvolvimento de *software*. Para tanto, é apresentada revisão de literatura sobre o que se entenderia por qualidade e seus conceitos conexos além de apresentar normas e certificações de boas práticas nos processos de implementação de sistemas. Em seguida, um relato de experiência da pesquisa empreendida demonstra - argumentativamente - os resultados obtidos.

Para se chegar aos resultados, serviu-se do seguinte método: qualitativo e quantitativo e tendo como instrumento de pesquisa a análise documental³.

Este artigo é organizado em duas partes. A primeira é uma apresentação do referencial teórico pertinente e a segunda parte é o relato de experiência no qual o método, os documentos, a análise e os argumentos são explicitados buscando suprir os objetivos.

O público alvo que se deseja alcançar são empresas que possuam o desenvolvimento de *software* como atividade-fim. Este trabalho pode ainda se mostrar relevante para a discussão acadêmico-teórica sobre o assunto abordado.

1. Revisão de literatura

Nesta revisão de literatura aborda-se os conceitos de qualidade, produtividade e competitividade, sob perspectiva de atualizada literatura técnica. Ato contínuo, descreve-se os aspectos normativos das ISO 25010, ISO 25012, ISO 15504, CMMI, MPS.BR, além dos conceitos primários da ferramenta SonarQube.

Conceituar a expressão "qualidade" é preocupar-se não tanto com uma abstração, mas é tratar de acepções que possuem sérias implicações em gestão, seja de empresas, de projetos, ou mesmo de órgãos públicos. Trata-se de buscar uma teoria que seja fundamentada e direcionada para prática. Esta foi exatamente a proposta de Paladini (2004) o qual afirmou ser o pensamento sobre a qualidade fator determinante para a sua gestão; à guisa de exemplo, o mesmo teórico apresentou a qualidade como sendo "a capacidade que um produto ou serviço tenha de sair conforme o seu projeto" (*ibid*, p. 22) possuindo por decorrência o seguinte: "O

³ A metodologia é tratada pormenorizadamente no Relato de Experiência.

que se considera é a relação entre o projeto e o produto, sem sequer verificar se existe relação real entre o projeto e os possíveis usuários daquilo que se projetou" (*ibid*, p. 22). É importante, portanto, à luz de Paladini, saber o que se entende e o que se busca quando o termo "qualidade" é tratado por alvo em um processo gerencial que vise ao desenvolvimento de um produto.

Ainda que conceituar qualidade seja reconhecidamente relevante para o gestor, há divergências nesta toada. Para Carpinetti:

O entendimento predominante nas últimas décadas e que certamente representa a tendência futura é a conceituação da qualidade como satisfação dos clientes. Essa definição contempla adequação ao uso ao mesmo tempo em que contempla conformidade com as especificações do produto (CARPINETTI, 2012, p. 12).

Carpinetti (2012) desenvolveu seu pensamento da qualidade como valoração da satisfação do cliente elencando diversos atributos⁴ que conferem qualidade ao produto, o qual quando possuir qualidade deverá ser capaz (elemento intrínseco) de satisfazer o cliente, e deve ainda satisfazer faticamente o cliente (elemento extrínseco do produto o qual consiste na subjetividade de cada cliente). Noutras palavras, a potencialidade de satisfação deve se atualizar em experiência de qualidade conforme o juízo de satisfação do cliente.

Conforme Garvin (2002), a qualidade é percebida de forma pluridimensional. Ou seja, em uma percepção transcendente, a qualidade é excelência, este é seu sentido denotativo, é a apresentação da idéia de qualidade como o é por si mesma e não como atributo de um produto; em uma percepção baseada no produto, a qualidade é a correta proporção de seus ingredientes, é a harmonia de suas partes; em uma percepção baseada no usuário, a qualidade é a efetiva satisfação dos consumidores de acordo com suas necessidades; em uma percepção baseada na produção, a qualidade é o atributo do produto que é produzido conforme o especificado segundo seu planejamento; em uma percepção baseada no valor, a qualidade é o atributo do produto que possui um preço justo no mercado de acordo com seu desempenho. Assim, Garvin (2002) explorou diferentes aspectos que são igualmente válidos para uma definição abrangente de qualidade.

⁴ *In verbis*: "Desempenho técnico ou funcional, facilidade ou conveniência de uso, disponibilidade, confiabilidade, manutenibilidade, durabilidade, conformidade, instalação e orientação de uso, assistência técnica, interface com o usuário, interface com o meio ambiente, estética, qualidade percebida e imagem da marca" (CARPINETTI, 2012, pp. 12-13).

O PMBOK ao tratar do gerenciamento de projetos, definiu o que se busca quando se trata de qualidade:

O gerenciamento da qualidade do projeto inclui os processos e as atividades da organização executora que determinam as políticas de qualidade, os objetivos e as responsabilidades, de modo que o projeto satisfaça às necessidades para as quais foi empreendido. Implementa o sistema de gerenciamento da qualidade por meio de políticas e procedimentos com atividades de melhoria contínua de processos realizadas durante todo o projeto, conforme apropriado (Guia PMBOK, 2008, p.189).⁵

Satisfeita, ainda que de forma preliminar, a inquirição a respeito do conceito de qualidade para os fins almejados neste trabalho, faz-se necessária a busca pelos significados possíveis do termo "produtividade". Uma vez que a produtividade seria decorrência lógica da qualidade, como justamente se percebe a partir da literatura que se apresenta a seguir. Nesse sentido e em apertada síntese, Campos apresenta seu conceito de produtividade como segue: "Aumentar a produtividade é produzir cada vez mais e/ou melhor com cada vez menos. Pode-se, pois, representar a produtividade como o quociente entre o que a empresa produz ("OUTPUT") e o que ela consome ("INPUT")" (CAMPOS, 1999, p. 2).

A expressão em tela alcança outro enfoque na teoria de Rocha (1995) o qual apresentou o conceito de produtividade sob o seguinte raciocínio de viés eminentemente econômico: os fatores de produção (classicamente tipificados em terra, capital e trabalho) são remunerados conforme a produção que obtiverem, assim, a produtividade será a capacidade e a efetiva produção de maior volume e qualidade com o menor custo.

A produtividade leva a empresa a ser competitiva. É um conceito conexo à produtividade, como a produtividade o é da qualidade. É o que se assume a partir do magistério de Campos (1999, p. 6) que assim a definiu: "Ser competitivo é ter maior produtividade entre todos os seus concorrentes".

Para Porter (1989), a estratégia competitiva em meio à concorrência, se mostra bem sucedida quanto maior for sua atratividade e rentabilidade. Portanto, competitivo é aquela empresa que supera a produtividade e a qualidade de seus

⁵ Esta é a definição mais prática para os fins didáticos.

concorrentes, mostrando-se atrativa aos clientes ou consumidores e rentável aos seus proprietários.

Os conceitos acima abordados contribuem para o que Campos (1999) denominou "sobrevivência" da empresa, a qual precisa apresentar qualidade, produtividade e competitividade para sobreviver no mercado na qual está inserida. *In verbis*:

[...] garantir a sobrevivência de uma empresa é cultivar uma equipe de pessoas que saiba montar e operar um sistema, que seja capaz de projetar um produto que conquiste a preferência do consumidor a um custo inferior ao de seu concorrente (CAMPOS, 1999, p. 8).

Isto posto, adentra-se aos aspectos de natureza mais normativa e menos doutrinária, assim, urge mencionar e posicionar os aspectos centrais que serão relevantes para o fim aqui proposto o que significa tratar de apresentar as normas ISO, CMMI e MPS.BR.

ISO é a sigla de "*International Organization for Standardization*"⁶, que é uma organização internacional composta por 164 países e 3.368 organismos técnicos.⁷

A ISO 25010 ou ISO/IEC 25010:2011, definiu as características de qualidade que todos os softwares devem ter, que são: Funcionalidade, Performance, Compatibilidade, Usabilidade, Confiabilidade, Segurança, Manutenibilidade, Portabilidade. Essa norma revisou e superou a ISO/IEC 9126-1:2001.⁸

A norma ISO 25012 especificou um modelo para a qualidade de dados que complementa a norma ISO/IEC 25010:2011. Aquela norma abordou especificamente a qualidade de dados como sendo multidimensional, classificada em duas categorias: a primeira reúne critérios relativos à qualidade inerente dos dados, onde possa atender a uma demanda informacional a partir dos dados em si; a segunda categoria, por outro lado, dependente do sistema e presume que a qualidade do dado é melhorada ou dependente do sistema computacional em que opera. No entanto, também há casos em que ambas categorias estão presentes.⁹

⁶ Em tradução livre: Organização Internacional de Padronização.

⁷ ISO. **About ISO**. 2013. Disponível em: < <http://www.iso.org/iso/home/about.htm> >. Acesso em: 06 nov. 2013. Tradução livre.

⁸ ISO. **ISO/IEC 25010:2011 - Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models**. 2011. Disponível em: < http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=35733 >. Acesso em: 06 nov. 2013. Tradução livre.

⁹ ISO. **ISO/IEC 25012:2008 Software engineering -- Software product Quality Requirements and Evaluation (SQuaRE) -- Data quality model 2008**. Disponível em: < http://www.iso.org/iso/catalogue_detail.htm?csnumber=35736 >. Acesso em: 06 nov. 2013. Tradução livre.

Por seu turno, a norma ISO 15504 definiu os requisitos para se realizar os processos de avaliações de processos de software com foco na melhoria dos processos - gerando um perfil dos processos (identificando pontos fracos e fortes) que serão utilizados para a elaboração de um plano de melhorias - determinando a capacidade dos processos, viabilizando a avaliação de um fornecedor em potencial.¹⁰

O CMMI é um modelo de certificação que reúne as melhores práticas para o aperfeiçoamento dos processos relacionados ao desenvolvimento, serviços e aquisições em geral. Essas coleções de melhores práticas também são conhecidas como modelos de referência, pois elas são utilizadas como referência para uma organização se comparar a outra. O CMMI foi concebido pelo SEI (*Software Engineering Institute*), que é um órgão integrante da universidade norte-americana Carnegie Mellon, e seu modelo segrega o conhecimento das melhores práticas em cinco níveis de maturidade: "Inicial", "Gerenciado", "Definido", "Gerenciado Quantitativamente" e "Otimizado".¹¹ A implantação desse método possibilita os seguintes benefícios para a organização: redução de custos, cumprimentos de prazos, melhoria na produtividade, melhoria na qualidade, melhoria na satisfação do cliente, excelente retorno do investimento (CMMI INSTITUTE, 2013).

Semelhante ao CMMI, há no Brasil um modelo para "Melhoria de Processos do Software Brasileiro" (MPS.BR), cujo objetivo principal é: "Impulsionar a melhoria da capacidade de desenvolvimento de software e serviços nas empresas brasileiras".¹² Sua estrutura se dá nos seguintes níveis de maturidade: A (Em Otimização), B (Gerenciado Quantitativamente), C (Definido), D (Largamente Definido), E (Parcialmente Definido), F (Gerenciado) e G (Parcialmente Gerenciado) (SOFTEX, 2013). Esses modelos, desde a ISO até a MPS.BR, visam a qualidade, que é o seu objetivo básico e comum; salienta-se, ademais, que o modelo brasileiro

¹⁰ ISO. ISO/IEC 15504-2:2003 - Information technology -- Process assessment -- Part 2: Performing an assessment. 2003. Disponível em: < http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=37458 >. Acesso em: 06 nov. 2013. Tradução livre.

¹¹ Wibas GmbH. **Wibas CMMI Browser**. 2013. Disponível em: <<http://www.cmmi.de/>>, Acesso em 03 nov. 2013. Tradução livre.

¹² Sociedade SOFTEX, **Qualidade – MPS.BR**. 2013. Disponível em: < <http://www.softex.br/mpsbr/> > Acesso em: 06 nov. 2013.

é um esforço nacional para certificar micro, pequenas e médias empresas em um tempo justo e custo razoável (SOFTEX. 2012).¹³

Por fim, como conceito preliminar, Campbell e Papapetrou (2013) afirmaram que o SonarQube é uma plataforma livre para gerenciar qualidade de código. Ele possui uma grande quantidade de recursos em seu núcleo, tais como: indicação de potenciais defeitos, regras de codificação, etc. Para esses autores, o SonarQube não apenas demonstra o erro, antes, oferece ferramentas para o gerenciamento da qualidade visando o acerto. Assim, passa-se ao exame propriamente dito da ferramenta em tela e como ela satisfaz o conceito e as exigências da gestão da qualidade.

2 Relato de experiência

Passa-se aqui, de forma mais detida, responder a questão que motiva este trabalho: Pode-se realizar o gerenciamento da qualidade do código desenvolvido por meio de análises de verificação e monitoramento por meio da ferramenta SonarQube? Para se responder o problema erguido, foi realizada uma pesquisa qualitativa e quantitativa cujo instrumento de pesquisa foi a análise documental da ferramenta SonarQube. Para tanto privilegiou-se a documentação disponível no portal da ferramenta - <http://www.sonarqube.org/> - destacando seu canal de exemplo - <http://nemo.sonarqube.org>. Em especial, mas não apenas, para exemplificar os conceitos e demonstrar os indicadores que permitirão responder à questão erguida, usou-se o projeto *open source*¹⁴¹⁵ OpenEJB¹⁶ que foi adotado por cobrir todos os eixos os quais serão a seguir explicitados. Tal projeto, em virtude da amplitude da ação das ferramenta não será aprofundado, mas foram abordados os elementos que permitirão responder à problematização suscitada.

¹³ Sociedade SOFTEX, **MPS.BR - Melhoria de Processo do Software Brasileiro: Guia Geral MPS de Software**. 2012. Disponível em: < http://www.softex.br/wp-content/uploads/2013/07/MPS.BR_Guia_Geral_Software_2012.pdf p.4.>, Acesso em 06 nov. 2013.

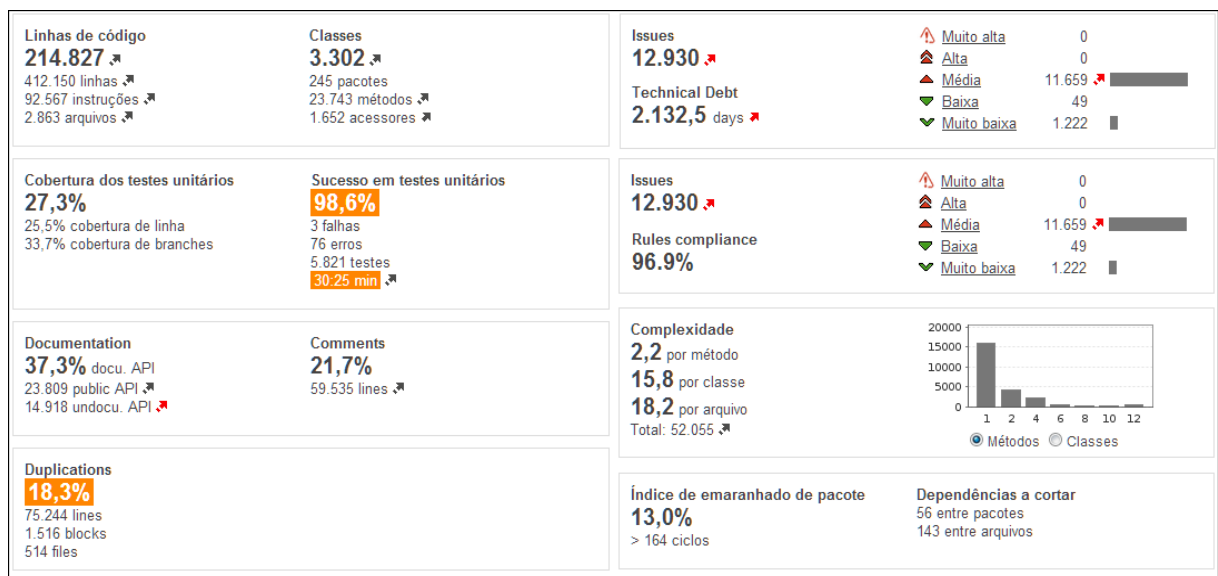
¹⁴ Open Source Initiative. **The Open Source Initiative**. 2008. Disponível em: < <http://www.opensource.org> > Acesso em: 06 nov. 2013. Tradução livre.

¹⁵ Ou seja, é um software livre conforme a *Open Source Initiative*.

¹⁶ SonarSource SA. **SonarQube Nemo**. 2013. Disponível em: < <http://nemo.sonarqube.org/dashboard/index/108716> > Acesso em: 12 nov. 2013.

Essa pesquisa voltou-se para os eixos fundamentais da ferramenta SonarQube¹⁷, que são sete desdobramentos de sua ação e que são elencados como segue: análise de *design* e arquitetura do projeto, duplicações de códigos, cobertura dos testes unitários, análises de distribuição da complexidade, potenciais falhas (*bugs*), codificação sem padronização e excesso ou ausência de comentários em códigos. A figura a seguir apresenta tais eixos.

Figura 1 - Dashboard inicial do projeto EJB Open na ferramenta SonarQube.



Fonte: <http://nemo.sonarqube.org/dashboard/index/108716>

O primeiro eixo, análise de *design* e arquitetura, como o nome sugere, trata do monitoramento durante a implementação do projeto para que esse não perca sua estrutura e desenho inicialmente planejados.

Figura 2 - Análise de *design* e arquitetura



Fonte: <http://nemo.sonarqube.org/dashboard/index/108716>

O que se observa na tabela é que há uma interdependência entre 13% dos pacotes desenvolvidos. A ferramenta, portanto, permitiu visualizar este emaranhado que causaria uma maior dificuldade em realizar a manutenção. Uma vez que há

¹⁷ Esses eixos também sofreram análise privilegiada por Campbell e Papapetrou, o que demonstra sua pertinência e relevância.

diminuição de interdependência entre pacotes e arquivos, então há ganhos de custo e de tempo gerando maior produtividade por meio de um projeto com boa arquitetura e assim, mais competitivo.

O segundo eixo, duplicação de códigos, trata, como sugere seu nome, sobre o mal reaproveitamento do código desenvolvido. Ele facilita, portanto, as futuras manutenções por indicar tais duplicações; o que doutra sorte exigiria uma duplicidade no esforço para a correção de eventuais problemas. Na figura abaixo, percebe-se a quantidade de código duplicado no projeto desenvolvido.

Figura 3 - Duplicação de códigos

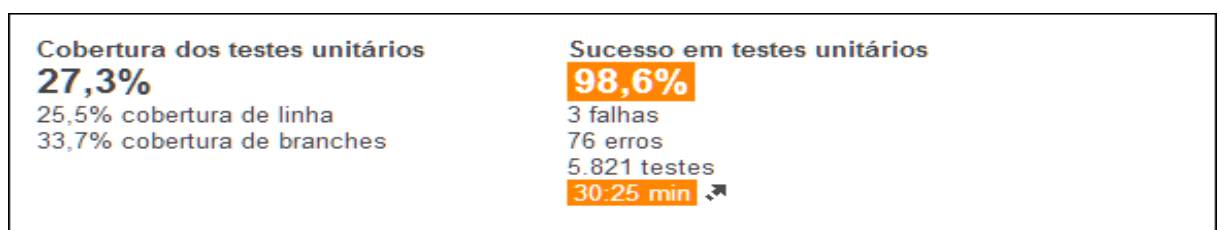


Fonte: <http://nemo.sonarqube.org/dashboard/index/108716>

Se a ferramenta for utilizada, então haverá diminuição de custos, o que implica maior produtividade¹⁸. E maior produtividade proporciona incremento da competitividade.

O terceiro eixo é a cobertura de testes unitários. Ou seja, é quando se divide o todo em partes e os testa separadamente a fim de garantir a qualidade das partes desenvolvidas visando atingir a excelência total do produto. Na figura abaixo, demonstra-se como a ferramenta atua.

Figura 4 - Cobertura de testes unitários



Fonte: <http://nemo.sonarqube.org/dashboard/index/108716>

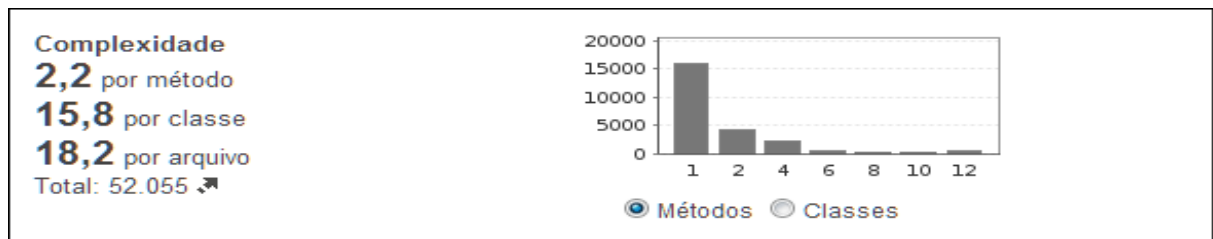
Percebe-se que há uma amostragem no exemplo supra, a qual permite inferir uma qualidade no todo. Sendo que a qualidade é uma dimensão da satisfação do

¹⁸ O conceito de produtividade é apresentado por Campos neste trabalho e adequa-se a este eixo.

cliente¹⁹, a ferramenta permitiu analisar e constatar que o produto possui alguns (ainda que poucos) defeitos, que conhecidos antes da entrega, serão corrigidos. Isso deve gerar maior qualidade para o cliente, e assim melhor competitividade para a empresa.

O quarto eixo é a análise de distribuição da complexidade. Para que se visualize como a ferramenta funciona neste eixo, uma comparação parece ser útil: é sabido que, normalmente, o desenvolvedor mais inexperiente implementa menos métodos que acumulam mais funções tornando-o complexo; enquanto o mais experiente consegue implementar mais métodos com menos funções, tornando-os mais simples, ou seja, não complexos. A ferramenta SonarQube identifica métodos muito complexos, auxiliando o desenvolvedor a dividir as diversas funções em mais métodos.

Figura 5 - Análises de distribuição da complexidade



Fonte: <http://nemo.sonarqube.org/dashboard/index/108716>

Para o julgamento a respeito da métrica na qual se define o nível de complexidade e, a *contrario sensu*, da simplicidade, remete-se ao portal da ferramenta SonarQube.²⁰ A partir de tal portal, constatou-se que o projeto na figura acima tem nível baixo de complexidade dos métodos (SONARSOURCE S.A., 2013). Implementando métodos menos complexos, a manutenção restará facilitada e já fora reconhecido que a manutenibilidade²¹ é um dos atributos da qualidade segundo Carpinetti (2012).

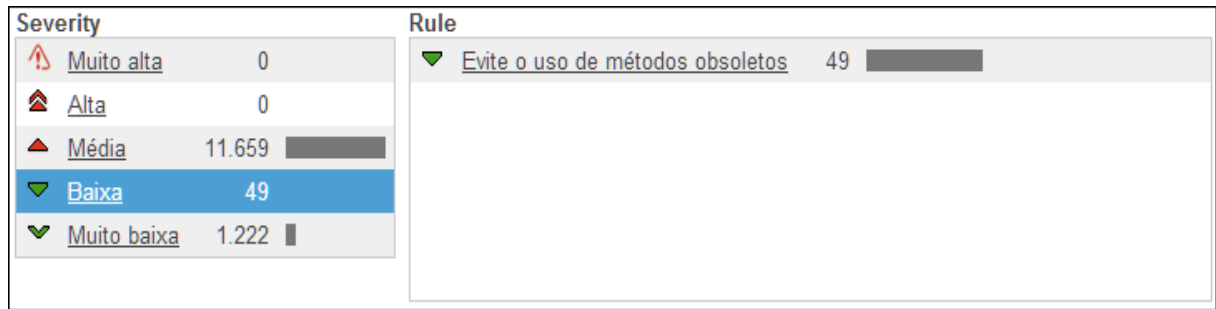
O quinto eixo, potenciais falhas ou potenciais *bugs*, representa eventuais problemas que podem ocorrer futuramente no código, ou seja, tem a imprevisibilidade por principal marca. Apresenta cinco níveis de severidade: muito alta, alta, média, baixa e muito baixa. Na figura abaixo, tem-se:

¹⁹ Considerando a pluridimensionalidade conceitual de qualidade (GARVIN, 2002).

²⁰ SonarSource SA. **Abacus Plugin**. 2013. Disponível em: < <http://docs.codehaus.org/display/SONAR/Abacus+Plugin> >.

Acesso em: 12 nov. 2013.

²¹ CARPINETTI, 2012, p. 12.

Figura 6 - Potenciais *bugs*

Fonte: <http://nemo.sonarqube.org/dashboard/index/108716>

De todas as linhas de código acima, há 11.659 ocorrências que são de severidade média, o que significa que essas exigem uma atenção por parte do desenvolvedor a fim de evitar falhas no sistema desenvolvido. Se falhas são evitadas, logo os custos são reduzidos, por isso é correto afirmar que há ganhos em produtividade. Além disso, o produto se torna mais confiável atendendo, portanto, ao que preceitua Carpinetti (2012).

O sexto eixo, codificação sem padronização, como o nome sugere, trata de analisar padrões de codificação, bem como de testes unitários aferindo se ambos estão em conformidade com os perfis de qualidade definidos no SonarQube²². A figura a seguir ilustra o que se afirma:

Figura 7 - Codificação sem padronização



Fonte: <http://nemo.sonarqube.org/dashboard/index/108716>

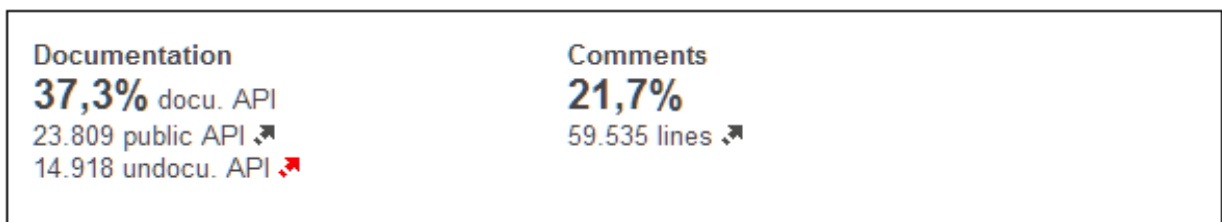
Na imagem acima, percebe-se que 96,9% do código desenvolvido está em conformidade com as regras estabelecidas na ferramenta SonarQube. Isto demonstra que o planejado foi concretizado. Assim, a qualidade que foi projetada tornou-se aferível por meio da ferramenta. Não foi a ferramenta que tornou possível a qualidade, neste sentido: o projetista a concebeu, entretanto, a ferramenta

²² Estes perfis podem ser personalizados para atender novas regras, as quais são definidas pelos desenvolvedores que sejam usuários do SonarQube, para padronização de códigos. A definição de tais regras pode ser mais profundamente visualizada em: <http://docs.codehaus.org/display/SONAR/Quality+Profiles>.

mensura o que foi implementado. Já fora assinalado, também, que a qualidade é obtida por meio da concretização do que foi planejado, e isto se dá por um gerenciamento que efetive o escopo (GUÍA PMBOK, 2008). Para tanto, é necessário corrigir desvios, e nisto a ferramenta se mostrou aderente.

O sétimo eixo, excesso ou ausência de comentários, cuida da criação de comentários explicativos das funções e ou métodos codificados. O comentário permite entender a função, gerando conhecimento que seja transparente a outros desenvolvedores tornando, assim, a curva de aprendizagem menor e a atividades de manutenção mais simples. Isso facilita a adaptação de novos recursos humanos e diminui custos. Por outro lado, o excesso de comentários em código pode causar uma má performance no sistema além de torná-lo confuso. O que se busca é o equilíbrio. Na figura abaixo, observa-se:

Figura 8 - Excesso ou ausência de comentários



Fonte: <http://nemo.sonarqube.org/dashboard/index/108716>

No projeto acima, 21,7% das linhas de código são comentários. O equilíbrio em seu emprego pareceu existente²³. Assim, o conceito de qualidade como uma virtude pluridimensional de Garvin (2002) é vislumbrada, uma vez que há uma harmonia entre conhecimento documentado que é um benefício do produto em si sem a perda da boa performance do sistema, permitindo seu bom uso.

O SonarQube, percebido por seus eixos em separado, mostrou-se eficaz na mensuração, avaliação e monitoramento da qualidade dos processos para o desenvolvimento do *software*.²⁴

Pelas razões acima apresentadas, é possível concluir que a resposta ao problema deve ser positiva. Não se ignora, todavia, que a qualidade é um conceito não estático em função de aspectos humanos, notadamente culturais, temporais e

²³ Não se ignora que há uma subjetividade na consideração sobre qual equilíbrio seria razoável. Desenvolvedores, equipes ou projetistas diferentes tendem a perceber o equilíbrio de forma distintas conforme as circunstâncias de cada um.

²⁴ A análise da ferramenta SonarQube como conjunto, é recomendação que se faz para futuras pesquisas.

geográficos. Nem se ignora também que há problemas ainda não resolvidos pela ferramenta estudada e que não foram explorados aqui em razão do objetivo da pesquisa empreendida bem como da amplitude de perspectivas de estudo possíveis.²⁵

Considerações finais

O objetivo da pesquisa foi apresentar instrumentos e estratégias para realizar o gerenciamento da qualidade no desenvolvimento de *software*. Para tanto, apresentou-se revisão de literatura sobre o que se entenderia por qualidade e conceitos conexos além de conceituar normas e certificações de boas práticas nos processos de implementação de sistemas.

Problematicou-se se a ferramenta SonarQube seria hábil para atingir o objetivo proposto. Após pesquisa documental apresentada no relato de experiência que se seguiu, constatou-se que a resposta deveria ser positiva.

Em vista do objetivo, o aspecto interdisciplinar cujo enfoque foi aqui privilegiado não permitiu um aprofundamento em outras perspectivas e aspectos técnicos da ferramenta apresentada, tal como algumas métricas que poderiam ser aprofundadas no relato de experiência mas que foram postas à margem para que o artigo atingisse seu escopo.

Uma escassez de literatura científica a respeito da ferramenta foi percebida durante a realização das pesquisas empreendidas, o que gerou dificuldades em sua realização. O que suscita, portanto, a recomendação de pesquisas futuras sobre a ferramenta SonarQube em uma perspectiva que aprofunde seu enfoque no gerenciamento da qualidade e ainda que analise seus efeitos a partir de um prisma global de sua atuação.

Ora, salienta-se, porém, que a escassez mencionada de literatura encontra sua razão de ser na atualidade do tema, pois a ferramenta é bastante contemporânea, ainda pouco conhecida e pouco explorada sob o prisma teórico²⁶. É relevante, oportuna e atual a discussão a respeito do tema. E isso o é não apenas

²⁵ É autoevidente que não haja ferramenta que tudo solucione porque isso não coaduna com o desenvolvimento intelectual e tecnológico da humanidade nem com a pesquisa científica.

²⁶ A literatura mais relevante sobre o tema sequer está traduzida para o português. Trata-se de publicação que ainda é vendida em inglês em formato digital uma vez que a forma física ainda não está disponível.

sob o enfoque acadêmico, mas é forçoso recomendar o uso da ferramenta nos setores profissionais afeitos à área de desenvolvimento de software.

Referências bibliográficas

CAMPBELL, Ann; PAPAPETROU, Patroklos. **SonarQube in Action**. Shelter Island: Manning, 2013.

CAMPOS, Vicente Falconi. **Controle da Qualidade Total** (no estilo japonês). Belo Horizonte: Editora de Desenvolvimento Gerencial, 1999.

CARPINETTI, Luiz Cesar Ribeiro. **Gestão da qualidade: conceitos e técnicas**. 2ed. São Paulo: Atlas, 2012.

GARVIN, David A. **Gerenciando a qualidade: a visão estratégica e competitiva**. Trad. João Ferreira Bezerra de Souza. Rio de Janeiro: Qualitymark, 2002.

Guia PMBOK. **Um Guia do Conhecimento em Gerenciamento de Projetos**. 4. ed. EUA: Project Management Institute, 2008.

ISO. **About ISO**. 2013. Disponível em: < <http://www.iso.org/iso/home/about.htm> >. Acesso em: 06 nov. 2013. Tradução livre.

_____. **ISO/IEC 25010:2011** - Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models. 2011. Disponível em: < http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=35733 >. Acesso em: 06 nov. 2013.

_____. **ISO/IEC 25012:2008** - Software engineering -- Software product Quality Requirements and Evaluation (SQuaRE) -- Data quality model. 2008. Disponível em: < http://www.iso.org/iso/catalogue_detail.htm?csnumber=35736 >. Acesso em: 06 nov. 2013.

_____. **ISO/IEC 15504-2:2003** - Information technology -- Process assessment -- Part 2: Performing an assessment. 2003. Disponível em: < http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=37458 >. Acesso em: 06 nov. 2013.

PALADINI, Edson Pacheco. **Gestão da qualidade: teoria e prática**. 2 ed. São Paulo: Atlas, 2004.

PORTER, Michael. **Vantagem Competitiva: criando e sustentando um desempenho superior**. Trad. Elizabeth Maria de Pinho Braga. Rio de Janeiro, Qualitymark, 1989 (27ª reimpressão).

ROCHA, Duílio. **Fundamentos técnicos da produção**. São Paulo: Makron Books, 1995.

SOFTTEX. **MPS.BR - Melhoria de Processo do Software Brasileiro: Guia Geral MPS de Software**. 2012. Disponível em: < <http://www.softtex.br/wp->

content/uploads/2013/07/MPS.BR_Guia_Geral_Software_2012.pdf >. Acesso em : 06 nov. 2013.

_____. **Qualidade - MPS.BR**. 2013. Disponível em : < <http://www.softex.br/mpsbr/> >. Acesso em : 06 nov. 2013

SONARSOURCE SA. **SonarQube**. 2013. Disponível em : < <http://www.sonarqube.org/> >. Acesso em : 06 nov. 2013.

_____. **Nemo SonarQube**. 2013. Disponível em : < <http://nemo.sonarqube.org> >. Acesso em : 06 nov. 2013.

_____. **Abacus Plugin**. 2013. Disponível em : < <http://docs.codehaus.org/display/SONAR/Abacus+Plugin> >. Acesso em : 06 nov. 2013.

WIBAS GMBH. **wibas CMMI Browser** [sic]. 2013. Disponível em : < <http://www.cmmi.de/> >. Acesso em 03 nov. 2013.