



## **ATHENA: UMA FERRAMENTA EETMS - ESTIMATIVA DE ESFORÇO DE TRABALHO EM MANUTENÇÃO DE SOFTWARE**

*Márcio Vicente de Carvalho<sup>1</sup>, Nelson Nunes Tenório Junior<sup>2</sup>*

**RESUMO:** Atualmente empresas de desenvolvimento de software investem muito dinheiro na busca de melhorias em seus produtos e serviços, já que seus clientes estão cada vez mais exigentes tanto em relação a qualidade, prazos e custos. Essas exigências são previstas em contratos que definem pesadas multas e sanções quando do não cumprimento de suas exigências. A premissa para que prazos e custos sejam cumpridos, é a estimativa de esforço de desenvolvimento, e após a entrega, o esforço para a manutenção do software, devido a isso, os pesquisadores tem o desafio de realizar pesquisas que tornem as estimativas mais precisas, e que não onerem ainda mais as equipes de software, afim de não encarecer ainda mais os custos do projeto. Dessa forma o objetivo deste projeto é o desenvolvimento de uma ferramenta com base em uma arquitetura proposta na tese de doutorado do Prof. Dr. Nelson Tenório Jr. (orientador deste trabalho<sup>0</sup>, que venha a auxiliar o processo de estimativa nas manutenções de software.

**PALAVRAS-CHAVE:** Estimativa de esforço, manutenção de software, técnicas de estimativas.

### **1 INTRODUÇÃO**

Esforço é a métrica mais popular em projetos de software (LAIRD, 2006), definido como a relação entre pessoa/hora, dia, semana, mês ou ano, sendo um dos principais desafios na área de estimativas de software. Estimar o esforço no desenvolvimento de software é fundamental, pois esta estimativa é utilizada na formação de custos e prazos, essenciais para o projeto (SOMMERVILLE, 2006). Para conseguir calcular o esforço é necessário conhecer outras métricas do projeto, como o seu tamanho (LAIRD, 2006). As mais comuns para mensurar o tamanho de um software são Linhas de Código (LOC), Pontos por Função (FP) ou Pontos por Caso de Uso (UCP), medidas essas utilizadas para a utilização de técnicas para as estimativas de esforço (GALORATH, 2006). Nos dias de hoje, as estimativas de esforço são baseadas em expertise de projetos similares anteriores. Contudo, esses dados não se aplicam quando o novo projeto apresenta especificações diferentes dos anteriores (PRESSMAN, 2009). Igualmente, empresas que não coletam e armazenam de forma eficiente os dados de projetos, possuem uma maior dificuldade na estimativas de esforço de novos projetos. E por esse motivo, as empresas que utilizam modelos de qualidade como CMMI, SPICE e ISO, têm menos dificuldade de fazê-las (SEI, 2006) (ISO, 1998) (IEEE, 1998) (PANDIAN, 2004). Quando na manutenção de software, a empresa aplica as mesmas técnicas de estimativas de esforço utilizadas no desenvolvimento. Porém, ao utilizar essas técnicas na manutenção, sua precisão tende a

<sup>1</sup> Acadêmico do Curso de Sistemas de Informação. Centro Universitário de Maringá (CESUMAR). Programa de Bolsas de Iniciação Científica do Cesumar (PROBIC). [marciocarvalho@live.com](mailto:marciocarvalho@live.com)

<sup>2</sup> Orientador e docente do Centro Universitário de Maringá (CESUMAR) . [nelson.tenorio@cesumar.br](mailto:nelson.tenorio@cesumar.br)

ficar comprometida, pois os valores das métricas na fase de manutenção diferem da fase de desenvolvimento. Por exemplo, o tamanho, acaba sendo modificado pela reutilização de componentes, diminuindo assim a manutenção a ser realizada (LUCIA, 2002). Uma das possíveis causas dessa dificuldade, é a não existência de ferramentas nem repositórios para coleta e armazenagem de estimativas de esforço, que são fundamentais para o auxílio na gerência da manutenção de projetos de software.

## **2 JUSTIFICATIVA**

Atualmente empresas de desenvolvimento de software investem muito dinheiro na busca de melhorias em seus produtos e serviços, já que seus clientes estão cada vez mais exigentes em relação a qualidade, prazos e custos. Essas exigências normalmente são previstas em contratos do tipo SLA (*Service Level Agreement*), o qual define pesadas multas e sanções quando do não cumprimento de suas exigências. A premissa para que os prazos e custos sejam cumpridos, é a estimativa de esforço de desenvolvimento, e após a entrega, o esforço para a manutenção do software, devido a isso, pesquisadores tem o desafio de buscar soluções que tornem as estimativas mais precisas, e que não onerem ainda mais as equipes de software, afim de não encarecer ainda mais os custos do projeto. Dessa forma o objetivo do projeto foi o desenvolvimento de uma ferramenta com base em uma arquitetura proposta na tese de doutorado do Prof. Dr. Nelson Tenório Junior (JUNIOR, 2009), no qual venha a auxiliar o processo de estimativa nas manutenções de software.

## **3 MATERIAL E MÉTODOS**

Inicialmente foi realizada uma pesquisa bibliográfica para ser identificadas as reais necessidades da área de estimativas de esforço para um projeto de software em uma organização, também foi estudo diferentes cenários atuais de estimativas para que fosse projetada uma ferramenta estimativa de esforço. Com base na pesquisa, foi projetada uma ferramenta que fosse executada em um ambiente SOA (WebServices), entretando, durante o desenvolvimento percebeu-se que seria mais interativo e ágil se a ferramenta fosse em um primeiro momento, desenvolvida em um ambiente Desktop. Toda a ferramenta foi concebida com softwares opensource, visando um baixo custo de implantação nas organizações, tornando assim a ferramenta mais comumente adotada. Para a validação da ferramentas foram utilizados os mesmos dados oriundos da tese do prof. Nelson Tenório Jr., pois assim possuíamos uma visão de dados já analiaados e processados para uma comparação mais eficiente.

## **A FERRAMENTA ATHENA**

Durante o desenvolvimento, foi possível observar que atualmente o processo de estimativa baseia-se em planilhas de cálculo e expertise de funcionários, fazendo com que o resultado obtido seja muitas vezes aquém do esforço real demandado, resultado esse agravado quando os dados vem de projetos e equipes diferentes, uma vez que os dados que são utilizados para realizar as estimativas muitas vezes diferem de projeto a projeto. Vale ressaltar que estimativas pouco precisas resultam em uma oneração de custos, prazos e retrabalho, e a ferramenta Ahena vem justamente para tentar sanar essa deficiência, pois ela implementa o Modelo E-10 (JUNIOR, 2009), o qual foi testado e avaliado de forma experimental, porém que veio a demonstrar uma significativa redução do esforço de realizar as estimativas de forma empírica e manualmente, gerando estimativas mais precisas. Com a ferramenta é possível migrar os dados atuais, criando um cenário personalizado de estimativas que permite uma configuração de quais dados

utilizar nos algoritmos de estimativas de forma simples, gerando assim uma estimativa mais precisa. A ferramenta também faz uso de retroalimentação, ou seja, ela própria alimenta de dados passados com o intuito de refinar as estimativas. Dessa maneira pode-se observar que, com o uso correto e prolongado da ferramenta, as estimativas tentem a alcançar o real esforço demandado.

#### 4 CONCLUSÃO

A ferramenta Athena possibilita um aprimoramento do Modelo E-10 (JUNIOR, 2009), já que a mesma torna menor o esforço de captação e análise dos dados estimados, contribuindo dessa forma com sua evolução. Como sugestão para melhorias na ferramenta, seria a integração com programas como MicroSoft Project, funcionalidades como o recálculo de estimativas com novos parâmetros e algoritmos de classificação.

#### REFERÊNCIA

- GALORATH, Daniel; EVANS, Michael. **Software Sizing, Estimation, and Risk Management**. Boston, MA, USA: Auerbach Publications, 2006.
- JÚNIOR, Nelson Nunes Tenório. **Modelo-E10: Um Modelo Para Estimativas de Esforço em Manutenção de Software**. 2009. 137f. Tese (Doutorado em Ciências da Computação) – Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, 2009.
- LAIRD, Linda; BRENNAM, Carol. **Software Measurement and Estimation - A Practical Approach**. 1st edition. New York, NY, USA: John Wiley and Sons, 2006.
- LUCIA, Silvio Stefanucci Andrea De; PENTA, Massimiliano DI; Ventuni, Gabriele. **Early effort estimation of massive maintenance processes**. In Software Maintenance, IEEE International Conference on, pages 234–237, Los Alamitos, CA, USA, 2002. IEEE Computer Society.
- PANDIAN, C. Ravindranath. **Software Metrics - A Guide to Planning, Analysis, and Application**. 1<sup>st</sup> edition. Boca Raton, FL, USA: Auerbach Publications a CRC Press Company, 2004.
- PRESSMAN, Roger. **Software Engineering: A Practitioner's Approach**. 7th Edition. Boston, USA: McGraw-Hill Science/Engineering/Math, 2009.
- SOFTWARE ENGINEERING INSTITUTE. **Cmmi for development**. Version 1.2. Pittsburgh, PE, USA: Carnegie Mellon, 2006.
- SOMMERVILLE, Ian. **Software Engineering**. 8th Edition. Boston, USA: Addison Wesley, 06/2006.
- THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC. **IEEE Computer Society. IEEE Standard for Software Quality Metrics Methodology**. NY, New York, USA: 1998.
- THE INTERNATIONAL ORGANIZATION FOR STANDARDIZATION AND THE INTERNATIONAL ELECTROTECHNICAL COMMISSION. **ISO/IEC. ISO/IEC TR 15504 Software Process Assessment**. USA, 1998.