



FDQ-TOOL UMA FERRAMENTA PARA A ANÁLISE DA QUALIDADE DA MÉTRICA DE ESFORÇO DO PDS

Marcos Mitsuo Ashihara¹, Nelson Tenório Jr², Rita Cristina Galarraga Berardi³

RESUMO: Este trabalho propõe a construção de uma ferramenta que apoie o modelo de avaliação da qualidade das métricas de esforço de software baseado na proposta de Berardi [2009]. A ferramenta tem o intuito de agilizar e facilitar o processo de coleta, consolidação e análise da qualidade dos dados advindos do desenvolvimento de software e com isso auxiliar os gestores na tomada de decisão deste processo por meio de um BI.

PALAVRAS-CHAVE: Avaliação; Esforço; Métricas; Qualidade.

1 INTRODUÇÃO

Quanto tempo durará o projeto? E quanto custará o projeto para a organização? Esses são os questionamentos das organizações quando estas necessitam de desenvolvimento, manutenção, aquisição ou customização de um software. Mas, no início do projeto ainda no levantamento de requisitos, não se conhece as características do produto que possa afirmar o seu tamanho final, então é imprescindível realizar estimativas de forma a reduzir os problemas relacionados à definição de tamanho e custo de software [VAZQUEZ; SIMÕES; ALBERT, 2008].

De acordo com Pressman [2006], a estimativa de esforço, tempo e custo nascem do planejamento do projeto de software. Na maioria das vezes as estimativas são orientadas unicamente pelos históricos passados, caso um novo projeto for comparável em formato de tamanho e função, então esse projeto poderá exigir aproximadamente o mesmo esforço, tempo e valor.

O gerente de projeto deve garantir que o nível determinado de qualidade do produto seja alcançado, e o gerente de qualidade estabelece procedimentos e padrões a serem seguidos durante o desenvolvimento de software, além de verificar sua realização [SOMMERVILLE, 2007]. Nas organizações em que utilizarem as medidas de produtividade e a qualidade avaliada, os gestores poderão aplicar metas de melhorias no processo de desenvolvimento de software (PDS), e se nesse caso obtiverem melhorias, um resultado positivo para a sua administração é alcançado [PRESSMAN, 2006].

¹Acadêmico do Curso de Sistema de Informação do Centro Universitário de Maringá – CESUMAR, Maringá – Paraná. Bolsista do Programa de Institucional de Bolsas de Iniciação Científica em Desenvolvimento Tecnológico e Inovação do PIBITI/CNPq. marcosashihara@gmail.com.

²Orientador, Coordenador - Centro Universitário de Maringá – CESUMAR, Maringá – Paraná. nelson.tenorio@cesumar.br

³Co-orientadora - PUC-Rio. rberardi@inf.puc-rio.br

Nas grandes organizações existem problemas com a Qualidade de Dados (QD), pois se preocupam comumente mais com a quantidade do que com a QD. Portanto, se a tomada de decisão for embasada em dados de baixa qualidade, afeta diretamente nos custos [BERARDI, 2009].

No PDS são coletadas diversas métricas, entretanto as métricas de esforço merece um cuidado maior, pois exerce um grande impacto na gestão de projetos, por medir o tempo que a equipe desenvolveu, revisou e/ou consertou um produto. A obtenção dos dados não é automatizada para citar as falhas nas medições, assim os dados de esforço são registrados manualmente por uma pessoa da equipe, contudo sem o compromisso de registrar no final de cada atividade. As avaliações de dados de esforço coletados são colocadas em diversos contextos, formatos, e diferentes finalidades de coleta, nesse aspecto aumentando assim a complexidade na definição, tornando-se impreciso aplicar uma avaliação dicotômica, o conceito de “bom” ou “ruim” [BERARDI, 2009].

2 MODELO PROPOSTO POR BERARDI

Berardi [2009] propõe um modelo para avaliar a QD coletados durante o PDS. Na tentativa de melhorar o processo de coleta. Para tanto Berardi [2009] sugere o modelo apresentado na Figura 1:

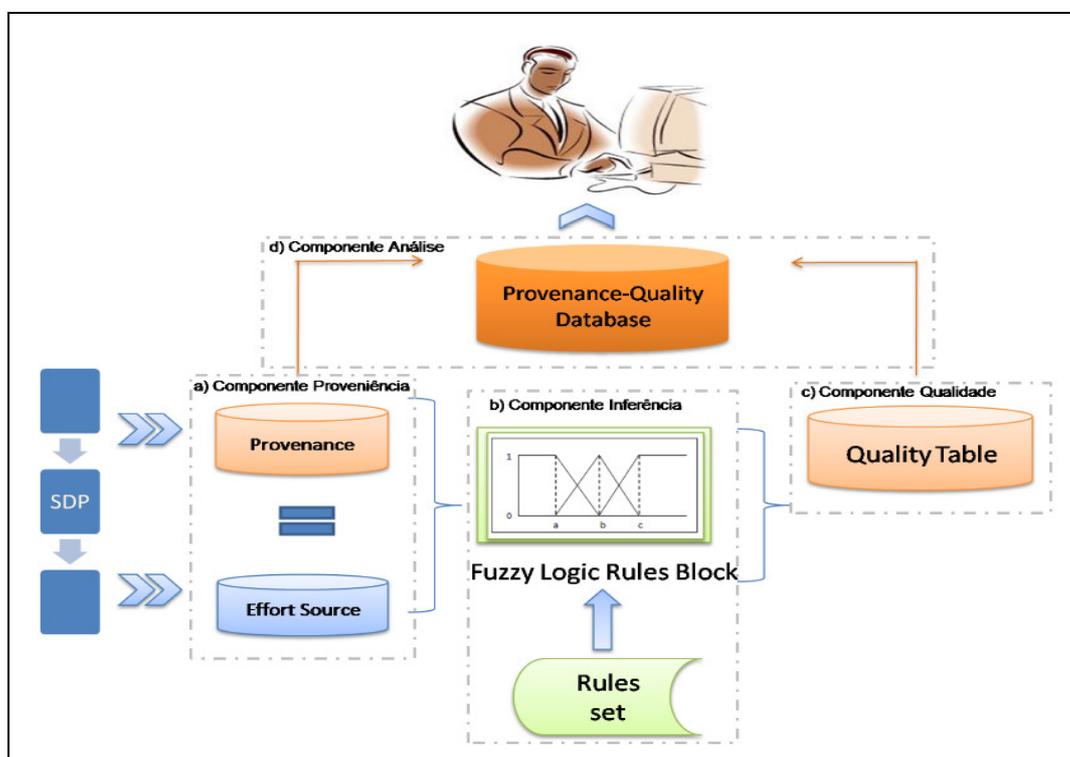


Figura 1 – Modelo proposto por Berardi [2009]. Fonte: [BERARDI, 2009].

O modelo é estruturado em quatro componentes de ação: (1) proveniência, (2) inferência, (3) qualidade e (4) análise. O processo é baseado na métrica de esforço, que ocorre durante o PDS e a medição em horas. As caixas da esquerda representam PDS. O componente de proveniência é formado por dois bancos de dados: *Effort Source* armazenam horas trabalhadas e informações de origem e *Provenance* foi construído e estruturado para grava o rastreamento das alterações dos registros do *Effort Source*. No componente de inferência ocorre à avaliação dos dados da métrica de esforço utilizando à lógica fuzzy. A lógica fuzzy funciona em ambientes de incerteza e imprecisões, demonstrando que possuem um papel de habilidade humana em tomar decisões

racionais nesses ambientes [Belchior, 1997]. O componente de qualidade é o banco de dados *Quality Table* onde são armazenados a QD de esforço que são os resultados da inferência dos dados. O componente de análise é o banco de dados *data warehouse* onde são gravados a QD de esforço e sua origem, e é de onde que se pode ser analisados os históricos de qualidade de esforço pelos gestores para auxiliar nas tomadas de decisões.

3 FDQ-TOOL (FUZZY DATA QUALITY – TOOL)

O modelo de Berardi [2009] foi devidamente testado por experimentos controlados obtendo resultados satisfatórios conforme relatos pela própria autora: “Foi atingido o objetivo principal dos testes de averiguar se a avaliação atende aos requisitos, simulando testes em um ambiente real e fazendo uso de dados reais. Os resultados obtidos e as análises realizadas mostraram que a pesquisa apresenta resultados satisfatórios e é promissora” [BERARDI, 2009].

No entanto, a forma como foi inicialmente implementado, torna sua utilização improvável em ambientes reais, devido ao envolvimento de um grande esforço manual. Por isso uma ferramenta que possibilita a automação das tarefas inerentes ao funcionamento do modelo proposto por Berardi [2009] oferecendo um ambiente de proveniência e análise dos dados coletados do PDS.

A FDQ-TOOL será implementada sob a GPL (*General Public License*) e seu código será aberto. Logo, as ferramentas utilizadas para seu desenvolvimento também serão baseadas em software livre.

Como base de dados será utilizado o PostgreSQL 9.0, como suíte de ferramenta o Eclipse IDE for Java EE Developers, como linguagem Java e, por fim, como suporte ao Fuzzy, a classe Java JFuzzyLogic é a definição de um BI (Business Intelligence) livre para interface com o usuário. Nessa primeira versão a ferramenta será desktop, pois a curva de aprendizado é menor e a quantidade de recursos oferecidos é maior.

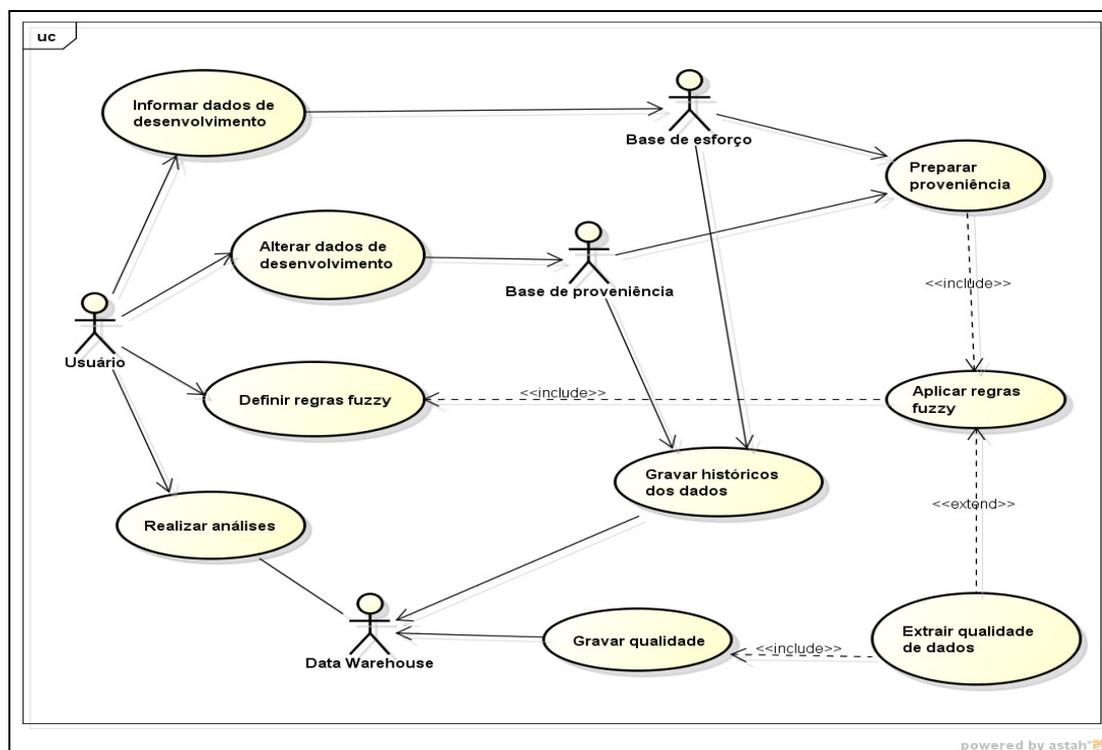


Figura 2 – Diagrama de caso de uso do modelo de Berardi [2009].

A Figura 2 ilustra o diagrama de caso de uso baseado no modelo de Berardi [2009]. Esse diagrama mostra como são distribuídos, com quem a ferramenta vai interagir e as funcionalidades para saber o que ela vai fazer, permitindo verificar o comportamento de cada um deles e suas relações promovendo o entendimento do modelo e desenvolver a ferramenta FDQ-TOOL. Assim, é possível observar a divisão das etapas das ações da ferramenta interagindo com os atores. O ator usuário realiza a ação “Informar dados de desenvolvimento” gerando informações à ferramenta, que verificará se eles são novos para serem gravados na base de esforço e a ação “Alterar dados de desenvolvimento” que gera informações de modificação de dados à ferramenta, que será gravado na base de proveniência. A ferramenta recuperará dados dos atores de base de esforço e proveniência para realizar a ação “Preparar proveniência” que organizarão as informações necessárias para executar os cálculos para “Aplicar regras fuzzy”. Nessa aplicação também precisará da ação “Definir regras fuzzy” realizada pelo ator usuário, deste modo é possível executar a ação “Aplicar da regra fuzzy” para obter os resultados de QD de esforço. Onde a ação “Extrair dados de qualidade” guarda os resultados de QD que deverão ser enviada para a próxima ação “Gravar qualidade” que grava os resultados no ator *data warehouse*. Além desses dados, será gravada também a origem destes que a ferramenta recuperará pela ação “Gravar históricos dos dados” das bases de esforço e proveniência. E finalmente o ator usuário realiza a ação “Realizar análises” que será recuperada os dados da *data warehouse*.

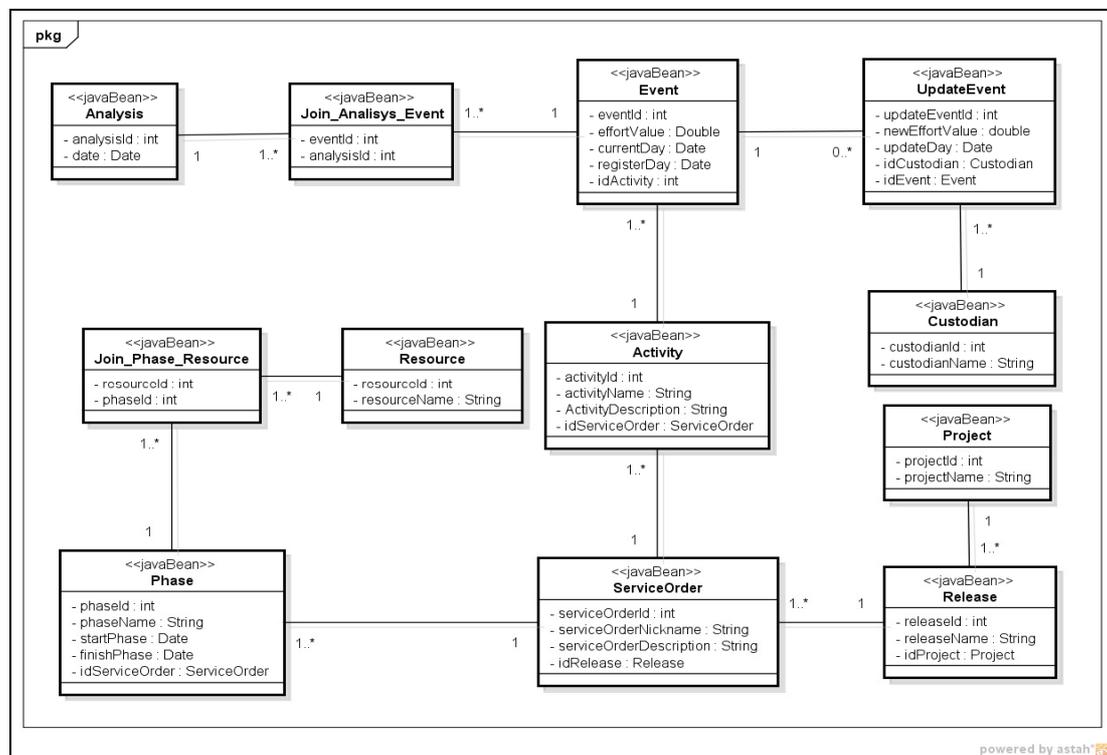


Figura 3 – Diagrama de classe do componente de proveniência.

A Figura 3 ilustra o diagrama de classe do componente de proveniência, onde foi necessária uma mudança adaptativas para facilitar a implementação da ferramenta. A base de *Effort Source* são representadas pelas classes *Project*, *Release*, *ServiceOrder*, *Phase*, *Resource*, *Activity*, *Event*, *Analysis* e *Custodian*, e a base de *Provenance* a classe *UpdateEvent*. Todas as informações dos novos eventos os dados são gravados em nas classes da base *Effort Source*. As alterações dos dados já existentes são gravadas na base de *Provenance*.

A Figura 4 ilustra a aplicação da lógica fuzzy apresentando os resultados das dimensões *accuracy*, *timeliness*, *completeness*, *consistency* e também da *quality* do componente de inferência, que já está implementada na ferramenta.

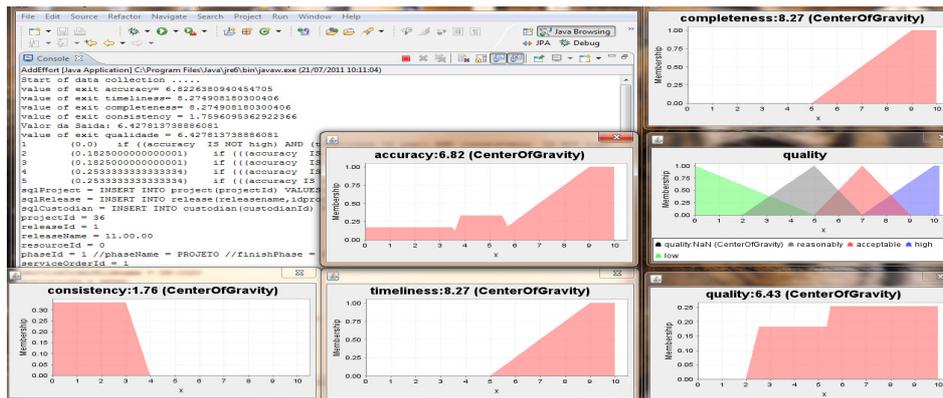


Figura 4 - Resultado apresentado no console do Eclipse na segunda fase de inferência.

4 CONSIDERAÇÕES FINAIS

A ferramenta está em fase de desenvolvimento. O modelo de dados da ferramenta referente à proveniência, o componente de inferência e o *data warehouse* foram devidamente modelados com base no trabalho de Berardi [2009], conforme apresentam as figuras 2, 3 e 4.

Atualmente o trabalho se encontra na fase de definição de uma ferramenta de BI para o componente de análise. Disponibilizando interfaces para que o gestor possa acompanhar e sugerir melhorias nas coletas dos dados no PDS.

A contribuição desta pesquisa é a construção da ferramenta FDQ-TOOL baseado no modelo Berardi [2009]. Assim, a ferramenta torna-se um facilitador para agilizar outros testes possibilitando a evolução do modelo proposto pela autora por meio da automatização da coleta de dados e da possibilidade de avaliar a qualidade de outras métricas, não limitadas só ao esforço, como realizado por Berardi [2009].

Futuramente espera-se que a FDQ-TOOL possa ser utilizada pelas organizações que mostram interesse em garantir a qualidade de seus softwares para auxiliar na gerência de seus PDSs, aumentando a visibilidade para as tomadas de decisões do processo.

REFERÊNCIAS

- BELCHIOR, A.D.. Um modelo fuzzy para avaliação da qualidade de software. 1997. 185 p. Tese (Doutorado em Ciências em Engenharia de Sistemas e Computação) - Universidade Federal do Rio de Janeiro, Rio de Janeiro. Disponível em: <http://www.boente.eti.br/fuzzy/tese-fuzzy-belchior.pdf>. Acessado em: Junho de 2011.
- BERARDI, R.C.G.. Avaliação de Qualidade de Dados de Métricas de Esforço Baseado em Data Provenance e Fuzzy Logic. 2009.134 p.. Dissertação (Mestrado em Ciências da Computação). Pontifícia Universidade Católica do Rio Grande do Sul.
- PRESSMAN, R.S.. Engenharia de Software. Tradução Jose Carlos Barbosa dos Santos, McGraw-Hill, 6 ed., p.54-83. São Paulo, 2006.

SOMMERVILLE, I.. Engenharia de Software. Tradução: Maurício de Andrade, Addison-Wesley, 8 ed., p.458-475. São Paulo, 2007.

VAZQUEZ, C.E.; SIMÕES, G.S.; ALBERT, R.M.. Análise de Pontos de Função: Medição, Estimativas e Gerenciamento de Projetos de Software. Erica, 8 ed., p.158-177. São Paulo, 2009.

Anais Eletrônico

VII EPCC – Encontro Internacional de Produção Científica Cesumar
CESUMAR – Centro Universitário de Maringá
Editora CESUMAR
Maringá – Paraná - Brasil