

SOFTWARE PARA ESTIMATIVAS ATRAVÉS DA ANÁLISE DE PONTOS POR CASOS DE USO¹

SOFTWARE FOR ESTIMATIVE THROUGH THE ANALYSIS OF POINTS BY CASE USAGE

Matheus Ricalde de Souza² e Gustavo Stangherlin Cantarelli³

RESUMO

O presente trabalho traz um estudo sobre Pontos por Casos de Uso (PCUs) e, a partir dele, o desenvolvimento de um *software*, na linguagem de programação Java, com o gerenciador de banco de dados SQL Server, que realiza a análise de PCUs calculando a complexidade dos atores e dos casos de uso automaticamente, através do descritivo dos casos de uso e os fatores ambientais e técnicos através do preenchimento realizado pelo usuário. Os relatórios obtidos através do sistema obtiveram as estimativas esperadas, permitindo concluir que a implantação do sistema é perfeitamente viável em empresas de desenvolvimento de *software*. Os testes realizados ainda permitiram o auxílio ao usuário, de maneira adequada, não somente na questão de agilidade, mas principalmente, na eliminação da etapa de treinamento de pessoas para a realização de estimativas através de PCUs.

Palavras-chave: Java, SQL Server, estimativa de tamanho de *software*.

ABSTRACT

This work presents a study on Points by Case Usage (PCUs) and, it develops a software written in Java with SQL Server database manager, which performs the PCU analysis by calculating the complexity of actors and Case Usage automatically through the description of the Case Usage and the environmental and technical factors by the user's input. The reports obtained through the system achieved the expected estimates, which allowed to conclude that the system implementation is perfectly feasible by software development companies. The tests performed also allowed the aid to the user in an appropriate way, not only in the speed issue, but mainly in eliminating the training phase of people for the realization of estimative through PCUs.

Keywords: Java, SQL Server, estimating software size.

¹ Trabalho Final de Graduação - TFG.

² Acadêmico do Curso de Sistemas de Informação - Centro Universitário Franciscano.

³ Orientador - Centro Universitário Franciscano. E-mail: gus.cant@gmail.com

INTRODUÇÃO

A qualidade do desenvolvimento do produto de *software* sempre foi um grande desafio nas organizações de seu desenvolvimento. A principal causa desse problema é a falta de um processo de desenvolvimento efetivo e claramente definido. Assim, a definição de um processo de *software* tornou-se um requisito essencial para a obtenção de produtos de *software* com qualidade e, para que seja possível a obtenção de uma melhora consistente, os processos necessitam de informações mais precisas sobre estimativas de tamanho de *software*, as quais podem ser realizadas por meio da análise de Pontos por Casos de Uso (PCUs).

Os PCUs foram criados por Gustav Karner em 1993, como uma adaptação específica dos Pontos por Função. A análise de sistemas orientados a objetos já utiliza os diagramas de casos de uso para a descrição de funcionalidades do sistema em relação ao usuário. A análise de PCUs foi criada para que seja possível estimar o tamanho do sistema nesta fase, utilizando documentos gerados nela como alternativa para o cálculo dimensional. Uma vez que os casos de uso do sistema são analisados, é possível estimar o tamanho do *software* baseado em um conjunto simples de métricas e modificadores.

Segundo Gielow (2003), o modo como a métrica de PCUs lança uma estimativa é o principal diferencial dela em relação a outras métricas de estimativas, já que o método estima o tamanho do *software* a partir do modo como os usuários o utilizarão, analisando a complexidade de ações requeridas por cada usuário e realizando uma análise de alto nível dos passos necessários para a realização de cada tarefa.

A partir dos problemas encontrados em relação à credibilidade de estimativas realizadas por equipes de desenvolvimento de *softwares*, foi elaborado este estudo sobre métricas de estimativas de PCUs, propondo um sistema no qual é possível estimar o tamanho do *software* a partir do modelo de casos de uso.

O sistema desenvolvido no presente trabalho realiza a estimativa da métrica de PCUs por meio do cálculo do PCUs não Ajustados (calculado automaticamente pelo descritivo dos casos de uso) X Complexidade dos Fatores Ambientais X Complexidade dos Fatores Técnicos, sendo possível ao usuário gerar relatórios simplificados, voltados para os clientes, e relatórios técnicos, voltados para a equipe de desenvolvimento.

Dessa forma, pode-se evidenciar, como objetivos específicos: a) desenvolver o sistema para realizar a análise da métrica de PCUs, atribuindo valores à complexidade dos atores e dos casos de uso, mediante o descritivo do modelo de caso de uso; b) atribuir valores aos Fatores Ambientais e Técnicos, através do preenchimento manual do usuário; c) realizar a geração de relatórios simplificados e complexos sobre PCUs de cada projeto; d) definir uma metodologia para desenvolvimento desta proposta; e e) validar a proposta através de estudos de caso específicos que contemplem o cenário proposto.

TRABALHOS CORRELATOS

Todos os trabalhos apresentados a seguir trazem aplicações ou estudos sobre PCUs.

COMPARING EFFORT ESTIMATES BASED ON USE CASE POINTS WITH ESTIMATITES

O trabalho desenvolvido por Anda (2002) relata os resultados de um estudo realizado para avaliar o método que estima o esforço de desenvolvimento de *software* baseado em PCUs, comparando-o com estimativas feitas por especialistas em desenvolvimento de *software*.

Os resultados demonstram que o método de PCUs pode ser usado com grande sucesso na estimativa de esforço de desenvolvimento de *software*. Além disso, a combinação de estimativas de peritos e estimativas baseadas em PCUs podem ser benéficas quando os avaliadores não têm experiência específica no domínio da aplicação dos PCUs.

EFFORT ESTIMATION TOOL BASED ON USE CASE POINTS

Seguindo na linha da aplicação de PCUs, Kusumoto (2004) desenvolveu um estudo visando ao desenvolvimento de um *software* denominado UEST, implementado em Java e Xerces2 Java Parser.

Como primeiro passo, o usuário desenha o caso de uso e salva-o no formato *XML Metadata Interchange* (XMI). Então, o *software* XMI analisador extrai automaticamente os atores e casos de uso a partir do arquivo de entrada de modelo de casos de uso salvo. Então, o *software* analisa a complexidade deles e calcula os PCU não Ajustados. Feito isso, o *software* mostra a lista de atores e casos de uso com sua complexidade. Por fim, são definidos os elementos técnicos e ambientais, e a calculadora de PCUs gera os resultados que são armazenados no banco de dados. A estimativa de esforço é calculada multiplicando o número de homens/hora pelo cálculo gerado pelos PCUs.

FERRAMENTA DE SUPORTE AO CÁLCULO DOS *USE CASE POINTS*

Gielow (2003) apresentou à Universidade Regional de Blumenau um estudo que tem como objetivo o desenvolvimento de uma ferramenta para cálculo de PCUs a partir da leitura de diagramas gerados pela ferramenta CASE Rational Rose. Para implementação do trabalho, foi utilizado o ambiente de desenvolvimento Visual C++ 6.0.

A tela principal do *software* apresenta como opções: abrir arquivo UML (*Unified Modeling Language*) do CASE Rational Rose, salvar o arquivo, abrir um arquivo salvo e imprimir um relatório referente ao cálculo. Depois de abrir um arquivo UML do CASE Rational Rose, o usuário define o tipo de ator de acordo com sua complexidade. Definido o grau de complexidade, o usuário pode

preencher os fatores técnicos e ambientais, informando o grau de influência de cada um. Por fim, o usuário insere a quantidade de homens/hora e seleciona a opção efetuar cálculo, o qual será realizado pelo *software*.

Gielow (2003) apresenta uma ferramenta que calcula a complexidade dos atores e dos casos de uso por meio do diagrama UML. Esse trabalho tem como diferencial a geração da complexidade dos atores e dos casos de uso a partir do descritivo dos casos de uso, possibilitando, assim, uma melhor usabilidade do usuário.

CONSIDERAÇÕES

Os três trabalhos apresentados visam a auxiliar o uso de PCUs, tanto por meio de comparações, como pelo desenvolvimento de um *software* para o auxílio do uso de PCUs. Assim como fez Kusumoto (2004), este trabalho utilizou a linguagem de programação Java; porém, com a diferença de que a aplicação não está voltada para a geração da complexidade dos atores e dos casos de uso a partir de um diagrama UML, e sim para a geração dos atores e dos casos mediante um descritivo de caso de uso. Gielow (2003) desenvolveu uma ferramenta que calcula os PCUs a partir dos diagramas UML. Neste trabalho, o desenvolvimento do *software* que calcula os PCUs teve seu foco no descritivo dos casos de uso e não nos diagramas, podendo, assim, favorecer uma melhor usabilidade do usuário.

REFERENCIAL TEÓRICO

Neste capítulo, serão apresentados conceitos essenciais para o entendimento deste trabalho. Serão abordados assuntos como Java, os conceitos de PCUs, o sistema gerenciador de banco de dados SQL Server e a metodologia de desenvolvimento de *software* ICONIX.

PONTOS POR CASOS DE USO (PCUs)

Karner (1993) baseou o modelo de PCUs no modelo de Pontos de Função, começando com a medição da funcionalidade do sistema, baseado no modelo de caso de uso em um número chamado Pontos por Caso de Uso Não Ajustados. Fatores técnicos envolvidos no desenvolvimento desta funcionalidade também são avaliados, semelhante aos Pontos de Função. O último passo na estimativa não é, no entanto, a partir dos Pontos de Função, sendo um novo fator chamado Fator Ambiental, proposto pelo autor. Os PCUs são produto desses três fatores.

De acordo com Freire (2003), a técnica de PCUs foi desenvolvida para que seja possível estimar o tamanho de um sistema na fase de levantamento de requisitos, utilizando-se dos próprios documentos gerados nessa fase de análise como subsídio para o cálculo.

Segundo Cohn (2005), os PCUs, em um projeto, são calculados a partir da equação entre os itens listados a seguir:

- I. O total da complexidade dos casos de uso do sistema;
- II. O total da complexidade dos atores no sistema;
- III. Fatores técnicos (como a portabilidade, o desempenho e a facilidade de manutenção) que não são descritos como casos de uso;
- IV. O ambiente em que será desenvolvido o projeto (assim como a linguagem e a motivação da equipe).

Os PCUs não Ajustados são calculados pela soma do total da complexidade dos atores com o total da complexidade dos casos de uso.

O Fator Técnico é calculado pela soma da multiplicação do peso de cada fator (no total são treze fatores) vezes o valor atribuído a ele. A Complexidade dos Fatores Técnicos (CFT) é realizada através do cálculo: $CFT = 0,6 + (0,01 \times FT)$ (equação definida por Gustav Karner).

O Fator Ambiental, por sua vez, é calculado pela soma da multiplicação do peso de cada fator (no total são oito fatores) vezes o valor atribuído a ele. A Complexidade dos Fatores Ambientais (CFA) é realizada pelo cálculo: $CFA = 1,4 + (-0,03 \times FA)$ (equação definida por Gustav Karner).

Para que seja possível calcular as estimativas, deve-se, ainda, atribuir às pessoas-hora por unidade de PCUs (o criador da métrica sugere o uso de 20 pessoas-hora) e o tamanho da equipe.

BANCO DE DADOS SQL SERVER

Devido ao desempenho e à integração com o Java, o SQL Server foi o sistema gerenciador de banco de dados (SGBD) escolhido para todos os dados que serão armazenados no banco de dados do presente trabalho.

As consultas do SQL Server são realizadas em SQL (*Structured Query Language*), uma linguagem padrão para bancos de dados relacionais em que é permitido adicionar, atualizar, recuperar e excluir informações em um banco de dados.

LINGUAGEM DE PROGRAMAÇÃO JAVA

A linguagem de programação Java é considerada de alto nível, adquirida pela Oracle em 2010 e desenvolvida pela Sun Microsystems (ORACLE, 2013).

A compilação de um programa em Java gera um *bytecode* e, para executá-lo, é necessário que um interpretador leia o *bytecode* e repasse as instruções ao processador da máquina específica, denominado JVM (*Java Virtual Machine*). Não é necessário recompilar um programa para que ele rode em

uma máquina diferente, basta que a plataforma tenha uma implementação que permita a emulação do JVM (DIAS; FONTES, 2003).

METODOLOGIA

A metodologia utilizada para o desenvolvimento de *software* foi o ICONIX. Desenvolvida pela ICONIX *software Engineering*, tal metodologia permitiu que a análise e a representação dos problemas fossem feitas de forma adequada através de seus componentes.

O ICONIX é iterativo e incremental, situado entre o RUP (*Rational Unified Process*) e o XP (*Extreme Programming*), ou seja: situa-se entre a complexidade de um e a simplicidade do outro (SILVA; VIDEIRA, 2001).

O presente trabalho consiste no desenvolvimento de um sistema que realize o cálculo da métrica de PCUs. O sistema consiste em calcular a complexidade dos atores e dos casos de uso automaticamente, a partir do descritivo dos casos de uso, este digitado pelo usuário, calculando, assim, os PCUs não Ajustados. O sistema ainda permite que o usuário atribua valores aos Fatores Ambientais e aos Fatores Técnicos manualmente, e além de permitir especificar o tamanho da equipe para a realização da métrica. Ao final do cálculo da métrica de PCUs, o usuário poderá gerar relatórios técnicos e simplificados, sendo o técnico mais complexo e voltado para que a equipe de desenvolvimento entenda todo o cálculo, e o simplificado, voltado ao cliente. O uso do sistema auxiliará os desenvolvedores de *software* a obter estimativas mais precisas sobre o desenvolvimento de produtos de *software*.

Para que os usuários possam manipular os dados no sistema, foi desenvolvida uma interface utilizando a linguagem Java e o sistema gerenciador de banco de dados SQL Server.

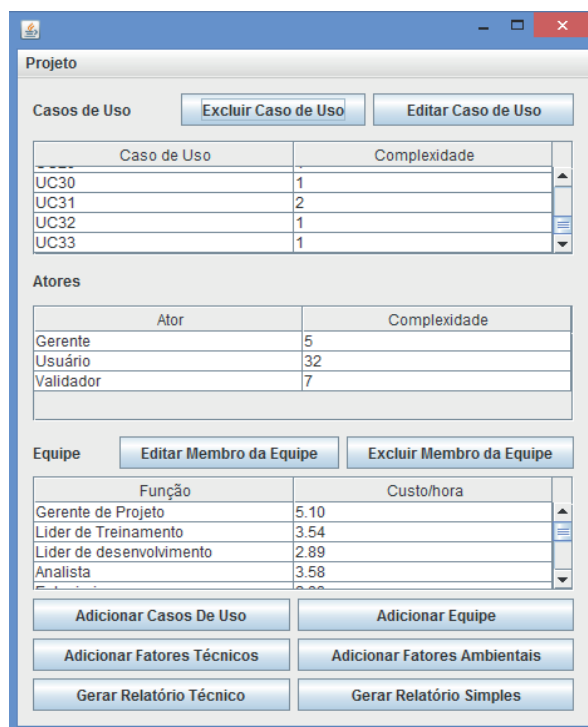
RESULTADOS OBTIDOS

Um dos principais problemas encontrados na métrica de PCUs é a falta de agilidade com que possa ser realizada a sua estimativa. O sistema desenvolvido busca uma maior agilidade na estimativa de custo de um projeto.

Diante da proposta, foram desenvolvidas interfaces para a atribuição dos fatores técnicos e dos fatores ambientais e para o preenchimento dos descritivos de casos de uso. Na figura 1, consta a interface principal do sistema com valores já atribuídos.

Conforme se observa, no exemplo da figura 1, após o usuário preencher os descritivos dos casos de uso, o sistema, automaticamente, calcula a complexidade dos casos de uso e dos atores. O usuário adicionou ao *software* os membros da equipe e seus respectivos custo/horas.

Figura 1 - Interface Principal do Sistema.



Após o usuário informar todos os dados necessários, pode-se, então, gerar relatórios técnicos e simplificados da análise de PCUs do projeto.

Nas figuras 2, 3 e 4, é demonstrado o relatório técnico gerado pelo sistema a partir do projeto criado como exemplo, o qual é voltado à equipe de desenvolvimento, sendo mais completo que o relatório simplificado, demonstrando, detalhadamente, como foi possível obter os PCUs.

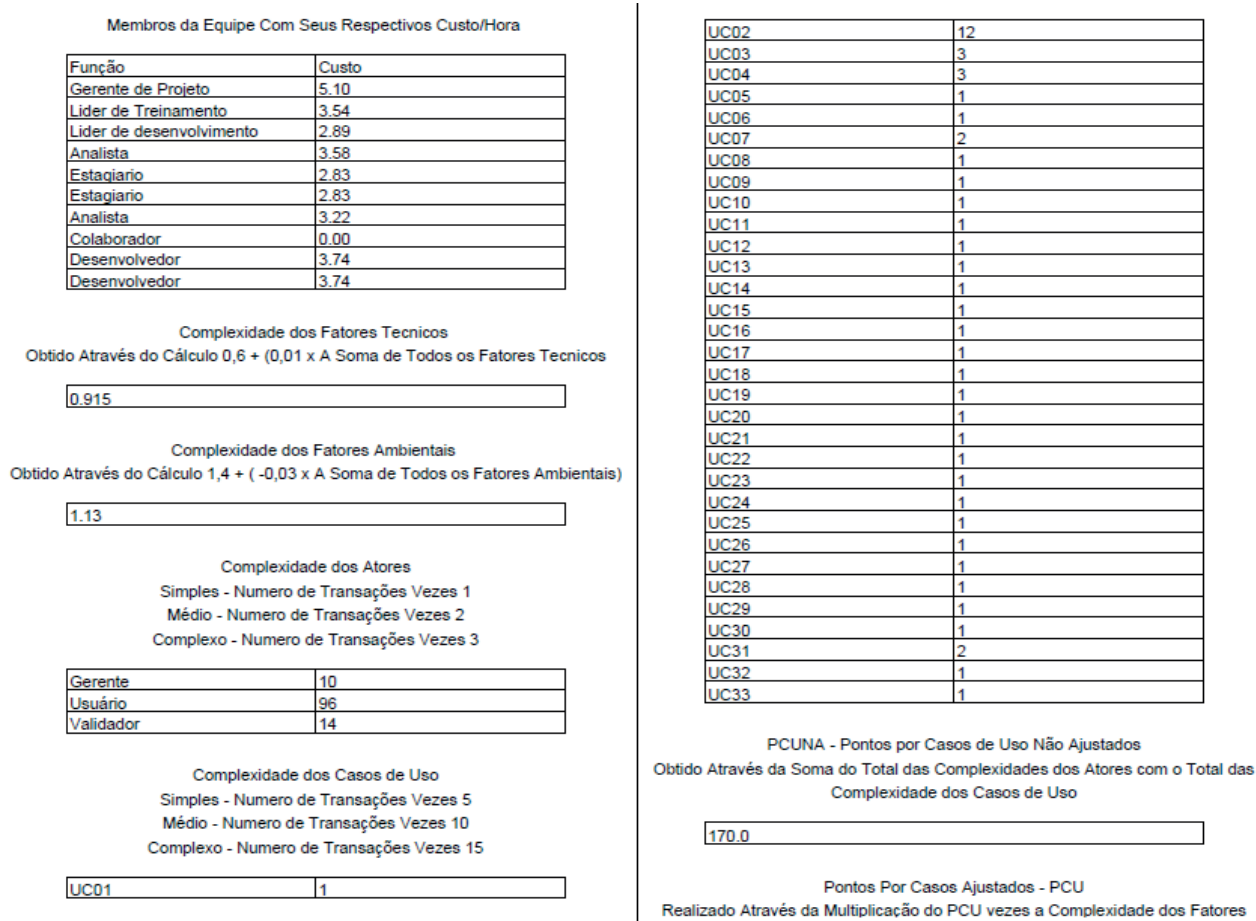
Na figura 2, são apresentados: os treze fatores técnicos, em que a complexidade de cada fator pode variar de zero até cinco, sendo cinco a complexidade mais alta, e zero a mais baixa; os oito fatores ambientais com a atribuição realizada pelo usuário, sendo que seis complexidades (de E1 até E6) podem variar de zero até cinco, sendo cinco a complexidade mais alta; e duas complexidades (E7 e E8), podendo variar de menos cinco até zero, sendo zero o menor valor da complexidade. Ainda na figura 2, são demonstrados os casos de uso e os atores com suas respectivas transações, os quais podem ser classificados em simples, médio e composto, de acordo com o número de transações.

Figura 2 - Páginas 1 e 2 do Relatório Técnico gerado pelo sistema.

Projeto Exemplo - Relatório Técnico.	
Lista dos Fatores Técnicos com Suas Respectivas Complexidades	
Descrição	Complexidade
T1 - Sistema Distribuído	2.0
T2 - Tempo de Resposta	1.0
T3 - Eficiência	3.0
T4 - Processamento Complexo	3.0
T5 - Código Reusável	3.0
T6 - Facilidade de Instalação	1.5
T7 - Facilidade de Uso	1.0
T8 - Portabilidade	4.0
T9 - Facilidade de Mudança	2.0
T10 - Concorrência	3.0
T11 - Recursos de Segurança	2.0
T12 - Acessível por Terceiros	2.0
T13 - Requer Treinamento	4.0
Lista dos Fatores Técnicos com Suas Respectivas Complexidades	
Descrição	Complexidade
E1 - Familiaridade com o Processo	3.0
E2	1.0
E3 - Experiência em Orientação a Objetos	3.0
E4 - Presença de Analista Experiente	1.0
E5 - Motivação	2.0
E6 - Requisitos Estáveis	4.0
E7 - Desenvolvedores em Meio Expediente	-2.0
E8 - Linguagem de Programação Difícil	-3.0
Lista dos Casos de Uso Com Suas Respectivas Transações	
Simple - Com Menos de 3 Transações	
Médio - Entre 4 e 7 Transações	
Complexo - Com Mais de 7 Transações	
Descrição	Numero de Transações
UC01	1
UC02	6
UC03	3
UC04	3
UC05	1
UC06	1
UC07	2
UC08	1
UC09	1
UC10	1
UC11	1
UC12	1
UC13	1
UC14	1
UC15	1
UC16	1
UC17	1
UC18	1
UC19	1
UC20	1
UC21	1
UC22	1
UC23	1
UC24	1
UC25	1
UC26	1
UC27	1
UC28	1
UC29	1
UC30	1
UC31	2
UC32	1
UC33	1
Lista dos Atores Com Suas Respectivas Transações	
Simple - Com Menos de 3 Transações	
Médio - Entre 4 e 7 Transações	
Complexo - Com Mais de 7 Transações	
Ator	Numero de Transações
Gerente	5
Usuário	32
Validador	7

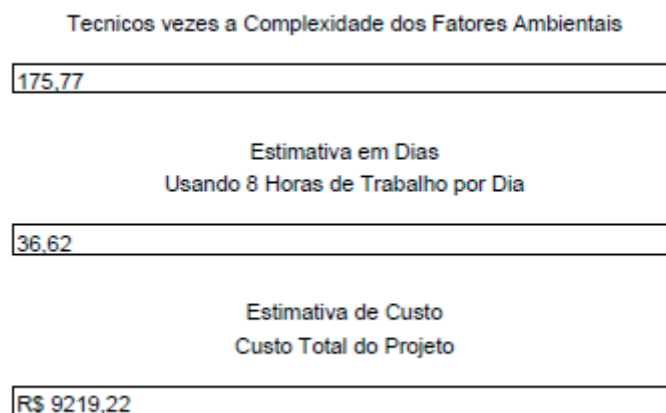
Na figura 3, são apresentados: os membros da equipe, adicionados pelo usuário, com seus respectivos custo/hora; a complexidade dos fatores técnicos e ambientais; e a complexidade dos atores e dos casos de uso, os quais também podem ser classificados como simples, médios e complexo. Ainda na figura 3, são demonstrados os PCUs não Ajustados, o cálculo para chegar a ele, os PCUs Ajustados e sua respectiva fórmula matemática.

Figura 3 - Páginas 3 e 4 do Relatório Técnico gerado pelo sistema.



As estimativas de dias, considerando oito horas de trabalho por dia, e os custos realizados pelo *software* são apresentadas na figura 4.

Figura 4 - Página 5 do Relatório Técnico gerado pelo sistema.



Baseado no estudo de Anda (2002), este trabalho foi realizado para que seja possível realizar a estimativa através de PCUs, ajudando os desenvolvedores a obterem uma previsão e um custo so-

bre o sistema a ser desenvolvido. A ferramenta desenvolvida realiza o cálculo de PCUs, assim como fizeram Kusumoto (2004) e Gielow (2003); porém, com a diferença de que a análise dos atores e dos casos de uso é realizada através dos descritivos dos casos de uso inseridos pelo usuário, e não pelo diagrama de casos de uso imortado de alguma ferramenta diferente.

CONCLUSÕES

Para a obtenção de uma melhora considerável, os processos devem conter informações mais precisas sobre estimativas de tamanho de *software*, a qual pode ser realizada por meio de PCUs. O sistema desenvolvido neste trabalho contribui para a obtenção de estimativas sobre o tamanho de *software* mediante a análise da métrica de PCUs, gerando relatórios simplificados e técnicos, em virtude da alta precisão da métrica e também pela falta de sistemas no mercado para a sua realização.

Os objetivos do trabalho foram alcançados, visto que o *software* realiza a estimativa do tamanho e do custo do projeto por meio da análise da métrica de estimativa de PCUs, precisamente. O *software* ainda pode ser utilizado em aulas de engenharia de *software* como uma ferramenta de apoio ao aprendizado.

O estudo e o desenvolvimento do sistema foram de grande valia para o conhecimento da comunicação entre linguagem de programação Java e o banco de dados SQL Server, além de permitir uma ampliação do conhecimento da métrica escolhida, bem como a realização de seu cálculo.

Os relatórios gerados pelo sistema obtiveram as estimativas esperadas, demonstrando todos os cálculos realizados para a obtenção destas e também todas as atribuições realizadas pelo usuário, como o custo/hora de cada funcionário que está participando do desenvolvimento do *software*, permitindo, assim, concluir que a implantação do sistema é perfeitamente viável no desenvolvimento de *software*. Ele ainda permite o auxílio ao usuário de maneira adequada, não somente na questão de agilidade, mas, principalmente, na eliminação da etapa de treinamento de pessoas para a realização de estimativas mediante PCUs.

Sugere-se o aperfeiçoamento e uma melhor usabilidade do sistema. Na versão atual, o usuário necessita realizar sua estimativa de custo em um computador. A sugestão é o desenvolvimento de um sistema para análise de PCUs para dispositivos móveis, buscando uma melhor portabilidade dele.

REFERÊNCIAS

ANDA, B. **Comparing Effort Estimates Based On Use Case Points With Expert Estimates. Empirical Assessment in software Engineering (EASE)**. Keele: Toronto, 2002.

COHN, Mike. **Estimating With Use Case Points**. 3. ed. Addison: Estados Unidos, 2005.

DIAS, Klessis; FONTES, Wescley. **Desenvolvimento de Aplicações para Dispositivos Móveis utilizando a Plataforma J2ME**. Belém: Brasil, 2003.

FREIRE, Herval. **Calculando estimativas**: O método de pontos de caso de uso. 2003. Disponível em: <<http://bit.ly/1HTFDs4>>. Acesso em: 15 ago. 2013.

GIELOW, Sandra Carla. **Ferramenta de Suporte ao Cálculo dos Use Case Points**. Blumenau: Universidade Regional de Blumenau, 2003.

KARNER, Gustav. **Resource Estimation for Objectory Projects. Objectory Systems SF AB Metrics**. 1993. Disponível em: <<http://www.bfpug.com.br>>. Acesso em: 15 ago. 2013.

KUSUMOTO, S. **Effort Estimation Toll Based On Use Case Points Method**. 2004. Disponível em: <<http://bit.ly/1Li7thI>>. Acesso em: 15 ago. 2013.

ORACLE. **Sun Microsystems**. 2013. Disponível em: <<http://bit.ly/1g2Jwvj>>. Acesso em: 15 ago. 2013.

SILVA, Alberto; VIDEIRA, Carlos. **UML, Metodologias e Ferramentas CASE**. Lisboa: Centro Atlântico, 2001.

