

INSPEÇÃO DE SOFTWARE COMO FERRAMENTA DE APOIO AO ENSINO: UMA PROPOSTA PARA ANÁLISE DE CÓDIGO-FONTE

Yury Alencar Lima¹

Elder De Macedo Rodrigues²

Maicon Bernardino³

Resumo:

Este trabalho tem por objetivo prover suporte ao ensino de Programação Orientada a Objetos (POO) em cursos de graduação da área da Ciência da Computação. O ensino de POO é um desafio para os professores, em especial pelo volume de linhas de códigos para revisar durante as aulas e na avaliação de trabalhos e provas. Além de disso, é um desafio detectar quais são os maiores déficit técnicos de um turma ou grupos de alunos. Com o objetivo de mitigar tais limitações e prover suporte a aprendizagem de POO este trabalho propõe o uso de técnicas de análise estática do código para avaliar a qualidade do código desenvolvido pelos alunos. Na execução deste trabalho foi utilizada a ferramenta de análise estática SonarQube. A ferramenta fornece uma ampla gama de funcionalidades, dentre elas podemos citar: Gestão de grupo, análise de código, classificação de defeitos, análise de complexidade de classe e funções, criação de métricas de análise, dentre outras. A aplicação da ferramenta durante a execução do trabalho permitiu aos professores criarem métricas com o objetivo de definir padrões a serem seguidos pelos alunos. Assim, caso ocorra alguma discrepância entre a especificação e o código, o erro é detectado e o professor avisado. Além disso, a ferramenta possui integração com o GitHub, possibilitando assim identificar qual aluno originou o erro, possibilitando ao professor trabalhar juntamente com o aluno na resolução. Nos casos de avaliações, como provas e trabalhos práticos, o professor pode visualizar facilmente as discrepâncias existentes entre o projeto e a especificação. Os resultados apresentados pela ferramenta permitem que o professor ajuste o conteúdo e formato de suas aula, com o objetivo de se mitigar o déficit técnico da turma. Outra vantagem do uso de ferramentas de inspeção é a melhora da qualidade da implementação por parte dos alunos, pois ele terá um feedback constante sobre o uso adequado dos padrões e até mesmo se a complexidade do código está aceitável. O ensino de Orientação a Objetos acaba focando em erros comuns praticados, entretanto nem todas as turmas apresentam as mesmas dificuldades, um ponto a ser salientado também é a complicação de fornecer atenção individual aos grupos. Com o uso da ferramenta é possível avaliar grupos de forma mais individuais e trabalhar a evolução de acordo com os problemas apresentados pelos mesmos. Com isto é possível destacar uma melhora da aprendizagem do ensino de Orientação a Objetos (OO) , em que os alunos por sua vez também terão contato com outras tecnologias utilizadas no âmbito profissional.

Palavras-chave: Ensino de Programação Orientada a Objetos, Qualidade de aprendizado, Gestão de alunos

Modalidade de Participação: Iniciação Científica

INSPEÇÃO DE SOFTWARE COMO FERRAMENTA DE APOIO AO ENSINO: UMA PROPOSTA PARA ANÁLISE DE CÓDIGO-FONTE

¹ Aluno de graduação. yuryalencar19@gmail.com. Autor principal

² Docente. elderrodrigues@unipampa.edu.br. Orientador

³ Docente. bernardino@acm.org. Co-orientador



INSPEÇÃO DE SOFTWARE COMO FERRAMENTA DE APOIO AO ENSINO DE PROGRAMAÇÃO ORIENTADA A OBJETOS: UMA PROPOSTA PARA ANÁLISE DE CÓDIGO-FONTE DE PROJETOS DE PROGRAMAÇÃO

1. INTRODUÇÃO

Atualmente, a criação de software possui uma série de padrões de desenvolvimento, arquitetura, convenções de código dentre outros. Estes têm como finalidade melhorar a comunicação entre toda a equipe, impactando também no entendimento do artefato criado e, conseqüentemente, na sua manutenibilidade. Com a finalidade de verificar essas normas, pode-se realizar a execução de inspeções que são geralmente baseadas em *checklist* e/ou outras técnicas. Como estas atividades são realizadas por meio de análises estáticas, as buscas por problemas acabam levando um tempo considerável. A ampla utilização de *checklist*, provoca com que os erros encontrados sejam semelhantes ou idênticos, mesmo que mais de um profissional realize o procedimento.

Devido a isto uma série de ferramentas foram desenvolvidas com a finalidade de realizar estas análises de forma automática, beneficiando o desenvolvedor e a equipe. Dentre estas ferramentas a SonarQube [CAMPBELL, 2017a] se mostrou eficiente no quesito análise e gestão do grupo, fornecendo gráficos com focos de defeitos encontrados, análises de complexidade do código gerado, e muitas outras funcionalidades, além de permitir a criação de métricas a serem utilizadas na análise. Provendo assim, a oportunidade de inserção de métricas específicas relacionadas à empresa utilizatória da ferramenta [CAMPBELL, 2017a].

A correção de trabalhos, tarefas dentro do ensino de programação muitas vezes é realizada por meio de análise estática destes itens. Entretanto, ainda é uma tarefa não trivial detectar a maior *déficit* de uma turma específica, pois como a análise é um processo que demanda tempo acaba se tornando inviável tal atividade. Sabendo disto, o foco do professor na maioria das vezes é a correção das dificuldades mais encontradas. Com isto o aprendizado do aluno acaba muitas vezes prejudicado, já que o esforço do professor em muitas ocasiões pode acabar direcionado a um *déficit* que a maioria dos alunos da classe não possui.

Com a finalidade de melhorar a aprendizagem dos alunos, agilizar o processo de correção de trabalhos entre outras melhorias, este artigo é composto de uma proposta de utilização de inspeção automatizada dentro do contexto de ensino-aprendizagem. Essa automação vem com a proposta de uso da ferramenta SonarQube, em que é possível gerir o grupo, analogamente seria a turma, ou os grupos da classe. Um fator importante da ferramenta é a possibilidade de inserção de novas métricas, com isso o professor pode criar um conjunto de medidas com a finalidade de validar os conteúdos expostos na disciplina. Outra funcionalidade importante também é a possibilidade de gerar gráficos baseados na quantidade de erros da equipe, verificando quais são os pontos de maiores dificuldades da turma. Assim, focando os esforços na melhoria e recuperação de defasagem na aprendizagem da classe.

Com o objetivo de apresentar esta proposta de uso e melhoria do ensino este artigo é composto pela Seção 2, que apresenta a como pode ser utilizada a



ferramenta com a finalidade de verificar quais são déficits da turma. Seguido da mesma contém a Seção 3, a qual mostra as melhorias e desvantagens com a adoção desta proposta. Finalizando, a Seção 4 apresenta os trabalhos futuros envolvendo esta proposta.

2. METODOLOGIA

Esta parte é composta por quatro etapas cruciais para a garantia da qualidade dos resultados finais, em que duas das mesmas são relacionadas a definição de métricas e configuração de projeto, e as outras relacionadas à gestão da turma. A primeira etapa explora a possibilidade de criação de métricas com a finalidade de validar entendimento da turma em relação à matéria de Programação Orientação a Objetos (POO) [RICARTE, 2001]. Assim, juntamente com o professor da disciplina é definido os padrões a serem detectados pela ferramenta classificando também em níveis diferentes de erros. Desta forma, é possível detectar problemas relacionados a programação e arquitetura, caso ocorra alguma discrepância do que foi especificado. A segunda etapa é formada pela configuração dos projetos, em que o professor juntamente com os alunos poderão escolher de acordo com a linguagem de programação qual tecnologia utilizar juntamente com o versionamento do código. Dentre as tecnologias compatíveis estão o Maven [O'BRIEN *et al.*, 2008], Gradle [BERGLUND, 2013] e muitas outras, entretanto é possível realizar o uso do SonarQube sem a integração. A ferramenta também possui compatibilidade com o GitHub [PIPINELLIS, 2015], possibilitando assim verificar o histórico de problemas no código, dentre estes inclusive quais alunos deram origem ao erro. Com isto, é possível também saber quais destes alunos também corrigiram os defeitos caso ocorra refatorações dentro do projeto.

A partir da terceira etapa do processo de avaliação é utilizada a parte da ferramenta que possibilita o gerenciamento do grupo. Então, o professor deve adicionar as devidas permissões a cada aluno alocando os mesmos dentro dos seus devidos projetos. Outra funcionalidade presente é o envio de e-mails diretamente a partir da ferramenta, designando correções de erros, em que o orientador pode avisar aos alunos no caso de problemas graves durante o desenvolvimento do trabalho. A última etapa visa verificar a análise dos resultados de cada grupo, e assim verificar o maior grau de defasagem na aprendizagem de cada grupo. Esta análise pode ser realizada por meio de gráficos de bolhas de cada projeto providos pela ferramenta, em que também é possível verificar a quantidade de erros, e quais alunos estão envolvidos. Ainda na etapa de verificação de resultados é possível verificar outros problemas como complexidade de classes e funções, em que é possível analisar a manutenibilidade do código do aluno. Assim, melhorar também o desenvolvimento do educando mesmo que não possua erros.

3. RESULTADOS E DISCUSSÃO

Este trabalho ainda está em desenvolvimento, e será aplicado nas disciplinas de Orientação a Objetos (OO) das turmas de Ciência da Computação e Engenharia de Software. Assim, ainda não possui resultados concretos, entretanto um fator importante é que pode facilitar e melhorar o desempenho da turma juntamente com o acompanhamento do desenvolvimento dos projetos. Outro ponto importante é que



para isto os alunos poderão fazer uso de outras tecnologias melhorando também no âmbito profissional. Entre as melhorias possíveis também se encontra a redução do tempo de correção por partes dos professores, desde que os padrões sejam devidamente especificados. Com a finalidade de garantir a cobertura dos erros necessários o professor da disciplina pode gerar projetos que contém os defeitos especificados e efetuar a inspeção com a ferramenta.

4. CONSIDERAÇÕES FINAIS

Até o momento com o conhecer e uso da ferramenta a mesma possui um grande potencial no uso profissional e possivelmente no meio educacional se devidamente configurada. Tendo em vista que a mesma está sendo aplicada ao final da disciplina de POO, como trabalho futuro será realizada a comparação dos resultados finais das turmas que não fizeram o uso da mesma como apoio com as que utilizaram desde o início da matéria. Assim, avaliando o potencial da ferramenta relacionada à gestão de qualidade também no meio educacional.

Outra aplicação da ferramenta irá acontecer na disciplina de Resolução de Problemas IV (RP4), já que matéria possui o foco nos testes e refatoração é possível verificar a qualidade do código final entregue pelos grupos. Assim, utilizando o SonarQube com métricas mais avançadas, não somente relacionadas a Orientação a Objetos, mas também com erros lógicos, grau de complexidade, vulnerabilidades dentre outras verificações.

5. REFERÊNCIAS

CAMPBELL, A. (2017a). Sonarqube - Documentation. Disponível em: <https://docs.sonarqube.org/display/SONAR/Documentation>. Acesso em: 11/09/2017.

PIPINELLIS, A. (2015). *GitHub Essentials*. Packt Publishing.

BERGLUND, Tim. *Gradle Beyond the Basics: Customizing Next-Generation Builds*. "O'Reilly Media, Inc.", 2013.

CAMPBELL, A. (2017b). Sonarqube - Metric Definitions. Disponível em: <https://docs.sonarqube.org/display/SONAR/Metric+Definitions>. Acesso em: 09/09/2017.

RICARTE, Ivan Luiz Marques. *Programação Orientada a Objetos: uma abordagem com Java*. Universidade Estadual de Campinas, Campinas, SP, Brasil, 2001.

O'BRIEN, Tim et al. *Maven: The Definitive Guide is a book about Apache Maven*. Number ISBN 10: 0-596-51733-5. 2008.