

## PROPOSTA DE MÉTRICA PARA CÁLCULO DO ESFORÇO DE TESTES DE INTEGRAÇÃO

**Autoria:** Danilo dos Santos, Violeta Sun, Edmir Parada Vasques Prado

### Propósito Central do Trabalho

A crescente demanda por desenvolvimento de softwares, com prazos cada vez mais arrojadados, leva as organizações a buscar maior qualidade dos sistemas desenvolvidos, sua conformidade, e cumprimento de seus prazos de entrega pré-estabelecidos. Em resposta a esta dinâmica, diversas técnicas e modalidades de testes de software surgiram, consolidando a prática como uma das fases essenciais do desenvolvimento de sistemas. Este trabalho aborda o desenvolvimento de uma métrica para estimativa de esforço de execução de testes de integração. Foi realizado um estudo de caso, na área de desenvolvimento de sistemas de informação de uma organização financeira brasileira, com o objetivo de: identificar os modelos e práticas de estimativa de esforço em testes de integração descritos na literatura; identificar, em uma organização financeira, os principais fatores que influenciam a estimativa de esforço em testes de integração e; aprimorar o modelo pré-existente na organização, com base em dados históricos de projetos desta instituição e em modelos e práticas de estimativa de esforço em difundidos na literatura. A métrica proposta teve como base o histórico de 26 projetos e contou com a participação de analistas que acompanharam a execução destes projetos. Com base no esforço e tamanho destes projetos, derivou-se uma fórmula para minimizar a diferença encontrada entre a estimativa dos projetos e o esforço realizado. Ao aplicar a nova fórmula aos projetos históricos, observou-se maior aderência entre os valores estimados e realizados, podendo desta forma, a instituição reduzir os riscos de grandes divergências entre tempo estimado e realizado na execução de testes integrados.

### Marco Teórico

Após a implementação do código de um sistema, existe a necessidade de verificar a ocorrência de erros. Esta tarefa é realizada por meio da disciplina de testes de software, a qual pode ser definida como um processo para avaliação de um sistema ou componente, por meios manuais ou automáticos, a fim de verificar se ele satisfaz os requisitos especificados ou se há diferenças entre os resultados esperados e os obtidos (ANSI/IEEE Standard 729, 1983). A atividade de teste de software corresponde a um subdomínio das atividades de verificação, validação e teste (VV&T). Estas atividades integram um processo que têm seu início com as revisões dos requisitos, passam para a revisão da análise e do projeto do software, e seguida pela inspeção do código, e finalmente chegam aos testes (SOMMERVILLE, 2007). A disciplina é estruturada em fases, que correspondem aos diferentes níveis de abstração dos defeitos tratados em cada uma delas. Podem ser estabelecidos como principais fases de teste, o teste de unidade, o teste de integração, e o teste de sistema (Myers, 1979). Genericamente, um produto de software pode ser testado a partir da aplicação de duas principais técnicas: O Teste Funcional, que analisa o software a partir de uma visão limitada de sua estrutura (Beizer, 1990); e o Teste Estrutural, que consiste no conhecimento da estrutura interna da implementação para a criação dos casos de teste (Meirelles, 2006). Num contexto de projeto de software, métricas são empregadas principalmente como ferramenta para garantir sua qualidade e auxiliar sua direção, a fim de que suas fases sejam guiadas de forma mais assertiva (Pressman, 2006). Elas permitem um método de avaliação de qualidade baseado em regras bem definidas, o que possibilita entendimento imediato dos indicadores, ao invés de postergá-lo. As estimativas constituem um mapa do caminho a ser seguido para outras atividades de planejamento de projetos, e representam as bases para o sucesso do projeto de software. Desta forma, muita atenção é exigida ao se estimar tempo e custo do

desenvolvimento de software (Pressman, 2006; Sommerville, 2004). A modelagem algorítmica de custos prevê, a partir do uso de fórmulas matemáticas, os custos baseados em estimativas sobre o tamanho do projeto, o número de engenheiros de software, além de fatores adicionais de processo e produto. Os modelos algorítmicos de custo podem ser desenvolvidos a partir da análise dos custos e dos atributos dos projetos, a fim de encontrar a fórmula que mais se aproxima da realidade (Sommerville, 2004). O COCOMO é um representante desses modelos, criado a partir de composição de uma grande base histórica de projetos de software. A primeira versão do modelo foi lançada em 1981 e em 1989 ela passou por um refinamento, e teve sua última versão, o COCOMO II, publicada em 2000 (Sommerville, 2004). Esta versão mais atual reconhece abordagens diferentes para o desenvolvimento de software. Além disso, ela apoia o modelo espiral de desenvolvimento, e é totalmente compatível com o desenvolvimento de software iterativo e incremental. O COCOMO II inclui dois modelos de processo para obtenção das estimativas: Desenho Inicial (Early Design) e Pós-Arquitetura (Post-Architecture). Além disso, é composto por cinco parâmetros exponenciais denominados fatores de escala (Scale Factors), e 17 parâmetros lineares denominados multiplicadores de esforço (Effort Multipliers). No escopo deste trabalho, é abordado o modelo de processo Pós-Arquitetura (COCOMO II, 2007).

### **Método de investigação se pertinente**

Este trabalho caracteriza-se como um estudo de caso exploratório onde buscou-se aprimorar uma métrica utilizada para estimar esforço de testes integrados. Este estudo foi realizado entre setembro e dezembro de 2012. A estratégia para definição do modelo apoiou-se nas alternativas para estimativa de custo e esforço de projetos propostas por Pressman (2006). Em sua totalidade, três principais atividades foram realizadas: (1) construção do modelo, (2) coleta de dados, e (3) calibragem do modelo, nesta ordem de execução. A primeira estratégia baseou-se na análise de projetos anteriores. Foram realizadas entrevistas com funcionários da organização envolvidos no processo de contratação e monitoramento de projetos já executados, com o objetivo de mapear o processo de execução de testes de integração para investigar os fatores que impactam sua execução. Para validar sua relevância no esforço de execução dos projetos de testes de integração, os fatores identificados durante o levantamento foram dispostos em um checklist, aplicado a um grupo de funcionários que representavam uma amostra aleatória de 10% do total de sistemas da organização. Cada sistema era representado por um funcionário, que possuía experiência em projetos de testes de integração do departamento representado. Os fatores cujo percentual de relevância superou 70% foram então revisados quanto à aplicabilidade ao modelo proposto. Uma vez confirmada sua relevância, estes fatores foram então relacionados aos modificadores de esforço do COCOMO II. A segunda estratégia consistiu na coleta de dados. Uma base histórica composta por 26 projetos aleatórios de testes de integração foi considerada. Destes projetos, foram coletados dados relacionados ao modelo em desenvolvimento, tais como tamanho funcional do projeto testado e esforço de execução dos testes. A partir das variáveis obtidas, a terceira estratégia foi a definição do modelo algorítmico de custos. Realizou-se a calibração tradicional do COCOMO II, modalidade de processo pós-arquitetura. Para calibragem do modelo foi utilizado o aplicativo Calico versão 7.00 (CALICO, 2012), utilizando-se 24, dos 26 projetos da base histórica

### **Resultados e contribuições do trabalho para a área**

O presente estudo de caso buscou definir um novo modelo para estimativa do esforço para execução de projetos de testes de integração, utilizado para o processo de contratação destes projetos por uma organização financeira. Com base nos levantamentos realizados, observou-se que embora haja relação entre o tamanho funcional do sistema e o esforço contratado para

execução de sua fase de testes de integração, alguns projetos possuem características que os desviam de uma tendência. As observações sustentam a aplicação do modelo de estimativa proposto, que, apesar de basear-se no tamanho funcional, visa incorporar as particularidades de cada projeto no cálculo de seu esforço. O principal objetivo da estratégia foi gerar uma planilha, contendo um mapeamento entre os fatores referidos e os multiplicadores de esforço do COCOMOII, cuja aplicação possibilitasse identificar, segundo os parâmetros deste modelo paramétrico, o Fator de Ajuste de Esforço (EAF - Effort Adjustment Factor) dos projetos de testes de integração. Assim, a planilha desenvolvida durante a primeira estratégia foi aplicada para os 26 projetos da base histórica, possibilitando a identificação do EAF de cada um deles. Diante do esforço, tamanho, e EAF de cada projeto da base histórica, foi possível calibrar o modelo em desenvolvimento. O modelo proposto, apoiado pela premissa de que existem fatores impactantes no esforço adicionais ao tamanho funcional dos projetos de software a serem testados, representa uma evolução de um modelo pré-existente, que considera basicamente este parâmetro – o tamanho funcional do sistema a ser testado – para cálculo do esforço de execução desta fase de testes. Diferentemente do modelo original, o modelo proposto deriva o esforço diretamente do tamanho funcional do projeto de sistema a ser testado, expresso em Pontos de Função. Além disso, o modelo busca absorver as particularidades e fatores adicionais de cada projeto, os quais modificam o esforço de execução dos testes de integração, tais como o nível de experiência da equipe ou a quantidade de cenários de teste a serem executados. Adotou-se a estratégia de construção de modelo de estimativas de esforço baseado na forma geral de estimativa algorítmica de custos. Foi realizada calibração tradicional do COCOMO II, modalidade de processo pós-arquitetura. O resultado da calibragem considerou a mensuração dos projetos em Pontos de Função e obteve uma capacidade de previsão de 88% dos projetos considerados com erro de até 30%. Desta forma espera-se obter um ganho de qualidade e redução de riscos no tempo estimado para a realização de testes integrados de futuros projetos. Esta medida permitirá uma maior assertividade na condução de projetos de testes integrados, podendo trazer benefícios financeiros e de qualidade significativos uma vez que se reduzirão as grandes divergências encontradas atualmente utilizando as métricas conhecidas. Como sugestão de projetos futuros, poderá ser realizado um estudo com os futuros projetos para se verificar a melhoria na previsão dos prazos comparados com o que efetivamente for realizado.

### Referências bibliográficas

- COCOMO (2007). COCOMO II: Model Definition Manual, v. 2.1, 1995-2000. Recuperado em 15 de agosto, 2012 de [http://sunset.usc.edu/csse/research/COCOMOII/cocomo\\_main.html](http://sunset.usc.edu/csse/research/COCOMOII/cocomo_main.html). ; Maldonado, J. C., Vicenzi, A. M. R., Barbosa, E. F., Delamaro, M. E. (1998) Aspectos Teóricos e Empíricos de Teste de Cobertura de Software. In: SBC. (Org.). VI Escola de Informática da SBC - Região Sul. Curitiba: SBC ; MÉTRICA (1993). In: IEEE Standard Glossary of Computer Languages. Recuperado em 23 de outubro, 2012 de <http://standards.iee.org/findstds/standard/610.13-1993.html>. ; Pressman, R. S. (2006) Engenharia de Software. 6. ed. São Paulo: Makron Books. ; Sommerville, I. (2007) Engenharia de Software. 8 ed. São Paulo: Pearson.