

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E
TECNOLOGIA DO SUDESTE DE MINAS GERAIS - CAMPUS RIO
POMBA

WITALLY LANA BITARÃES

**Aplicação Seletiva de Métricas Heterogêneas no
Planejamento e Estimativa de Software**

RIO POMBA

2018

Witally Lana Bitarães

Aplicação Seletiva de Métricas Heterogêneas no Planejamento e Estimativa de Software

Trabalho de Conclusão de Curso apresentado ao Instituto Federal de Educação, Ciência e Tecnologia do Sudeste de Minas Gerais - Campus Rio Pomba, como requisito parcial para a conclusão do Curso de Graduação em Bacharel em Ciência da Computação.

BANCA EXAMINADORA

Prof. Dr. Sc. Wellington Moreira de Oliveira - Orientador,
Instituto Federal de Educação, Ciência e Tecnologia do
Sudeste de Minas Gerais - Campus Rio Pomba

Prof. Esp. Sc. Silder Lamas Vecchi, Instituto Federal de
Educação, Ciência e Tecnologia do Sudeste de Minas Gerais
- Campus Rio Pomba

Prof. Dr. Sc. José Rui Castro de Sousa, Instituto Federal de
Educação, Ciência e Tecnologia do Sudeste de Minas Gerais
- Campus Rio Pomba

Rio Pomba
2018

Resumo

O uso de métricas para a geração de estimativas do tempo total do projeto, ou do esforço a ser dedicado em teste no sistema ainda é pouco difundido entre as empresas. Isso se deve muitas vezes a complexidade dos itens pedidos pelas métricas e pelo pouco contato das empresas com estimativas automatizadas. Por outro lado, utilizar métricas na fase que antecede a implementação pode trazer benefícios extremamente positivos em tempo de desenvolvimento, custo do software e qualidade. Afim de auxiliar na geração das estimativas, existem na teoria ferramentas que utilizam o uso de métricas de projeto ou métricas para a geração de teste no sistema. Entretanto, tais ferramentas não possuem métricas de projeto e de teste na mesma aplicação, e não permitem o cálculo utilizando a combinação de duas ou mais estimativas. Com o intuito de auxiliar o responsável pelo projeto na escolha, utilização e integração de múltiplas métricas. Este trabalho apresenta uma nova ferramenta que permite ao desenvolvedor de software escolher a métrica ou a combinação delas mais adequada para um determinado projeto. Tal abordagem pode facilitar o planejamento do projeto, aumentar a eficiência nas estimativas geradas, entre outros benefícios.

Keywords: Engenharia de Software, Estimativa de Software, Métricas.

Abstract

The use of metrics to generate estimates of the total project time or the effort to be dedicated in testing in the system is still not widespread among companies. This is often due to the complexity of the items requested by the metrics and by the little contact of the companies with automated estimates. On the other hand, using metrics in the pre-deployment phase can bring extremely positive benefits in development time, software cost, and quality. In order to aid in the generation of estimates, there are in theory tools that use the use of design metrics or metrics for the generation of test in the system. However, such tools do not have software and test metrics in the same system, and do not allow calculation using the combination of two or more estimates. In order to assist the project manager in the choice, use and integration of multiple metrics. This work presents a new approach that allows the software developer to choose the metric or the combination of them more suitable for a given project. Such an approach can facilitate project planning, increase efficiency in the estimates generated, among other benefits.

Keywords: Software Engineering, Software Estimation, Metrics.

Lista de Figuras

2.1	Complexidade dos Atores.	9
2.2	Complexidade dos Casos de Uso.	9
2.3	Visão geral da métrica APT.	12
4.1	Arquitetura do Sistema	17
5.1	Estimativas de Projeto	24
5.2	Estimativas de Teste	25

Lista de Tabelas

4.1	Projetos utilizados no estudo	19
5.1	Relação das Estimativas Simples	22
5.2	Relação das Estimativas Múltiplas	23

Lista de Abreviaturas e Siglas

APT	: <i>Análise de Pontos de Teste;</i>
TCP	: <i>Pontos por Casos de Teste;</i>
UCP	: <i>Pontos por Caso de Uso;</i>
APF	: <i>Análise por Pontos de Função;</i>
Cocomo II	: <i>Constructive Cost Model II;</i>
IFPUG	: <i>International Function Point User Group;</i>
TPNAA	: <i>Total de Pesos NãoAjustados dos Atores;</i>
TPNACU	: <i>Total de Pesos Não Ajustados dos Casos de Uso;</i>
PCUNA	: <i>Pontos de Casos de Uso não Ajustados;</i>
FCT	: <i>Fator de Complexidade Técnica;</i>
FCA	: <i>Fator de Complexidade Ambiental;</i>
PCUA	: <i>Pontos de Casos de Uso Ajustados;</i>
CRUD	: <i>Create, Read, Update and Delete;</i>
SGBD	: <i>Sistema Gerenciador de Banco de Dados;</i>
PMBOK	: <i>Project Management Body of Knowledge;</i>
SQL	: <i>Structured Query Language;</i>
SGBD	: <i>Sistema Gerenciador de Banco de Dados;</i>
TI	: <i>Tecnologia da Informação;</i>

Sumário

1	Introdução	1
1.1	Hipótese	3
1.2	Objetivos	3
1.3	Organização do Trabalho	4
2	Referencial Teórico	5
2.1	Estimativas	5
2.2	Métricas	6
2.2.1	Análise por Pontos de Função (APF)	6
2.2.2	Modelo de Construção de Custo (COCOMO II)	7
2.2.3	Pontos por Casos de Uso (UCP)	8
2.2.4	Análise de Pontos de Teste (APT)	11
2.2.5	Pontos por Casos de Teste (TCP)	12
3	Trabalhos Relacionados	14
4	Abordagem	16
4.1	Arquitetura de Aplicação Seletiva de Métricas de Software	16
4.2	Aplicação Seletiva de Métricas de Software	18
4.3	Implementação da ferramenta de estimativa de <i>software</i>	20
5	Avaliação	21
5.1	Estimativa Simples	21

5.2	Estimativa Múltipla	23
6	Conclusão	26
	Referências	27

Capítulo 1

Introdução

Um dos maiores problemas relacionados ao gerenciamento de *software*, consiste em conseguir medições e estimativas que possam ser as mais próximas das reais ao final do projeto. Estimar e planejar o projeto na fase que antecede a implementação e implantação do *software* pode devolver resultados significativos sobre tempo, custo e qualidade do produto final (PRESSMAN; MAXIM, 2016). Para obter uma estimativa leva-se em conta que o modelo pode possuir condições para determinar precisamente o resultado planejado ou não possui dados e variáveis suficientes para obtê-los (MAGELA, 2006). Atualmente a quantidade de projetos em que são realizados estimativas em sua fase inicial cresce em números surpreendentes, porém, muitos deles perecem antes mesmo da fase de implantação do *software*. O *Standish Group's* publicou em seu relatório anual o “*The CHAOS Report*” (STANDISH, 2012), uma pesquisa feita em mais de 50 mil projetos em todo mundo. Onde 27% dos projetos atingiram seus objetivos de prazo e custo pré-estabelecidos, 56% foram entregues fora do prazo estabelecido e com orçamentos acima do previsto e 17% foram encerrados antes mesmo da fase de entrega dos projetos. Em outra pesquisa realizada pelo *Standish Group's* em 2015 (STANDISH, 2015), 29% dos projetos avaliados foram entregues dentro do prazo e custo. Desses projetos aproximadamente 52% foram entregues, porém ultrapassaram os valores de custo e prazo estabelecidos. E 19% dos projetos foram iniciados e não finalizados. Esses dados mostram que apesar do crescimento em porcentagem de projetos que chegam ao final nas condições de tempo e prazo terem aumentado. Ainda existe um longo caminho a ser percorrido no gerenciamento e planejamento de projetos. Isso se deve as dificuldades encontradas em estimar de forma correta projetos na fase que antecede a implementação.

Outro fator que pode afetar o tempo, prazo e qualidade do projeto final, é a falta de utilização de estimativas de testes a serem dedicados no projeto. Testar o *software* de

maneira correta pode auxiliar o gerente de projetos em conseguir devolver a seu cliente um produto de maior qualidade. Em toda criação ou alteração feita no software é necessária a realização de testes funcionais, ou até mesmo de desempenho, garantindo que a alteração tenha sido realizada de acordo com o esperado (MATIELI; ARAUJO, 2016). Teste de *software* é a parte do desenvolvimento em que o *software* é avaliado, para verificar se possui todas as especificações solicitadas. É avaliado também se as funcionalidades implementadas são capazes de cumprir todos os requisitos necessários no ambiente de trabalho de sua atuação (NETO, 2007). Sem o devido planejamento, a estimativa geral de entrega e custo do *software* pode ficar comprometida. Isso pode gerar um descontentamento do cliente caso o atraso na entrega se concretize. Neste sentido, o uso de métricas de estimativas do projeto ou de esforço a serem dedicados em testes, podem auxiliar os responsáveis pelo projeto a obter uma estimativa prévia sobre o projeto realizado.

Afim de auxiliar o planejamento de *software*, foram desenvolvidas ferramentas que automatizam o uso das métricas de teste e métricas que estimam as dimensões dos projetos de software (WENDERLICH, ; CARVALHO, 2012; HEIMBERG; GRAHL, 2005; BELGAMO; FABBRI, 2004; VAVASSORI, 2001). Apesar destas ferramentas auxiliarem na estimativa e planejamento do projeto, nenhuma delas possibilita a estimativa utilizando métricas de teste e métricas de *software* ao mesmo tempo. Estes trabalhos também não permitem ao usuário a possibilidade de utilizar a junção de duas métricas em uma mesma estimativa. Esta limitação pode ser um problema quando o responsável pelo projeto, deseja trabalhar com métricas distintas das oferecidas por tais ferramentas. Neste sentido, este trabalho apresenta uma nova abordagem que oferece ao responsável pelo projeto uma ferramenta de estimativa semiautomática, baseada em diferentes métricas. Tal abordagem permite que o gerente de projetos escolha a métrica que melhor se adéqua as características do projeto desenvolvido. Permitindo valores mais precisos e podendo realizar comparações dos resultados entre as diferentes métricas disponíveis. Além disso o gerente de projeto tem a possibilidade de realizar uma média entre as métricas disponíveis, podendo obter estimativas mais exatas sobre o projeto escolhido.

A abordagem apresentada neste trabalho se baseou em uma pesquisa sobre as métricas mais presentes em estudos e que possuíam uma maior aceitação nos projetos. Assim, este estudo inclui duas métricas de teste de *software* (CARVALHO, 2012; PATEL, 2001): (i) Análise de Pontos de Teste (APT) que é baseada na análise por pontos de função e (ii) Estimativa em Pontos por Casos de Teste (TCP) que estima os projetos com base nos casos de teste do sistema. A ferramenta possui também três métricas de estimativas de projeto (LIMA, ; BELGAMO; FABBRI, 2004; VAZQUEZ, 2003; AGUIAR, 2017; MANUAL,

1995): (i) Pontos por Casos de Uso (UCP) que estima o tamanho do sistema com base principalmente nas especificações de casos de uso; (ii) Análise por Pontos de Função (APF) métrica baseada nos pontos de função do sistema, muito utilizada por empresas em estimativas de *software* e (iii) *Constructive Cost Model II* (CoCoMo II) métrica de alta precisão baseada em linhas de código ou pontos de função. Assim o sistema é capaz de devolver resultados de estimativas precisas sobre tempo, prazo e qualidade de cada projeto, tornando o planejamento prévio mais seguro e com menor pretensão a erros.

1.1 Hipótese

Todo desenvolvimento ou criação de *software* exige um planejamento prévio sobre os processos que serão implementados no sistema. Esse planejamento se torna ainda mais importante, na medida que o tamanho e complexidade do *software* aumentam. Isso ocorre pois, é necessário a realização de um estimativa para saber o tempo total do projeto, custo do *software* e mão de obra necessária no desenvolvimento, para se ter um controle sobre o projeto desenvolvido.

Desta forma, é possível planejar o projeto desenvolvido utilizando métricas de teste ou de *software*. Isso pode auxiliar os desenvolvedores a obter um controle maior sobre o projeto desenvolvido, com um planejamento mais confiável e com menor pretensão a erros.

Assim sendo, a hipótese definida neste trabalho aborda a utilização de métricas de teste e métricas de *software*, afim de auxiliar os desenvolvedores no planejamento do sistema. A utilização das métricas pode resultar em estimativas mais próximas das reais, principalmente quando são utilizadas a junção de duas métricas. A partir do levantamento realizado neste trabalho, nenhuma abordagem ou ferramenta disponível na literatura fornece tais características.

1.2 Objetivos

Este trabalho tem como objetivo principal criar uma abordagem inovadora que estima o esforço gasto no desenvolvimento de um projeto, aplicando múltiplas métricas para obter um resultado mais preciso, e auxiliar no planejamento prévio do projeto. O trabalho também abordou a utilização de estimativas utilizando a junção de mais de uma métrica, afim de retornar resultados ainda mais satisfatórios. Em complemento, os objetivos

secundários são:

1. Apresentar uma breve descrição de todo o referencial teórico que envolve as métricas de teste e de *software* utilizadas para construir a proposta monográfica.
2. Difundir os benefícios do uso das métricas, afim de aumentar o número de empresas que medem o esforço de seus projetos previamente, utilizando métricas para testes e métricas de software.
3. Diminuir o número de falhas e evitar gastos desnecessários com possíveis correções ou prorrogação do prazo de entrega do projeto.
4. Possibilitar a escolha de estimativas utilizando uma métrica ou a junção de duas métricas distintas, afim de obter possivelmente uma melhor estimativa.
5. Comparar a ferramenta proposta neste trabalho com outras abordagens.
6. Verificar a viabilidade de utilização da abordagem definida neste trabalho.

1.3 Organização do Trabalho

O artigo está dividido da seguinte forma. A Seção 2 apresenta as métricas utilizadas neste trabalho, exibindo seus conceitos e explicando a forma como cada uma trabalha suas informações. Na Seção 3 é feita uma descrição dos trabalhos relacionados na área. Visando um melhor entendimento sobre o funcionamento da ferramenta, na Seção 4 é mostrada a arquitetura do sistema, com a descrição das funções presentes na abordagem. Nesta mesma seção são apresentados exemplos de como os dados são inseridos e como o sistema trata as informações, expondo o funcionamento interno e externo da arquitetura. Na Seção 5, é feito um estudo de caso sobre os testes feitos na ferramenta e os resultados por elas obtidos. A Seção 6 conclui o trabalho, expondo qual métrica implementada no sistema demonstrou um percentual mais alto de acerto, comparada ao tempo real do projeto. Por fim, listamos algumas propostas de trabalhos futuros, visando o aperfeiçoamento e crescimento do sistema.

Capítulo 2

Referencial Teórico

Neste capítulo são apresentados termos e conceitos utilizados no estudo. Além disto, também são apresentados as métricas presentes nesta abordagem. Dentre as métricas estão as de estimativas de *software* Cocomo II, APF e UCP, as métricas de teste APT e TCP. É exposto as particularidades de cada métrica e sua descrição.

2.1 Estimativas

Segundo Sommerville (2003), o uso de estimativas torna-se necessário quando o objetivo é estabelecer um orçamento para o projeto ou para definir um preço do software para o cliente. Atualmente, grande parte das empresas geram as estimativas baseadas em experiências de projetos passados. Isso é comum pois, caso se tenha um projeto passado similar ao atual, estimar esforço, prazo e custo, torna-se uma tarefa mais fácil, uma vez que estes são muito próximos (PRESSMAN, 2005). Porém, devido a particularidades que podem estar presentes em cada projeto, determinar com total exatidão os valores de prazo e custo final, acaba se tornando impossível antes da finalização do projeto. Desta forma, tudo o que pode ser feito, até os cálculos mais rigorosos, são consideradas estimativas. Independente de quais cálculos ou métodos forem empregados, os resultados podem variar em precisão e acurácia comparados com os reais (VAZQUEZ, 2003).

Segundo Sommerville (2003), as técnicas de estimativa possuem seus próprios pontos positivos e negativos. Se tratando de grandes projetos, pode ser necessário a utilização de diversos métodos de estimativa de custos para comparar seus resultados. Se em seu resultado é mostrado valores extremamente diferentes, isso sugere que não se tem informações suficientes para fazer a estimativa. É necessário obter mais informações e repetir o processo de estimativa, até que as estimativas sejam semelhantes.

Para Pressman (1995), ferramentas automatizadas de estimativas que possuem uma ou mais técnicas, quando combinadas com uma interface gráfica para o usuário, fornecem uma opção atraente para a estimativa.

2.2 Métricas

As métricas de *software* têm como finalidade medir e avaliar atributos internos do projeto, permitindo assim ao responsável pelo projeto recolher informações sobre tempo, custo e esforço antes mesmo do desenvolvimento do produto estar concluído (FERNANDES, 1995). Para Pressman e Maxim (2016) o *software* pode ser considerado o elemento virtualmente mais caro de todos os sistemas baseados em computador.

Assim, em sistemas complexos um pequeno erro de estimativa pode ocasionar uma grande diferença entre o lucro e o prejuízo, podendo tornar o excesso de custo com o sistema algo desastroso (TRODO, 2009). Outro fator que pode causar um mal desenvolvimento do projeto, é a falta de planejamento de testes no sistema a ser gerado. Corrigir um erro na fase final do projeto pode aumentar drasticamente o custo do *software*.

Para considerar a utilização das métricas útil no ambiente real, uma métrica de estimativa precisa ser simples e ao mesmo tempo calculável, persuasiva, consistente e objetiva. Não pode depender da linguagem de programação em que o projeto será feito e fornecer informações concretas ao gerente de projeto (PRESSMAN, 2005).

2.2.1 Análise por Pontos de Função (APF)

A Análise por Pontos de Função é um método de medição do desenvolvimento do *software* criada em meados dos anos 70. Esta métrica é considerada uma das mais conhecidas no mundo, principalmente depois que começou a ser gerenciada pelo Grupo Internacional de Usuários de Pontos por Função (*International Function Point User Group* - IFPUG) (TAVARES, 2002). A métrica tem como objetivo estimar o tempo de desenvolvimento a ser gasto em determinado projeto levando em conta os pontos de função presentes no *software* sob o ponto de vista dos usuários. A capacidade de conseguir mensurar o tamanho de um *software* mesmo sem ter uma única linha de código, permite que a estimativa seja feita utilizando a visão do usuário. Ou seja, é possível a geração de estimativas utilizando apenas a documentação do sistema que será desenvolvido (TAVARES, 2002). Para a geração da estimativa, o manual da métrica APF que é disponibilizado pelo IFPUG classifica os seguintes tipos de elementos funcionais:

1. Entrada Externa – EI (*External Input*) – transações lógicas onde dados entram na aplicação e mantém dados internos.
2. Saída Externa – EO (*External Output*) – transações lógicas onde dados saem da aplicação para fornecer informações para usuários da aplicação.
3. Consulta Externa – EQ (*External Query*) – transações lógicas onde uma entrada solicita uma resposta da aplicação.
4. Arquivos Lógicos Internos – ILF (*Internal Logical File*) – grupo lógico de dados mantido pela aplicação.
5. Arquivos de Interface Externa – EIF (*External Logical File*) – grupo lógico de dados referenciado pela aplicação, mas mantido por outra aplicação.

Por ser uma métrica já antiga na teoria, a estimativa APF foi perdendo espaço por não ser tão efetiva na medição de projetos mais recentes no mercado, por exemplo, os projetos que utilizam o paradigma de Orientação a Objetos (OO). Mais informação sobre a métrica APF podem ser encontrados no site do IFPUG¹, responsável pela manutenção da mesma.

2.2.2 Modelo de Construção de Custo (COCOMO II)

O Modelo de Construção de Custo II (*Constructive Cost Model*) mais conhecido como Cocomo II, teve sua criação baseada na métrica conhecida como Cocomo 81, que foi publicada originalmente em 1981 por Barry Boehm (MAGELA, 2006). O método Cocomo II consiste em uma abordagem de desenvolvimento baseada em modelos de prototipação e o desenvolvimento pela agregação de componentes. As estimativas geradas pela métrica são condizentes com as tecnologias modernas existentes no mercado, mensurando esforços, prazos e custos com um alto índice de acerto (SOMMERVILLE, 2003).

A métrica Cocomo II é geralmente estimada utilizando dois modelos, o *Post-Architecture* que é um modelo mais detalhado usado quando o projeto possui toda sua documentação completa. E o modelo *Early Design* que é um modelo de alto nível usado para explorar alternativas arquitetônicas ou estratégias de desenvolvimento incremental (BOEHM, 2000). Além disso, é possível realizar estimativas com o Cocomo II utilizando a contagem de linhas de código (*LOC*) ou a contagem por pontos de função. Na abordagem proposta

¹<http://www.ifpug.org>

neste trabalho foram utilizados para a implementação da métrica Cocomo II o modelo *Post-Architecture* e a contagem por pontos de função. Essa escolha levou em conta que o objetivo do trabalho é conseguir a estimativa mais exata possível na fase que antecede o desenvolvimento do *software*.

O modelo COCOMO é o um dos mais concretos e documentados em termos de estimativas de esforço. Esta métrica é baseada em uma análise feita por Boehm e fornece fórmulas que possibilitam determinar o cronograma do tempo de desenvolvimento, esforço geral envolvido no mesmo, assim como interrupção do esforço por fase/atividade e ainda o esforço de manutenção (PETERS; PEDRYCZ, 2001). Informações mais detalhadas podem ser encontradas no manual oficial da métrica utilizado como base para este estudo (BOEHM, 1995).

2.2.3 Pontos por Casos de Uso (UCP)

Diferente da métrica APF a estimativa UCP (*User Case Points*) é uma métrica mais recente disponível na teoria, conseguindo assim um bom desempenho em projetos orientados a objetos. A métrica UCP realiza suas estimativas baseada principalmente nos casos de uso do sistema (BELGAMO; FABBRI, 2004).

A estimativa UCP segue como base a utilização do diagrama de casos de uso para a geração das estimativas, são seguidos os seguintes passos:

1. Calcular a Complexidade dos Atores do Software – Os atores presentes no diagrama de casos de uso devem ser classificados de acordo com o seu nível de complexidade, como mostrado na Figura 2.1. Definidos as complexidades de cada ator do sistema, esses valores são inseridos em uma fórmula, onde é multiplicado o ator pelo seu respectivo peso de complexidade. O resultado será o TPNAA (Total de Pesos Não Ajustados dos Atores).

Complexidade do Ator	Descrição	Peso
Simple	Muito poucas entidades de Banco de Dados envolvidas e sem regras de negócio complexas	1
Médio	Poucas entidades de Banco de Dados envolvidas e com algumas regras de negócios complexas	2
Complexo	Regras de negócios complexas e muitas entidades de Banco de Dados presentes	3

Figura 2.1: Complexidade dos Atores.

Fonte: Adaptado de Karner (1993, p. 2)

$$TPNAA = 1 \cdot NumAtoresSimple + \cdot NumAtoresMdios \quad (2.1a)$$

$$+ \cdot NumAtoresComplexo \quad (2.1b)$$

Fonte: Adaptado de Karner (1993)

2. Calcular a Complexidade dos Casos de Uso do Software – Nesta etapa, é feita uma contagem dos casos de uso dos sistemas e é atribuído o grau de complexidade com base no número de classes e transações. O grau de complexidade dos casos de uso pode ser visto na Figura 2.2. Definidos a complexidade de cada Caso de Uso, os mesmo são inseridos na fórmula e multiplicados pelo seu determinado peso. Assim é possível obter o valor do TPNACU (Total de Pesos Não Ajustados dos Casos de Uso).

Típos de Casos de Uso	Descrição	Peso
Simple	Considerar até 3 transações com menos de 5 classes de análise	5
Médio	Considerar 4 a 7 transações com 5 a 10 classes de análise	10
Complexo	Considerar 7 transações com pelo menos 10 classes de análise	15

Figura 2.2: Complexidade dos Casos de Uso.

Fonte: Adaptado de Karner (1993, p. 3)

$$TPNACU = 5 \cdot NumCasoUsoSimples + 10 \cdot NumCasoUsoMedios \quad (2.2a)$$

$$+ 15 \cdot NumCasoUsoComplexo \quad (2.2b)$$

Fonte: Adaptado de Karner (1993)

3. Calcular os Pontos de Caso de Uso Não Ajustados – Esse item realiza a soma dos valores encontrados nos itens anteriores, como pode ser visto na fórmula abaixo. Assim, obtemos o valor do PCUNA (Pontos de Casos de Uso não Ajustados).

$$PCUNA = TPNA + TPNACU \quad (2.3)$$

Fonte: Adaptado de Karner (1993)

4. Determinar o Fator de Complexidade Técnica – Nesta etapa, é listado para o usuário diversos fatores que podem influenciar na complexidade técnica do sistema, dentre eles temos: Usabilidade, Portabilidade, Sistemas Distribuídos entre outros. Cada fator possui um peso correspondente, assim, realizando o somatório dos itens, e multiplicando cada um pelos seus pesos e por alguns valores constantes, é possível obter o FCT (Fator de Complexidade Técnica).

$$FCT = 0,6 + (0,01 \cdot SomatriodosTi \cdot Peso) \quad (2.4)$$

Fonte: Adaptado de Karner (1993)

5. Determinar o fator de complexidade ambiental: – Nesta parte são listados fatores ambientais, que possuem interferência direta no resultado das estimativas, como por exemplo: experiência na linguagem de programação, motivação, requisitos estáveis etc. Cada fator possui um peso respectivo, como pode ser visto na fórmula abaixo, é feito um somatório dos itens e multiplicados pelo peso de cada um, esse valor será somado e multiplicado por valores constantes pré definidos. O resultado será o FCA (Fator de Complexidade Ambiental)

$$FCA = 1,4 + (-0,03 \cdot SomatriodosFi \cdot Peso) \quad (2.5)$$

Fonte: Adaptado de Karner (1993)

6. Calcular os Pontos de Casos de Uso ajustados: Nesta etapa é multiplicada cada item anterior, assim podemos obter o PCUA (Pontos de Casos de Uso Ajustados).

$$PCUA = PCUNA \cdot FCT \cdot FCA \quad (2.6)$$

Fonte: Adaptado de Karner (1993)

7. Calcular a estimativa de horas de programação: Neste último item são realizados os cálculos finais para a geração das estimativas. Karner (KARNER, 1993) o criador da estimativa, sugere utilizar 20 pessoa-hora. Assim, multiplicando o PCUA \cdot *quantidade de pessoa – hora*, obtemos *total de horas do sistema*.

2.2.4 Análise de Pontos de Teste (APT)

A métrica Análise de Pontos de Teste tem como objetivo calcular o esforço a ser gasto em horas de teste no projeto. Para isso, é necessário ter acesso a descrições detalhadas do processo e um modelo de dados lógicos, de preferência incluindo uma tabela CRUD. A métrica APT utiliza em seu cálculo de estimativa a métrica APF ou derivações da mesma, o que deixa ainda mais sugestivo a semelhança entre o nome das duas, apesar de terem finalidades distintas (VEENENDAAL; DEKKERS, 1999).

Para obtenção da estimativa algumas etapas devem ser informadas e calculadas. Abaixo na Figura 2.3 podemos ver de forma geral, os passos para atingir a estimativa em horas.

O número de pontos de teste necessários para os testes dinâmicos é calculado para cada função com base no número de pontos de função presentes no sistema. Serão informados os fatores de funções dependentes (complexidade, interfaces, uniformidade, importância do usuário e intensidade de uso), a qualidade dos requisitos relacionados e a estratégia de teste para as características de qualidade dinâmica. A soma destes pontos de teste atribuídos a cada função é o número de pontos de teste dinâmico.

O número de pontos de teste estático para medir as características de qualidade, é calculado com base no número total de pontos de função para o sistema. São utilizados também a qualidade dos requisitos e a estratégia de teste para as características de qualidade estáticas. Isto resulta no número de pontos de teste estático.

Para se obter o número de horas de teste primárias, é necessário multiplicar o número total de pontos de teste pelos fatores de ambiente de teste e produtividade. As horas de teste primárias representam o volume de trabalho envolvido nas atividades de teste primárias, como por exemplo, o tempo necessário para as fases de Preparação, Especificação, Execução e Conclusão.

O total de horas de teste é calculado adicionando alguns fatores sobre o planejamento

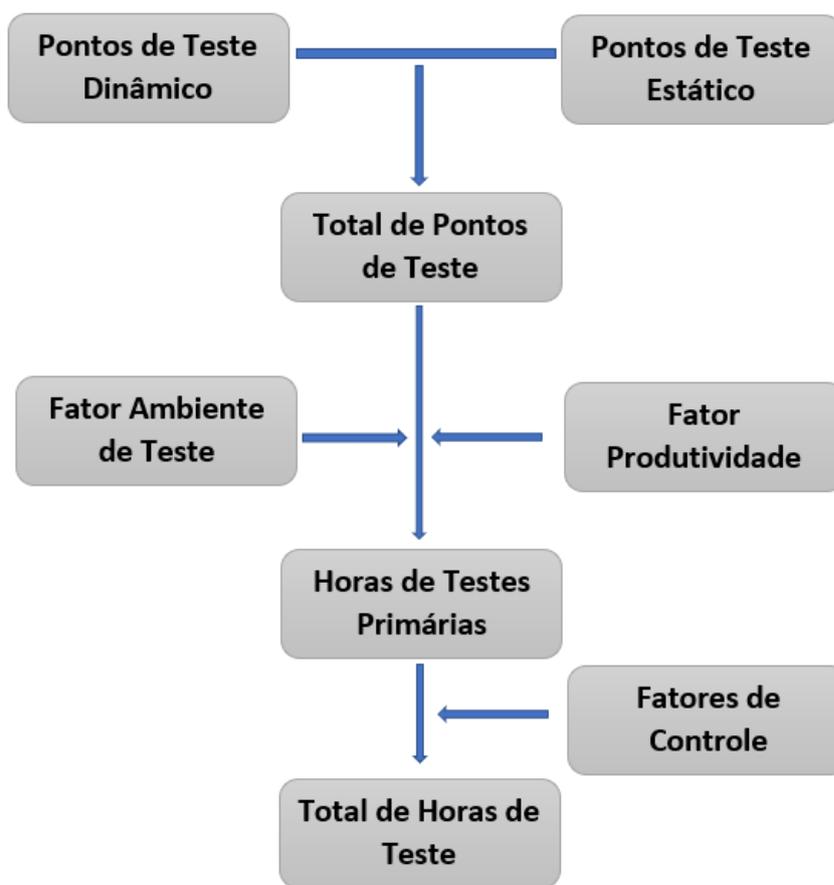


Figura 2.3: Visão geral da métrica APT.

Fonte: Adaptado de Veenendaal e Dekkers (1999, pp. 47-59)

e controle ao total de horas primárias. O tamanho deste subsídio, que representa o volume de trabalho envolvido nas atividades de gerenciamento, dependem do tamanho do time de teste e da disponibilidade de ferramentas de gerenciamento.

2.2.5 Pontos por Casos de Teste (TCP)

A métrica de teste TCP realiza uma estimativa do tempo a ser dedicado em testes nos projetos. Sua abordagem tem como principais objetivos determinar o nível de complexidade de cada ciclo de teste, gerando uma representação do esforço envolvido no teste (PATEL, 2001). A estimativa gerada pela métrica TCP é realizada em atividades de testes diferentes. Geralmente projetos são a combinação de quatro modelos diferentes: (i) Geração de Caso de Teste; (ii) Geração automatizada de scripts; (iii) Execução de Teste Manual; (iv) Execução Automática de Testes.

A análise TCP consiste em um processo de sete etapas, definidas em: (i) Identificar os Casos de Uso; (ii) Identificar Casos de Teste; (iii) Determinar TCP para Geração de

Caso de Teste; (iv) Determinar TCP para Automação; (v) Determinar TCP para execução manual; (vi) Determinar TCP para Execução Automatizada; (vii) Determinar TCP total. Ao final das etapas, os valores obtidos são inseridos em fórmulas que resultarão no TCP final, que será o total de horas a serem dedicadas no projeto.

Capítulo 3

Trabalhos Relacionados

As métricas de estimativas ainda são pouco difundidas entre as empresas. Isso ocorre muitas vezes pela complexidade dos itens que são pedidos, e por não existirem muitos sistemas que permitem ao desenvolvedor gerar estimativas de forma automatizada. A seguir são descritas algumas abordagens que utilizam métricas para a geração de estimativas.

No trabalho de WENDERLICH (2008) foi proposta a criação de uma ferramenta de estimativas utilizando as métricas UCP, APF e Cocomo II. Essa ferramenta teve como objetivo de auxiliar o responsável pela gestão do software nas estimativas a serem geradas, podendo realizar estimativas utilizando mais de uma métrica de projeto.

O trabalho proposto por JÚNIOR (2003), propôs a criação de uma ferramenta para estimativas de custos de projetos, baseada no livro *Project Management Body of Knowledge* (PMBOK). Em seu estudo foi aplicado a métrica APF afim de estimar os projetos e auxiliar o desenvolvedor seguindo as regras propostas pelo PMBOK.

A abordagem apresentada por Kusumoto, Matukawa, Inoue, Hanabusa e Maegawa (2004) visa a criação de uma ferramenta para geração de estimativas baseada na métrica UCP (*User Case Points*). Levando em consideração que as estimativas realizadas utilizando esta métrica são feitas geralmente de forma manual. O autor propôs uma ferramenta que realizasse a estimativa do projeto de forma automatizada, na fase que antecede o desenvolvimento do *software*.

O estudo realizado por Carvalho, Essado e Ambrosio (2012) apresentou resultados da aplicação de uma métrica para estimativa de teste de software, conhecida como Análise de Pontos de Teste (APT) em um sistema de processamento de transações. A métrica é utilizada para quantificar o esforço da equipe de analistas responsáveis por planejar e executar os testes. A Análise de Pontos de Teste é utilizada em um sistema transacional

que envolve ambientes web, processos de dados em lote (*batch*) e suas interfaces.

Em seu trabalho, os autores Matieli e Araujo (2016) fazem uma abordagem geral sobre diversas métricas de software presentes na teoria. Os autores realizam uma análise de algumas métricas existentes mostrando os pontos fortes e fracos de cada uma. Também discutem o fato de que apesar de tantas métricas existentes na teoria, ainda se tem um número baixíssimo de empresas que as utilizam em seus projetos, e que nenhuma é considerada dominante ou mais recomendável. O artigo não realiza a aplicação de algumas das métricas em projetos, o que poderia dar uma melhor noção da aplicabilidade em situações específicas.

No trabalho de Belgamo e Fabbri (2004) é descrito algumas especificações sobre a métrica Pontos por Casos de Uso (UCP). O objetivo do trabalho é mostrar que um dos problemas da pouca utilização da métrica UCP pelas empresas pode ser levado pelo baixo contato das empresas com as métricas. Outro fator é a falta de padronização dos casos de uso de um sistema, levando em conta que este é um requisito essencial para a geração da estimativa utilizando a UCP. Assim a proposta é, padronizar os casos de uso, utilizando também o documento de requisitos para fazer tal padronização, obtendo assim melhores estimativas sobre os projetos utilizando a métrica UCP.

A abordagem proposta por Trodo (2009) realizou um levantamento sobre algumas métricas de teste existentes na literatura, e sobre a importância de se realizar os testes de software. Foram implementadas algumas experiências em empresas do Rio Grande do Sul, onde se avaliou o uso de métricas para estimativa de teste. Segundo a autora ainda se tem um déficit muito grande de empresas da área de TI que utilizam métricas para o planejamento do projeto. Assim, a autora deixa como opções de trabalhos futuros realizar um acompanhamento maior a empresas e seus projetos, afim de descobrir como são tratados os resultados do desenvolvimento dos projetos até a fase final.

Diferentemente dos trabalhos apresentados que têm como objetivo a utilização de apenas uma métrica em seus projetos, ou a utilização de apenas métricas de estimativas de projeto. A abordagem proposta nesse trabalho traz a criação de uma ferramenta que trabalhe com múltiplas métricas de projeto como APF, UCP e Cocomo II. E também que possua métricas para a geração de estimativas de esforço de testes a serem dedicadas nos projetos como APT e TCP. Assim, existe a possibilidade de escolha do responsável pelo projeto sobre qual métrica utilizar. A ferramenta também permite gerar estimativas sobre horas de testes, além de poder gerar a média das estimativas geradas pelas métricas, obtendo assim uma maior exatidão nos resultados.

Capítulo 4

Abordagem

Como foi descrito, existem algumas ferramentas capazes de estimar projetos na fase que antecede o desenvolvimento do *software*. Porém, o uso de métricas como forma de geração de esforço, tanto do tempo total do projeto, quanto do tempo a ser dedicado a testes ainda é pequeno. O sistema descrito nesta abordagem permite a geração de estimativas utilizando múltiplas métricas de projeto e também métricas específicas para o cálculo de horas de teste a serem dedicadas no projeto. Além disso, permite a realização do cruzamento das estimativas geradas, garantindo que o resultado seja uma estimativa mais exata possível.

4.1 Arquitetura de Aplicação Seletiva de Métricas de Software

A arquitetura proposta no sistema tem como objetivo a geração de estimativas, afim de devolver para o usuário o tempo de esforço a ser dedicado em seu projeto. A arquitetura é composta por componentes, que geram as estimativas referentes a cada métrica presente no sistema, como pode ser visto na Figura 4.1.

O componente *Métricas de Teste*, é responsável por conter as informações das métricas de teste presentes no sistema. Desta forma, o usuário pode consultar e conhecer as informações presentes em cada uma delas. O componente *Métricas de Software* traz as métricas utilizadas nas estimativas de tempo totais dos projetos. Isso permite, que o usuário, possa escolher entre métricas de teste e *software*, permitindo selecionar a que mais se adéqua ao seu sistema.

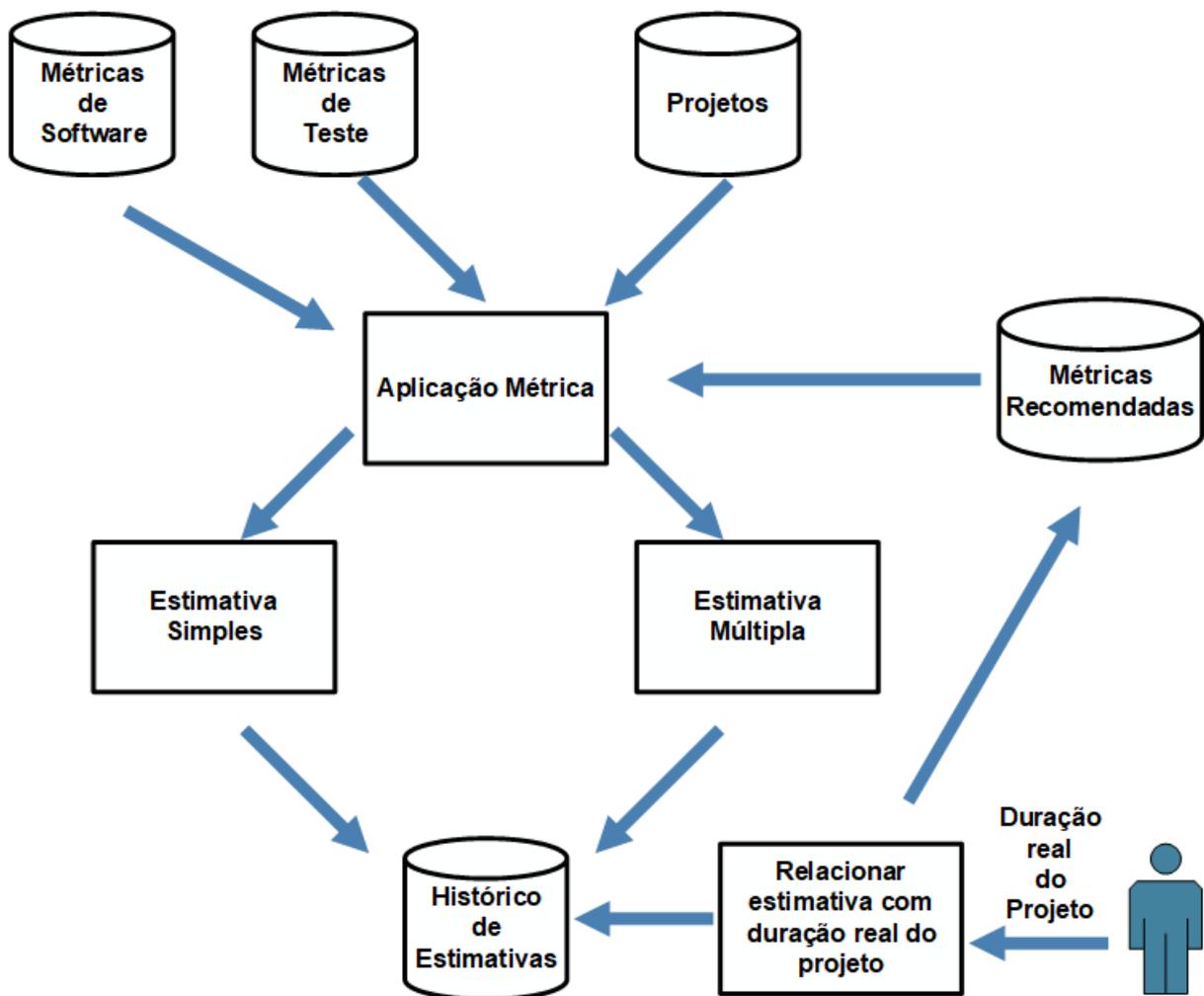


Figura 4.1: Arquitetura do Sistema

Para a geração das estimativas, é necessário que o responsável pelo sistema faça o cadastro do projeto que deseja estimar. As informações de cadastro de cada projeto, como nome, tipo de projeto (Sistema Financeiro, ERP, Comercial, etc.), nome do responsável pelo sistema, entre outras, ficam armazenadas no componente *Projetos*.

O componente *Aplicação Métrica* contém as informações presentes no componente *Métricas de Teste*, *Métricas de Software* e no componente de *Projetos*. Assim, é possível gerar tanto *Estimativa Simples*, contendo apenas uma métrica, quanto *Estimativa Múltipla*, onde é selecionado a combinação de duas métricas.

O componente *Histórico de Estimativas* contém todas as estimativas geradas pelas métricas presentes no sistema. Desta forma, juntando as informações das estimativas com informações sobre o tempo real do projeto, que é informada pelo usuário, é possível visualizar qual métrica ou quais métricas retornam as estimativas mais próximas do real. Essas informações ficam presentes no componente *Métricas Recomendadas*, e é exposta

ao usuário sempre que este decida gerar uma nova estimativa.

4.2 Aplicação Seletiva de Métricas de Software

No estudo de caso proposto, foram selecionados alguns projetos com o intuito de gerar estimativas utilizando as métricas e verificar a eficácia dos resultados obtidos. Foram buscados projetos *open source* em repositórios online, pela liberdade de acesso as informações necessárias. Desta forma, para realização do estudo foram utilizados alguns projetos disponíveis no GitHub¹ que permite a visualização de projetos postados por usuários do site de forma gratuita. Devido ao fato que para gerar as estimativas tanto das métricas de projetos quanto das métricas de teste é necessário que se tenha informações completas da documentação dos projetos buscados. Foram analisados somente os projetos que continham tais informações, isso foi um dificultador do estudo, pois, nem todas as empresas ou instituições disponibilizam ou tem o hábito de gerar a documentação dos sistemas. Foram selecionados apenas os projetos que possuíam diagramas de casos de uso, documento de requisitos, datas de início e término dos projetos etc. Além de um plano detalhado sobre os testes realizados no *software*, esse item se torna crucial para a geração das estimativas utilizando as métricas de testes.

Durante o estudo foram selecionados 14 projetos para a geração das estimativas, estes projetos deviam cumprir alguns itens básicos para serem utilizados no estudo. A separação dos projetos e dos itens avaliados podem ser vistos na Tabela 1. Foi observado se os projetos possuíam itens em sua documentação que são essenciais para a geração das estimativas. Entre os itens estão, o documento de requisitos, relação do tempo de início e término do projeto, plano de teste ou afins. Também foi avaliado se os projetos possuíam informações que retratassem a equipe de desenvolvimento, como o conhecimento sobre a linguagem utilizada no projeto.

¹<https://github.com/>

Tabela 4.1: Projetos utilizados no estudo

PROJETO	MÉTRICAS					Documento de Requisitos	Data Inicial e Final do Projeto	Plano de Teste	Detalhamento da Equipe
	APF	Cocomo II	UCP	APT	TCP				
SIGS	V	V	V	V	V	V	V	V	V
MedExplorer	V	V	V	X	X	V	V	X	V
Projeto Flash	V	V	V	V	V	V	V	V	V
Deputinder	V	V	V	V	V	V	V	V	V
Dr. Down	V	V	V	X	X	V	V	X	V
Face Travels	V	V	V	X	X	V	V	X	V
Projeto E-Commerce	V	V	V	X	X	V	V	X	V
Merenda Mais	V	V	V	V	V	V	V	V	V
Receita Mais	V	V	V	V	V	V	V	V	V
CRP	V	V	V	X	X	V	V	X	V
Quero Cultura	V	V	V	V	V	V	V	V	V
Escola X	V	V	V	V	V	V	V	V	V
SiMCTA	V	V	V	X	X	V	V	X	V
Laços de Alegria	V	V	V	V	V	V	V	V	V

Os projetos que continham as especificações receberam uma marcação verde, caso estes itens fossem suficientes para a geração da estimativa, a mesma marcação era feita sobre a métrica. As marcações em vermelho estão nos projetos que não possuem algum componente, ou suas informações não são suficientes para a geração da estimativa. Neste caso, todos os projetos apresentados podem ser utilizados nas estimativas correspondentes as métricas APF, UCP, Cocomo 2, que são métricas de *software*. Porém apenas oito projetos apresentam um documento que retrate os testes que foram realizados. Por esse motivo o restante dos projetos não foram utilizados nas métricas APT e TCP que são métricas de cálculo de esforço de teste.

4.3 Implementação da ferramenta de estimativa de *software*

A ferramenta foi totalmente desenvolvida utilizando a linguagem de programação Java FX, que é uma das vertentes presentes na linguagem Java. Esta foi escolhida por possuir uma rica interface remetendo a aplicações web com características de um programa tradicional de um *desktop*. Além disso, a linguagem é formada pelo princípio de orientação a objetos, isso permite manter o sistema seguro, flexível e escalável.

Para o desenvolvimento da interface foi utilizado o *Scene Builder*. Esta é uma ferramenta de *layout* que permite a criação de interface interativas em XML ligadas ao Java FX de forma mais rápida e prática. Foi utilizado também o SGBD *MySQL* para a persistência das informações com o banco de dados.

Capítulo 5

Avaliação

Afim de verificar a exatidão das métricas presentes na ferramenta, foram realizadas estimativas sobre 14 projetos presentes no repositório GitHub. Assim foi possível comparar os resultados obtidos em cada estimativa tanto nas estimativas de projeto quanto nas estimativas de horas de teste de software. Cada projeto foi avaliado para que fossem coletas as informações requeridas por cada métrica na geração das estimativas.

5.1 Estimativa Simples

Como pode ser observado na Tabela 2, a coluna tempo real do projeto total contém a quantidade real dos dias que foram gastos para desenvolver cada projeto. Essas informações foram utilizadas na comparação com as métricas APF, Cocomo II e UCP que são métricas que calculam o esforço total do projeto. Na coluna tempo real do projeto de teste estão informados os tempos reais que foram dedicados a testes nos projetos, essas informações foram utilizadas nas métricas de teste APT e TCP. Nas posições contendo marcações na cor vermelha estão os projetos que não puderam ser estimados pelo fato de não conterem algum requisito especificado na Tabela 1, que foi o caso somente das métricas para o cálculo de esforço em teste.

Com o estudo de caso, pôde ser verificado a eficiência nos resultados obtidos por cada métrica na estimativa simples. No caso das métricas de projeto, a métrica APF obteve sua maior taxa de acerto no projeto Laços de Alegria com um percentual de 100% de acerto. E obteve seu menor percentual de aproximadamente 75% de acerto. A métrica UCP chegou a alcançar 100% no projeto Laços de Alegria e teve seu valor percentual mais distante nos projetos DepuTinder e Receita Mais de aproximadamente 88% de acerto. A métrica de projeto que obteve as estimativas mais próximas da reais foi a Cocomo II que

nos projetos SIGS, DepuTinder e Laços de Alegria atingiu 100% de acerto. O menor percentual da métrica Cocomo II foi no Projeto Flash com aproximadamente 85% de acerto.

Tabela 5.1: Relação das Estimativas Simples

PROJETO	METRICAS					Tempo Real do Projeto Total (Dias)	Tempo Real do Projeto de Teste (Dias)
	APF (Dias)	CocomoII (Dias)	UCP (Dias)	APT (Dias)	TCP (Dias)		
SIGS	55	61	60	14	13	61	15
MedExplorer	1000	930	900	X	X	932	X
Projeto Flash	60	55	50	8	10	48	12
DepuTinder	50	55	63	12	15	55	14
Dr. Down	130	150	160	X	X	149	X
Face Travels	40	35	45	X	X	38	X
Projeto E-Commerce	130	116	150	X	X	119	X
Merenda Mais	75	60	70	17	20	66	17
Laços de Alegria	80	80	80	19	21	80	20
CRP	64	75	75	X	X	73	X
Quero Cultura	66	72	76	20	19	73	18
Escola X	55	60	55	12	10	57	14
SiMCTA	79	84	80	X	X	83	X
Receita Mais	61	70	60	15	18	68	17

Nas estimativas utilizando as métricas de teste também foram obtidos bons resultados. A métrica APT teve seu melhor percentual no projeto Merenda Mais com 100% de acerto no número de dias que deveriam ser dedicados a testes no projeto. O menor percentual da métrica APT foi no Projeto Flash com 67%. A métrica TCP teve seu melhor percentual no projeto Laços de Alegria com um percentual de aproximadamente 95% e pior desempenho foi no projeto Escola X com aproximadamente 71% de acerto.

5.2 Estimativa Múltipla

Como já foi citado anteriormente, o usuário além de ter a possibilidade da geração de estimativas simples com as métricas presentes na ferramenta, pode também realizar estimativas que combinem o resultado de duas métricas disponíveis. Na Tabela 3 pode ser observado os resultados de esforço das combinações entre as métricas APF/UCP, Cocomo 2/UCP e APT/TCP. Devido ao fato de que a métrica Cocomo 2 possui em seu cálculo a utilização da métrica APF foi descartada na ferramenta qualquer combinação entre estas duas métricas, afim de evitar redundâncias.

Tabela 5.2: Relação das Estimativas Múltiplas

PROJETO	MÉTRICAS (Média)			Tempo Real do Projeto Total (Dias)	Tempo Real do Projeto de Teste (Dias)
	APF/UCP (Dias)	Cocomo II/UCP (Dias)	APT/TCP (Dias)		
SIGS	58	61	14	61	15
MedExplorer	950	915	X	932	X
Projeto Flash	55	53	9	48	12
DepuTinder	57	59	14	55	14
Dr. Down	145	155	X	149	X
Face Travels	43	40	X	38	X
Projeto E-Commerce	140	133	X	119	X
Merenda Mais	73	65	19	66	17
Laços de Alegria	80	80	20	80	20
CRP	70	75	X	73	X
Quero Cultura	71	74	20	73	18
Escola X	55	58	11	57	14
SiMCTA	80	82	X	83	X
Receita Mais	61	65	17	68	17

Do mesmo modo que na Tabela 2 as duas últimas colunas da Tabela 3 correspondem ao tempo real total do projeto e ao tempo real de teste a ser dedicado no projeto respectivamente. As marcações na cor vermelha correspondem aos projetos que não puderam ser estimados pelas métricas de teste, por não conterem em sua documentação informações referentes aos testes realizados no sistema.

Nas estimativas múltiplas geradas pela combinação APT/TCP o percentual de acerto nos projetos DepuTinder, Laços de Alegria e Receita Mais foi de 100% e o pior desempenho foi no Projeto Flash com 75% de acerto. Mostrando que a utilização da estimativa múltipla para o cálculo do esforço em teste em alguns casos podem devolver resultados mais satisfatórios.

Nas estimativas utilizando as combinações entre as métricas de projeto a combinação APF/UCP chegou a valores muito próximos dos reais. A estimativa alcançou 100% de percentual de acerto no projeto Laços de Alegria, estimando que o tempo a ser dedicado ao projeto seria de 80 dias, mesmo número do tempo real gasto no projeto. Na estimativa Cocomo II/UCP os resultados foram ainda mais satisfatórios chegando a um percentual de 100% nos projetos SIGS e Laços de Alegria.

Afim de ilustrar melhor os resultados obtidos nas estimativas, foram gerados gráficos das estimativas simples e múltiplas de *software*. Essas estimativas podem ser vistas na Figura 5.1 e das estimativas simples e múltiplas de teste presentes na Figura 5.2.

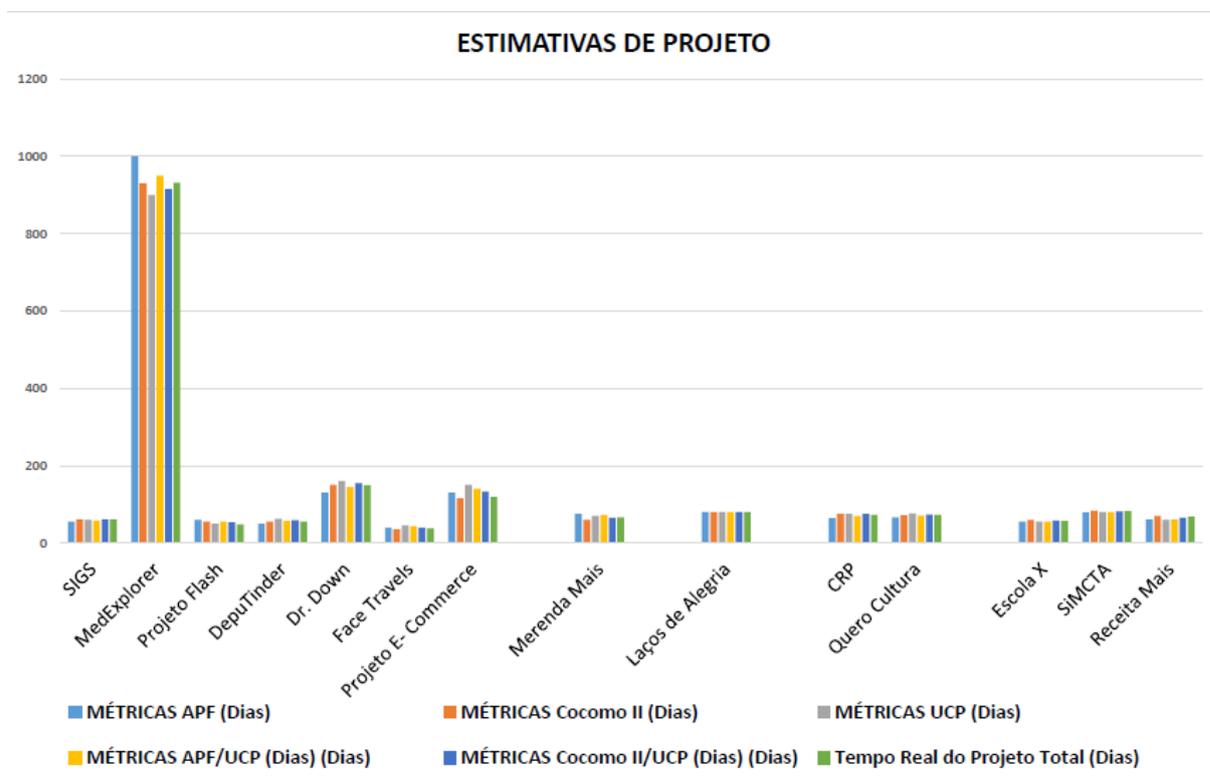


Figura 5.1: Estimativas de Projeto

No gráfico da Figura 5.1 os valores que representam o tempo real gasto nos projetos pode ser visto na cor verde. Nas demais cores estão as estimativas apresentadas pelas métricas APF, Cocomo II, UCP, Cocomo II/UCP e APF/UCP. Como pode ser visto, os valores estimados com as métricas, se comparados com os valores reais, obtemos números muito próximos. Vale destacar que o uso das estimativas múltiplas auxiliou bastante para que o esforço gerado fosse o mais exato possível.

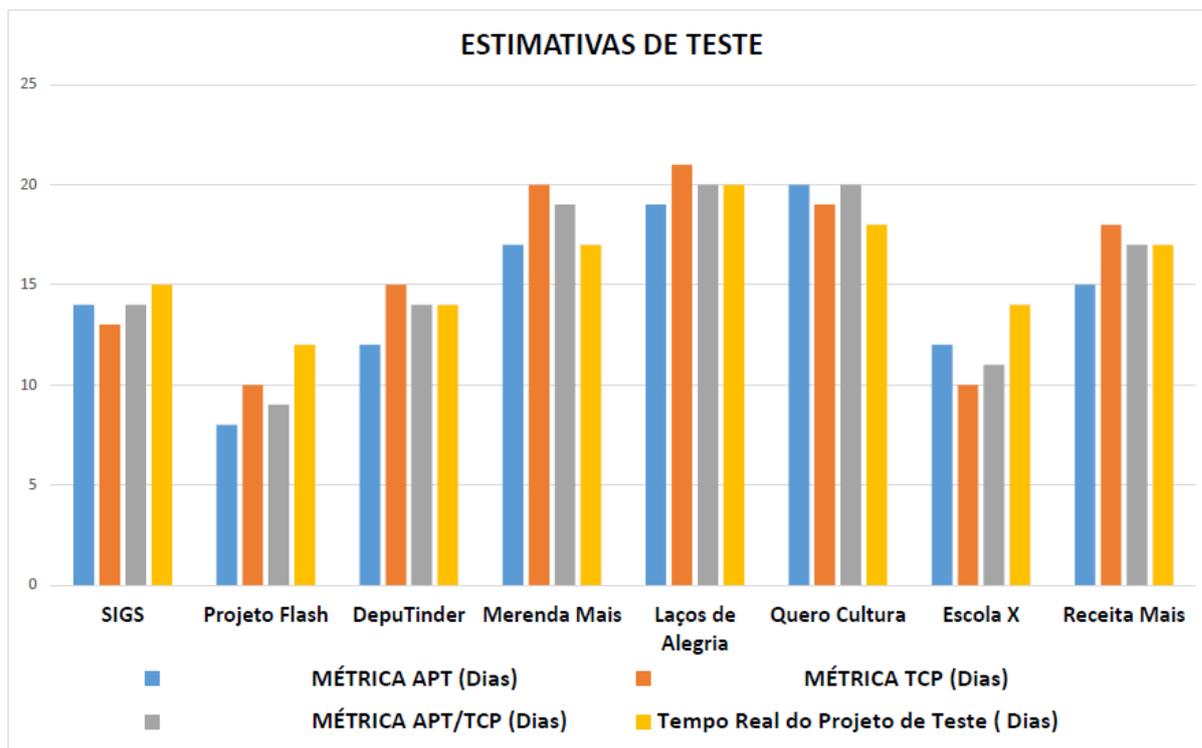


Figura 5.2: Estimativas de Teste

No gráfico da Figura 5.2 o valor real gasto nos projetos pode ser visto na cor laranja. Nas demais cores estão as estimativas geradas pelas métricas APT, TCP e pela combinação APT/TCP. Neste caso também é visível que o uso da média dos resultados obtidos pelas métricas APT e TCP trouxe estimativas mais próximas das reais.

Segundo Pressman (2005) os testes de software representam em média de 30% a 40% do tempo total do projeto. No livro *Facts and fallacies of software engineering* (Fatos e falácias da engenharia de software) (GLASS, 2002, p. 100), o autor remete que a porcentagem de testes dedicadas ao sistema é de aproximadamente 30%. No estudo realizado neste trabalho, foram calculadas também as horas de testes sobre o tempo total gasto nos projetos, afim de avaliar o percentual utilizado em testes em cada projeto. Os valores obtidos pelas métricas de teste obtiveram uma variação na maioria dos casos de 25% a 30% do tamanho total do projeto. Os valores reais coletados dos projetos obtiveram uma variação entre 24% e 27%. Desta forma, as métricas de teste presentes na ferramenta, obtiveram estimativas muito próximas dos valores reais apresentados nos projetos, e também estimativas próximas ao percentual descrito na teoria como valores aceitáveis de teste em projetos.

Capítulo 6

Conclusão

O estudo de caso proposto permitiu o controle de diversos projetos ao mesmo tempo, possibilitando ao usuário a geração de estimativas utilizando múltiplas métricas de projeto e de teste. Além de poder gerar estimativas com base na combinação de resultados de duas métricas distintas.

Com base no estudo de caso realizado, concluímos que na geração de estimativas utilizando apenas uma métrica, os resultados da métrica de projeto Cocomo II foram os mais exatos. A métrica atingiu em alguns casos valores idênticos aos reais, seguida pelas métricas UCP e APF. Na geração de estimativas simples sobre o esforço de teste, a métrica APT obteve valores mais próximos dos reais em relação a métrica TCP. Entretanto em alguns casos ambas conseguiram um percentual de 100% de acerto. Nas estimativas múltiplas de tempo total de projeto a combinação Cocomo II/UCP se destacou por devolver um percentual muito alto de acerto. Essa estimativa alcançou na maior parte dos projetos valores superiores as demais métricas de projeto. Na estimativa múltipla entre APT/TCP os valores obtidos também superaram as estimativas geradas pelas métricas APT e TCP sozinhas. Isso mostra que a utilização da média entre os resultados das métricas na geração de estimativas podem significar ganhos consideráveis no cálculo de esforço dos projetos na fase que antecede a implementação.

Como trabalhos futuros estão incluídos a geração de estimativas utilizando um número maior de projetos. Isso poderia ajudar na sugestão ao usuário sobre qual estimativa devolve os melhores resultados para o tipo de projeto inserido, aumentando a aceitação da ferramenta e minimizando os erros gerados pelas estimativas. Além do aperfeiçoamento da ferramenta e seus componentes, permitindo assim uma maior facilidade e aceitação por parte dos usuários.

Referências

- PRESSMAN, R.; MAXIM, B. *Engenharia de Software - 8ª Edição*. [S.l.]: McGraw Hill Brasil, 2016. ISBN 978-85-8055-534-9. page.11, page.66
- MAGELA, R. *Engenharia de software aplicada: fundamentos*. Rio de Janeiro–RJ: Alta Books, 2006. page.11, page.77
- STANDISH, G. The CHAOS Report 2012. 2012. Disponível em: <<http://standishgroup.com/>>. page.11
- STANDISH, G. The CHAOS Report 2015. 2015. Disponível em: <<http://standishgroup.com/>>. page.11
- MATIELI, L. V.; ARAUJO, F. O. de. ESTIMATIVA DE TESTE DE SOFTWARE: LEVANTAMENTO BIBLIOGRÁFICO EM VEÍCULOS BRASILEIROS E INTERNACIONAIS. *Relatórios de Pesquisa em Engenharia de Produção*, v. 15, n. 1, p. 1, 2016. Disponível em: <<http://www.rpep.uff.br/index.php/RPEP/article/view/11>>. page.22, page.1515
- NETO, A. C. D. *Introdução a Teste de Software*. [s.n.], 2007. Disponível em: <https://www.researchgate.net/publication/266356473_Introducao_a_Testes_de_Software>. page.22
- WENDERLICH, A. FERRAMENTA DE CÁLCULO DE ESTIMATIVAS DE SOFTWARE. Disponível em: <<http://www.fattocs.com/files/pt/livro-apf/citacao/AlexandreWenderlich-2008.pdf>>. page.22
- CARVALHO, N. G. de; ESSADO, M.; AMBROSIO, A. M. Estimativa de teste de software: A análise de ponto de teste aplicado em um sistema de processamentos de transações financeiras. *São José dos Campos, São Paulo, Brasil*, 2012. Disponível em: <<http://mtc-m21b.sid.inpe.br/col/sid.inpe.br/mtc-m21b/2014/04.20.22.08/doc/estimativa-teste-software.pdf>>. page.22, page.1414
- HEIMBERG, V.; GRAHL, E. A. Estudo de Caso de Aplicação da Métrica de Pontos de Casos de Uso numa Empresa de Software. *XIV Seminário de Computação–SEMINCO, Outubro*, 2005. Disponível em: <<http://www.inf.furb.br/seminco/2005/artigos/130-vf.pdf>>. page.22
- BELGAMO, A.; FABBRI, S. Um Estudo sobre a Influência da Sistematização da Construção de Modelos de Casos de Uso na Contagem dos Pontos de Casos de Uso. *III Simpósio Brasileiro de Qualidade de Software*, 2004. Disponível em: <http://www.mct.gov.br/upd_blob/0002/2831.pdf>. page.22, page.33, page.88, page.1515

- VAVASSORI, F. B.; SOUZA, E. W.; FIAMONCINI, J. C. Ferramenta CASE para gerenciamento de projetos e métricas de software. *Tools Session of XV SBES, Rio de Janeiro-RJ*, 2001. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/sbes/2001/027.pdf>>. page.22
- PATEL, N.; GOVINDRAJAN, M.; MAHARANA, S.; RAMDAS, S. Test Case Point Analysis. *Cognizant Technology Solutions, White Paper*, 2001. Disponível em: <<http://msqaa.org/Newsletter/TestCasePointAnalysis.pdf>>. page.22, page.1212
- LIMA, A. C.; PINTO, A. F.; CAMPOS, D. S. Pontos de Função. Disponível em: <http://www.gestao.mt.gov.br/images/files/Advogado_teste22112012154044.pdf>. page.22, page.33
- VAZQUEZ, C. E.; SIMOES, G. S.; ALBERT, R. M. Análise de pontos de função. *São Paulo: Érica*, 2003. Disponível em: <<http://www.fattocs.com/files/pt/apresentacoes/Palestra9JGES.pdf>>. page.22, page.33
- AGUIAR, M. ESTIMANDO OS PROJETOS COM COCOMO II. out. 2017. page.22, page.33
- MANUAL, M. D. COCOMO II. 1995. Disponível em: <http://csse.usc.edu/csse/affiliate/private/COCOMOII_2000/COCOMOII-040600/model_man.pdf>. page.22, page.33
- SOMMERVILLE, I. *Engenharia de Software, Tradução de André Maurício de Andrade Ribeiro; Revisão técnica de Kechi Hirama*. [S.l.]: São Paulo, Addison Wesley, 2003. page.55, page.77
- PRESSMAN, R. S. *Software engineering: a practitioner's approach*. [S.l.]: Palgrave Macmillan, 2005. page.55, page.66, page.2525
- VAZQUEZ, C. E.; SIMÕES, G. S.; ALBERT, R. M. Análise de pontos de função: medição, estimativas e gerenciamento de projetos de software. *Editora Érica, São Paulo*, v. 3, 2003. page.55
- PRESSMAN, R. S. *Engenharia de software*. [S.l.]: Makron books São Paulo, 1995. page.66
- FERNANDES, A. A. Gerência de software através de métricas: garantindo a qualidade do projeto, processo e produto. 1995. page.66
- TRODO, L. D. Uso de métricas nos testes de software. 2009. Disponível em: <<http://www.lume.ufrgs.br/handle/10183/18574>>. page.66, page.1515
- TAVARES, H.; CARVALHO, A. E. S. de; CASTRO, J. Medição de pontos de Função a Partir da Especificação de Requisitos. In: *WER*. [s.n.], 2002. p. 278–298. Disponível em: <<https://pdfs.semanticscholar.org/4375/fa5528a9062c7ed478876786164d3fb5f364.pdf>>. page.66
- BOEHM, B. W.; Clark; Horowitz; Brown; Reifer; Chulani; MADACHY, R.; STEECE, B. *Software Cost Estimation with Cocomo II with Cdrom*. 1st. ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2000. ISBN 978-0-13-026692-7. page.77

- PETERS, J. F.; PEDRYCZ, W. Engenharia de software: teoria e prática. *Rio de Janeiro: Campus*, v. 681, n. 519.683, p. 2, 2001. page.88
- BOEHM, B.; CLARK, B.; HOROWITZ, E.; WESTLAND, C.; MADACHY, R.; SELBY, R. Cost models for future software life cycle processes: COCOMO 2.0. *Annals of Software Engineering*, v. 1, n. 1, p. 57–94, dez. 1995. ISSN 1022-7091, 1573-7489. Disponível em: <<https://link.springer.com/article/10.1007/BF02249046>>. page.88
- KARNER, G. Resource estimation for objectory projects. *Objective Systems SF AB, Citeseer*, v. 17, 1993. page.99, page.1010, page.1111
- VEENENDAAL, V. E. P.; DEKKERS, T. Test point analysis : a method for test estimation. *Project control for software quality : proceedings of the combined 10th European Software Control and Metrics conference and the 2nd SCOPE conference on software product evaluation, April 27-29, 1999, Herstmonceux, England, 1999*. Disponível em: <<https://research.tue.nl/en/publications/test-point-analysis-a-method-for-test-estimation>>. page.1111, page.1212
- WENDERLICH, A. Ferramenta de cálculo de estimativas de software. 2008. page.1414
- JÚNIOR, L. P. PROTOTIPO DE UMA FERRAMENTA PARA GERÊNCIA DE CUSTOS EM PROJETOS DE SOFTWARE BASEADA NO MODELO PMBOK. p. 83, 2003. page.1414
- KUSUMOTO, S.; MATUKAWA, F.; INOUE, K.; HANABUSA, S.; MAEGAWA, Y. Estimating effort by use case points: method, tool and case study. In: *10th International Symposium on Software Metrics, 2004. Proceedings*. [S.l.: s.n.], 2004. p. 292–299. page.1414
- GLASS, R. L. *Facts and fallacies of software engineering*. [S.l.]: Addison-Wesley Professional, 2002. page.2525