

LUIS FERNANDO DEMARCHI DO AMARAL

**ESTUDO EMPÍRICO DE DIFERENTES ABORDAGENS DA MÉTRICA
DE PONTOS DE CASO DE USO PARA DETERMINAÇÃO DO
ESFORÇO DE UM PROJETO DE SOFTWARE**

São Paulo
2014

LUIS FERNANDO DEMARCHI DO AMARAL

**ESTUDO EMPÍRICO DE DIFERENTES ABORDAGENS DA MÉTRICA
DE PONTOS DE CASO DE USO PARA DETERMINAÇÃO DO
ESFORÇO DE UM PROJETO DE SOFTWARE**

Monografia apresentada ao PECE –
Programa de Educação Continuada em
Engenharia da Escola Politécnica da
Universidade de São Paulo como parte
dos requisitos para a conclusão do curso
de MBA em Tecnologia de Software.

Área de Concentração: Tecnologia de
Software

Orientador: Prof. Dr. Sérgio Martins
Fernandes

São Paulo
2014

AGRADECIMENTOS

Primeiramente a minha família, e especialmente minha mãe pelo apoio dado, não só durante o desenvolvimento dessa monografia mas durante toda minha vida e nos momentos e decisões mais difíceis.

Ao orientador Prof. Dr. Sérgio Martins Fernandes, pelos conselhos, conhecimento, experiência e principalmente paciência durante a realização desse trabalho.

Aos professores e colegas do curso de MBA em Tecnologia de Software pelo conhecimento transmitido, sem o qual tal trabalho não seria possível.

Aos amigos pelo suporte e incentivo.

E aos membros dessa banca pela presença nessa apresentação, pelas críticas e sugestões de melhoria.

RESUMO

A gerência efetiva de um projeto de software possui disciplinas gerência de tempo e custo, intimamente relacionadas, que são de grande relevância para gerentes de software e os envolvidos no seu desenvolvimento, já que a entrega do produto no tempo estimado é tipicamente essencial para o cliente. Foram diversas as tentativas de se obter uma métrica ou técnica para que uma estimativa com maior precisão fosse feita a um custo factível. Algumas dessas métricas são bastante comuns no mercado de desenvolvimento de software, como a Análise de Pontos de Função e a Métrica de Pontos de Caso de Uso. A primeira pode ser utilizada após o levantamento de funcionalidades do sistema, enquanto a segunda pode ser estabelecida ainda durante a fase de levantamento de requisitos. Esse trabalho efetua uma análise empírica baseada no estudo de diferentes abordagens da métrica de Pontos de Caso de Uso, numa tentativa de avaliá-las com base em fatores como complexidade e precisão.

Palavras-chaves: Pontos de Caso de Uso. Métricas de Software. Estimativa de tempo. Casos de Uso

ABSTRACT

The effective management of a software project comprises disciplines such as time and cost management, which are closely related, and are of great relevance for software managers and those involved in its development, since the delivery of the product in the estimated time is typically essential for the client. There were several attempts to obtain a metric or a technique to more accurately predict effort and, consequently, project time. Some of these metrics are quite common in the software development community, such as Function Point Analysis and Use Case Points. The first one can be used after the elicitation of system functionalities, while the second one can be further established during the elicitation of requirements. This paper executes an empirical analysis based on the study of different variations of the Use Case Points technique in an attempt to evaluate them based on factors such as complexity and precision.

Key words: Use case points. Software metrics. Time estimate. Use cases.

LISTA DE ILUSTRAÇÕES

Figura 1 – Fluxo de Atividades para a aplicação da metodologia	44
Figura 2 – Comparativo de tempo entre as abordagens estudadas	59
Figura 3 – Erro relativo das abordagens estudadas	60

LISTA DE TABELAS

Tabela 1 – Métricas de Produto	19
Tabela 2 – Métricas de Processo	19
Tabela 3 – Tabela de Peso - Grau de Complexidade das Funções	22
Tabela 4 – Vantagens e Desvantagens das Métricas Avaliadas	29
Tabela 5 – Pontuação dos atores	30
Tabela 6 – Peso dos casos de uso	31
Tabela 7 – Complexidade de Fatores Técnicos	31
Tabela 8 – Influência do Fator Técnico	32
Tabela 9 – Complexidade de Fatores Ambientais	32
Tabela 10 – Diferenças entre as abordagens Original e Simplificada	34
Tabela 11 – Classificação dos Atores na métrica USP	37
Tabela 12 – Classificação das pré-condições na métrica USP	38
Tabela 13 – Classificação dos Cenários na métrica USP	38
Tabela 14 – Classificação das exceções na métrica USP	39
Tabela 15 – Classificação das pós-condições na métrica USP	39
Tabela 16 – Fatores Ambientais Adicionais na estimativa por pontos de Caso de Uso	41
Tabela 17 – Classificação dos atores e cenário principal do caso de uso	52
Tabela 18 – Aplicação dos Fatores de ajuste técnicos	52
Tabela 19 – Aplicação dos Fatores de ajuste ambientais	53
Tabela 20 – Ajuste de fatores - Abordagem Original	54
Tabela 21 – Obtenção da Produtividade - Abordagem Original	54
Tabela 22 – Cálculo das estimativas - Abordagem Original	55
Tabela 23 – Ajuste de fatores - Pontos de Caso de Uso Simplificada	55
Tabela 24 – Obtenção da Produtividade - Pontos de Caso de Uso Simplificada	56
Tabela 25 – Cálculo das estimativas - Pontos de Caso de Uso Simplificada	56
Tabela 26 – Obtenção dos Pontos de Caso de Uso Desajustados - Pontos de Tamanho de Caso de Uso	57
Tabela 27 – Ajuste de Fatores - Pontos de Tamanho de Caso de Uso	58
Tabela 28 – Obtenção da Produtividade - Pontos de Tamanho de Caso de Uso	58
Tabela 29 – Cálculo das estimativas - Pontos de Tamanho de Caso de Uso	58
Tabela 30 – Comparativo de tempo estimado nas abordagens avaliadas e o tempo efetivo	59
Tabela 31 – Erro Relativo das abordagens avaliadas	60

LISTA DE ABREVIATURAS E SIGLAS

AFP	Adjusted Function Points
AIE	Arquivos de Interface Externa
ALI	Arquivos Lógicos Internos
CE	Consulta Externa
COCOMO	Constructive Cost Model
EE	Entrada Externa
EF	Environmental Factors
FP	Function Points
ISO	International Organization for Standardization
KDSI	Kilo Delivered Source Instructions
SE	Saída Externa
TCF	Technical Complexity Factors
UCP	Use Case Points
UFP	Unadjusted Function Points
UML	Unified Modeling Language
UUCP	Unadjusted Use Case Points
VAF	Value Adjustment Factor

SUMÁRIO

1	Introdução	13
1.1	Motivação e relevância do problema	13
1.2	Objetivo	13
1.3	Justificativa	14
1.4	Estrutura do trabalho	15
2	Revisão Bibliográfica	16
2.1	Introdução	16
2.1.1	Conceitos Fundamentais sobre métricas de Software	17
2.1.2	Classificação das métricas de Software	17
2.1.3	Métricas Relevantes no Desenvolvimento de Software	18
2.1.4	Utilidade de Métricas de Software	18
2.2	Métricas de tamanho de software mais conhecidas e seu contexto de aplicação	20
2.2.1	Análise de Pontos de Função	21
2.2.1.1	Introdução	21
2.2.1.2	Metodologia	21
2.2.1.3	Contexto de Aplicação	23
2.2.2	COCOMO	24
2.2.2.1	Introdução	24
2.2.2.2	Metodologia	25
2.2.2.3	Contexto de Aplicação do COCOMO	27
2.2.3	Análise de Pontos de Caso de Uso	27
2.2.3.1	Introdução	27
2.2.3.2	Metodologia	27
2.2.3.3	Contexto de Aplicação	28
2.2.4	Comparação entre as métricas	28
2.3	A métrica de Pontos de Caso de Uso	29
2.3.1	Introdução	29
2.3.2	Abordagem Original	29
2.3.2.1	Metodologia para aplicação da métrica	30
2.3.2.1.1	Pontos de Caso de Uso Desajustados	30
2.3.2.1.2	Fatores de complexidade técnica	31
2.3.2.1.3	Fatores Ambientais	32
2.3.2.2	Comentários gerais sobre a técnica	33

2.3.3	Análise de Pontos de Caso de Uso Simplificada	33
2.3.3.1	Apresentação da abordagem	33
2.3.3.1.1	Complexidade dos atores na métrica Pontos de Caso de Uso	34
2.3.3.1.2	Fatores de Ajuste	35
2.3.3.1.3	Passos vs. Transações	35
2.3.3.2	Comentários gerais sobre a técnica por seus criadores	36
2.3.4	Pontos de Tamanho de Caso de Uso	36
2.3.4.1	Apresentação da abordagem	36
2.3.4.1.1	Classificação dos Atores	37
2.3.4.1.2	Classificação das pré-condições	38
2.3.4.1.3	Classificação do cenário principal	38
2.3.4.1.4	Classificação dos cenários alternativos	38
2.3.4.1.5	Classificação das exceções	39
2.3.4.1.6	Classificação das pós-condições	39
2.3.4.1.7	Aplicação dos valores de ajuste	40
2.3.4.2	Comentários pós-aplicação da técnica	40
2.3.5	Análise de Pontos de Caso de Uso com adição de novos fatores ambientais	40
2.3.5.1	Apresentação da abordagem	40
2.3.5.2	Comentários gerais pós-aplicação da técnica	41
2.4	Considerações do Capítulo	41
2.4.1	Síntese geral	41
2.4.2	Determinação da abordagem mais complexa e mais simples para estudo de caso	42
3	Aplicação das abordagens	43
3.1	Metodologia	43
3.1.1	Seleção de casos de uso	44
3.1.1.1	Descrição da etapa	44
3.1.1.2	Justificativa da etapa	45
3.1.2	Revisão e padronização dos casos de uso	45
3.1.2.1	Descrição da etapa	45
3.1.2.2	Justificativa da etapa	46
3.1.3	Contagem e classificação dos atores	46
3.1.3.1	Descrição da etapa	46
3.1.3.2	Justificativa da etapa	46
3.1.4	Contagem dos passos dos casos de uso	46
3.1.4.1	Descrição da etapa	46

3.1.4.2	Justificativa da etapa	47
3.1.5	Aplicação da abordagem original	47
3.1.5.1	Descrição da etapa	47
3.1.5.2	Justificativa da etapa	49
3.1.6	Aplicação da métrica de Pontos de Caso de Uso Simplificada	49
3.1.6.1	Descrição da etapa	49
3.1.6.2	Justificativa da etapa	49
3.1.7	Aplicação da métrica de Pontos de Tamanho de Caso de Uso	49
3.1.7.1	Descrição da etapa	49
3.1.7.2	Justificativa da etapa	50
3.1.8	Metodologia de análise dos resultados	50
3.1.8.1	Descrição da etapa	50
3.1.8.2	Justificativa da etapa	50
3.2	Execução da metodologia proposta	51
3.2.1	Seleção, revisão e padronização dos casos de uso	51
3.2.2	Contagem de atores e passos	51
3.2.3	Aplicação da abordagem original	51
3.2.4	Aplicação da métrica de Pontos de Caso de Uso Simplificada	54
3.2.5	Aplicação da métrica de Pontos de Tamanho de Caso de Uso	56
3.2.6	Discussão dos resultados	59
3.3	Considerações do Capítulo	60
4	Considerações Finais	62
4.1	Conclusões	62
4.2	Contribuições do trabalho	63
4.3	Trabalhos Futuros	64
	Referências	65

1 INTRODUÇÃO

1.1 Motivação e relevância do problema

A necessidade de uma estipulação de tempo de desenvolvimento com um maior índice de precisão é fundamental para empresas de software, já que a satisfação de seus clientes está intimamente relacionada à entrega de seus produtos no tempo adequado. Algumas técnicas e métricas para estimativa de complexidade do software foram desenvolvidas para servir de base à predição de tempo e consequentemente custo de um sistema, tais como COCOMO, Pontos de Função, Pontos de História e Pontos de Casos de Uso. Dentre estas, a métrica de Pontos de Caso de Uso (UCP) se mostra muito útil no desenvolvimento de software que contempla essa maneira de se eliciar requisitos, já que se utiliza de casos de uso obtidos na fase de análise do ciclo de desenvolvimento.

O uso da métrica de Pontos de Caso de Uso é justificável para alguns autores quando a comparamos com outras métricas de estimativa de tamanho de software. Para Schofield, Armemtrout e Trujillo (2013) existem vantagens na aplicação da técnica, tais como a agilidade e facilidade para se obter a estimativa, se a comparamos ao FP, e a independência de tecnologia e facilidade de validar sua aplicação repetidamente se a comparamos com a métrica de Pontos de História. Mishra, Hazra e Mall (2011) afirmam, ainda, que a métrica de Pontos de Caso de Uso possui uma grande facilidade para ser automatizada quando comparada a métricas tradicionais.

A técnica de Pontos de Caso de Uso, no entanto, não é mantida por uma organização mundialmente conhecida como a técnica de Pontos de Função, que ainda possui um padrão ISO para que seja aplicada de maneira uniforme (20926:2009, 2009). Essa falta de padronização abre margem a diferentes maneiras de aplicar a técnica. Essas diferentes maneiras de se aplicar a técnica fez com que algumas variações da abordagem original passassem a ter utilizadas. Dentre elas podemos citar: pontos de caso de uso simplificada, pontos de caso de uso com adição de fatores ambientais, a métrica de pontos de tamanho de caso de uso.

Baseando-se em algumas dessas vertentes da métrica de UCP, tal estudo avalia o grau de precisão com que se pode obter o esforço de desenvolvimento de software.

1.2 Objetivo

O presente trabalho tem por objetivo avaliar a precisão de abordagens da técnica de Pontos de Casos de Uso como base para estimativa do esforço de projetos de

desenvolvimento de software. Para que tal objetivo fosse atingido foi necessário um estudo inicial das diferentes abordagens dessa métrica que existem atualmente e a partir da seleção de duas destas, uma de maior complexidade e uma mais simplicista, aplicar cada uma das técnicas selecionadas em um conjunto de projetos, juntamente com a abordagem original, para avaliar a precisão do esforço de desenvolvimento obtido por elas.

A análise das duas variantes de Pontos de Caso de Uso foi feita aplicando cada uma delas em um número pré-determinado de projetos com o objetivo de levantar a produtividade de equipe. A partir da determinação de tal produtividade as técnicas foram reaplicadas em outro conjunto de projetos e com a produtividade já determinada anteriormente foi avaliada a precisão que tais métricas conseguiram atingir os resultados coletados.

1.3 Justificativa

Tendo em vista a dificuldade de empresas de software em determinar o esforço de seus projetos e os problemas ocasionados por essa dificuldade, a utilização de métricas de estimativa com maiores índices de precisão se torna essencial em qualquer processo de desenvolvimento de software. Dentre esses problemas ocasionados pela falta de precisão na estimativa de tempo de um projeto, pode-se citar: cronogramas imprecisos, qualidade reduzida devido à falta de tempo, custos adicionais, sobrecarga de trabalho aos envolvidos no projeto. Heemstra (1992) apresenta alguns casos de fracasso na área de desenvolvimento de software, devido a projetos com estimativas ineficientes:

- O sistema de automatização de educação do governo holandês foi três vezes o preço previsto.
- O desenvolvimento de software para subsídios de aluguel do governo holandês foi de duas vezes o preço previsto.
- A automatização da polícia alemã custou, ao invés de US\$ 21 milhões como prevista, US\$ 43 milhões. Além disso, o software não atendia as necessidades básicas.

Mukhopadhyay, Vicinanza e Prietula (1992) afirmam ainda que fazer uma estimativa muito abaixo do tempo real pode ocasionar em julgamentos irreais sobre o benefício do projeto para a empresa, dados seus custos reais. Além disso, muitos desses projetos são lançados somente para cumprir seu orçamento, omitindo características

importantes do sistema, ou deixando etapas como a de testes para trás. Fazer uma estimativa que seja muito além do tempo real pode acarretar em projetos inflacionados, aumentando o custo do projeto, ao colocar menos pressão sobre os programadores para serem mais produtivos. Adicionalmente, projetos com potencial benefício podem ser erroneamente rejeitados por serem muito caros.

Observando ainda a crescente utilização do paradigma de orientação a objetos e de Casos de Uso como forma de elicitação de requisitos, inclusive sendo esta recomendada por designers da UML como entrada nos processos de design, validação e verificação do ciclo de desenvolvimento de software (JACOBSON, 1992), a técnica de Pontos de Caso de Uso se torna apropriada para empresas que utilizam tal artefato em seus processos. Segundo Aguiar (2003), a aplicação de tal técnica ainda não é amplamente difundida se comparada à métrica de Pontos de Função e ainda não possui um padrão internacional para ser utilizada. No entanto, possui algumas vantagens quando comparadas a outras métricas tradicionais. Atualmente, ainda não existe um padrão internacional para a sua aplicação, apesar de seu uso ter se difundido. Dada sua relevância num conjunto de métricas de estimativa funcionais, esse trabalho, vem então avaliar a precisão com que algumas variantes da métrica de Pontos de Caso de Uso conseguem se aproximar do esforço real obtido no desenvolvimento de software.

1.4 Estrutura do trabalho

Este trabalho é composto de quatro capítulos, incluindo este que introduz o problema inicial da dificuldade de se medir atributos de software e a partir destes, obter-se indicadores relevantes e úteis sobre os projetos analisados.

O segundo capítulo apresenta o embasamento teórico utilizado para se entender os conceitos de métricas, bem como as métricas mais utilizadas em processos de software. Um estudo mais aprofundado da métrica de Pontos de Caso de Uso é realizado, a qual será utilizada empiricamente para análise de sua eficácia na obtenção de estimativas em projetos.

O terceiro capítulo aborda a metodologia utilizada para a análise de diferentes abordagens da métrica de Pontos de Caso de Uso, bem como os resultados obtidos empiricamente através dessa análise. Os resultados coletados são analisados e uma breve discussão sobre a validade das métricas nesse estudo é apresentado.

No último capítulo, são apresentadas as principais conclusões e contribuições deste trabalho para a melhoria do processo de gestão de estimativa de tamanho e as perspectivas de trabalhos futuros.

2 REVISÃO BIBLIOGRÁFICA

2.1 Introdução

O processo de medição faz parte do cotidiano do ser-humano. Desde o cálculo de menor percurso para chegar a um destino até o cálculo de custos para uma construção civil, as atividades humanas frequentemente se utilizam do processo da medição. Para Fenton e Pfleeger (1997), medição “é um processo pelo qual números ou símbolos são atribuídos a propriedades de entidades do mundo real de modo a descrevê-las de acordo com regras claramente definidas”. A definição dessas regras e a atribuição de símbolos ou números a tais entidades permite que estas sejam comparáveis de acordo com um atributo em comum.

O processo de medição faz parte de um processo ainda maior, o de quantificação. A quantificação de uma entidade pode ser obtida diretamente (medição) ou indiretamente, através do processo de cálculo, onde medidas são combinadas de modo a quantificar um atributo cujo valor deseja-se entender.

Pressman (2005) define medida como “uma indicação quantitativa da extensão, dimensão, capacidade ou tamanho de algum atributo de um produto ou de um processo” e métrica como “uma medida quantitativa do grau em que um sistema, componente ou processo possui determinado atributo”.

Para a melhor compreensão do processo de medição, alguns conceitos devem ser entendidos (TRAINNING EDUCATION SERVICES, 2013):

- **Unidade de medida:** Menor unidade que define o trabalho a ser realizado.
- **Medida:** Quantificação de um atributo (unidade de medida) do projeto.
- **Medição:** Coleta de uma medida do projeto em um determinado instante.
- **Métrica:** Função de dois ou mais valores de medidas.
- **Indicador:** Uma medida ou métrica utilizada como critério para tomada de decisões
- **Meta:** Nível de resultado que deve ser atingido no projeto.

A necessidade de se medir um software é pertinente a muitos dos envolvidos em sua concepção e criação. O processo de medir as características de um software ou sistema pode oferecer uma visão de quanto tempo será necessário para seu desenvolvimento, de quanto ele está condizente com especificações de requisitos ou mesmo

em que pontos do sistema será necessário um maior investimento em verificações e validações.

2.1.1 Conceitos Fundamentais sobre métricas de Software

A frase de DeMarco apud Fenton e Pfleeger (1997) “Você não pode predizer nem controlar o que você não pode medir” dá dimensão da importância de se medir um software, já que tais medidas são necessárias para controle de processos e atividades e utilizadas para predição de tempo e custo de um projeto. Essa necessidade na obtenção de medidas que possam auxiliar nos processos de desenvolvimento de software proporcionou o surgimento de algumas métricas de software ainda amplamente utilizadas por gerentes de projeto, engenheiros de software e analistas de sistemas.

Uma métrica de software, segundo Goodman (2004), é a “Contínua aplicação de técnicas baseadas em medição para o processo de desenvolvimento de software e seus produtos, para fornecer informações oportunas e relevantes para melhorar esse mesmo processo e produtos”.

Uma métrica de software, segundo Pressman (2005), deve ser simples e computável e ser facilmente derivável. Deve ser empírica e intuitivamente convincente, satisfazendo noções intuitivas do engenheiro sobre o atributo do produto em causa. Não deve ser dependente de linguagem de programação e consistente no uso de unidades e dimensões.

Métricas de software devem ser ainda (TRAINNING EDUCATION SERVICES, 2013):

- **Objetivas:** reduzir ou minimizar a influência do julgamento pessoal na coleta, cálculo e análise dos resultados
- **Fáceis de entender:** comprehensíveis por várias audiências (gerentes, técnicos, usuários)
- **Efetivas no custo:** o valor da informação obtido como resultado das medições deve ser maior que o custo de coletar, armazenar e calcular.
- **Informativas:** devem propiciar informação que possibilitem avaliar acertos de decisões e ações no passado e prever a possibilidade de eventos futuros.

2.1.2 Classificação das métricas de Software

Existem várias áreas de aplicação de métricas de software e várias maneiras de classificar tais métricas. Dentre as áreas mais importantes podemos destacar: estimativa de custo e tamanho; controle do desenvolvimento de projetos; predição de

níveis de qualidade; checagem da arquitetura do sistema; pontos propensos a mudanças como código-fonte refatorável; e principalmente geração de informação gerencial como índices de produtividade, qualidade e efetividade de processo.

Fenton e Pfleeger (1997) classificam as métricas de software em três categorias:

- **Métricas de Processos:** relacionadas à ordem e relação de atividades dependentes de uma medida de tempo.
- **Métricas de Produtos:** utilizadas para medir artefatos, entregáveis ou documentos resultantes de um processo ou atividade.
- **Métricas de Recursos:** utilizadas para medir entidades requeridas por um processo e atividade

Muitos outros autores, como por exemplo, Singh, Singh e Singh (2011), preferem classificar as métricas somente em Métricas de Processos e Métricas de Produtos. Para Gopu (2003) existem duas subcategorias de métricas de processos, sendo essas as métricas estruturais, utilizadas no início do desenvolvimento de software tais como medidas de estabilidade, complexidade de revisão, fluxo de informações; e as métricas de código, aplicadas em estágios posteriores do ciclo de desenvolvimento, como Linhas de Código (LOC) e complexidade de McCabe's que mede complexidade dos ciclos existentes no código fonte do sistema.

Métricas de Produtos, de acordo com Bundschuh e Dekkers (2008), podem medir atributos arquiteturais como número de componentes e acoplamento, atributos de qualidade como performance, usabilidade, manutenção, além de complexidade do software, tamanho funcional e documentação existente para o produto. Esse trabalho aborda, em sua essência, métricas de software de produto utilizadas para estimativa de tamanho funcional.

2.1.3 Métricas Relevantes no Desenvolvimento de Software

Algumas métricas de software se difundiram na indústria de software e são comumente utilizadas. Nas tabelas 1 e 2 algumas dessas métricas são apresentadas, seguindo a classificação de métricas de processos e produtos.

2.1.4 Utilidade de Métricas de Software

Métricas de software possuem vários benefícios quando utilizadas ao longo do processo de desenvolvimento, como por exemplo:

- No estudo comparativo de abordagens de design de sistemas de software.

Tabela 1 – Métricas de Produto

	Medida básica	Exemplo
Tamanho	Linhas de Código, Pontos de Função	Total de linhas de código por aplicação, Total de Pontos de Função por aplicação
Qualidade	Defeitos e tamanho da aplicação	Defeitos por Ponto de Função pós entrega, defeitos por linha de código pós entrega
Documentação	Páginas e documentação	Número de páginas por documento de requisito e por módulo de especificação
Complexidade do Sistema	Complexidade e módulos	Dados por módulo

Fonte: (BUNDSCUH; DEKKERS, 2008)

Tabela 2 – Métricas de Processo

	Medida básica	Exemplo
Esforço	Esforço em horas despendido pela equipe	Esforço por projeto ou por etapa de projeto
Custo	Custo e tamanho	Unidade monetária por ponto de função
Qualidade	Densidade dos defeitos por projeto ou fase	Defeitos por ponto de função
Eficiência	Esforço e tamanho	Esforço em horas por ponto de função

Fonte: (BUNDSCUH; DEKKERS, 2008)

- Na análise, comparação e estudo crítico de linguagens de programação.
- Na comparação e avaliação de produtividade e capacidade de pessoas envolvidas no desenvolvimento de software.
- Na preparação de especificações de qualidade de software.
- Na avaliação de complexidade do código gerado.
- Na avaliação do esforço a ser colocado no design e desenvolvimento dos sistemas de software.
- Na provisão de feedback para gerentes sobre o progresso e a qualidade durante várias fases do ciclo de desenvolvimento.
- Na alocação de recursos de testes para o software gerado.

No entanto, métricas de software possuem limitações, como relatado por Singh, Singh e Singh (2011), tais como:

- Dificuldade e algumas vezes alto custo para que sejam aplicadas.

- Dificuldade para validação de dados históricos e empíricos, já que a verificação da precisão de métricas de software muitas vezes necessita dessa base histórica de informações.
- Utilidade para o gerenciamento de produtos de software, mas não para avaliar o desempenho da equipe técnica.
- Definição e derivação geralmente baseada em convenções não padronizadas e dependentes de ferramentas e ambiente de trabalho.
- Estimativas dependentes de variáveis muitas vezes não conhecidas pela equipe.

2.2 Métricas de tamanho de software mais conhecidas e seu contexto de aplicação

O tamanho de um software é essencial na estimativa de tempo de desenvolvimento e na obtenção de seu preço final, já que existe uma relação direta entre quantidade de funcionalidades ou linhas de código e tempo para desenvolvimento. Esse tamanho do software pode ser medido de diversas formas, desde o tamanho que o software ocupa em disco, bem como a quantidade de linhas de código, ou quantidade de interfaces com o usuário e relatórios. No entanto, uma maneira comum de se medir o software é através do número de funções requeridas pelo usuário, comumente conhecido como tamanho funcional. A necessidade de se medir o tamanho funcional de um software não é recente. Inúmeras métricas foram elaboradas, sendo que algumas dessas têm um uso relativamente maior e se adequam melhor a contextos específicos. A seguir são apresentadas as métricas de estimativa de tamanho mais difundidas e comumente utilizadas bem como a metodologia abordada para que sejam aplicadas e o contexto de aplicação que melhor de adequa a elas.

A seção aborda métricas de tamanho de software, pois independente do processo de software utilizado no mercado, os questionamentos relativos a tempo e custo de um projeto são relevantes a muitos dos envolvidos no projeto. Além das métricas citadas nesta seção, há diversas outras. As três métricas aqui apresentadas (Análise de Pontos de Função, COCOMO e Análise de Pontos de Caso de Uso) foram selecionadas com base na atual utilização pelo universo da Tecnologia da Informação (TI) e na existência de referências bibliográficas sólidas. Algumas métricas, embora amplamente utilizadas, como por exemplo a Análise de Pontos de História, não foram descritas nessa seção devido a baixa quantidade de referências encontradas.

2.2.1 Análise de Pontos de Função

2.2.1.1 Introdução

Uma das primeiras métricas a se popularizar e ser difundida e utilizada na comunidade de desenvolvimento de software foi a Métrica de Pontos de Função (TRAINNING EDUCATION SERVICES, 2013).

A métrica de Pontos de Função foi pensada para suprir a necessidade existente de uma métrica de tamanho que seja independente de linguagem de programação. Consiste em medir as funcionalidades do sistema através de documentos de especificação de requisitos, especificação de design ou do sistema em si.

Essa técnica foi apresentada pela primeira vez em 1979 em uma conferência por seu criador Allan Albrecht. Desde então tal técnica vem sendo amplamente utilizada. Contratos de compra de software podem utilizar as medidas obtidas pela métrica como base para estipular esforço e preço e seus valores são amplamente aceitos. São várias as organizações de consultoria e os treinamentos comerciais disponíveis para o aprendizado da métrica.

2.2.1.2 Metodologia

Albrecht, o criador da técnica propõe dois estágios de processo para o cálculo de Pontos de Função, sendo esses:

- Contagem de fatores que são manifestações externas da aplicação, correspondentes as interações entre usuários e sistema. Posterior atribuição de pesos aos fatores obtidos que refletem valor funcional ao cliente.
- Ajuste do resultado para efeito de outros fatores relacionados à complexidade técnica da aplicação.

O primeiro estágio consiste basicamente em identificar as fronteiras da aplicação, identificar as entradas, saídas, consultas, arquivos lógicos internos e arquivos de interface externos, atribuindo um número de pontos de função a cada um desses fatores de acordo com sua complexidade e ao final realizar a soma dos pontos de função. Tal resultado captura as necessidades funcionais do usuário.

Os itens essenciais para a contagem de pontos de função podem ser divididos em:

- **Arquivos Lógicos Internos (ALI):** São grupos de dados ou informações de controle logicamente relacionados e facilmente reconhecidos pelo usuário. São mantidos dentro da fronteira da aplicação.

Tabela 3 – Tabela de Peso - Grau de Complexidade das Funções

Tipo de Função	Grau de Complexidade		
	3,0	4,0	6,0
Entrada Externa (EE)	3,0	4,0	6,0
Saída Externa (SE)	4,0	5,0	7,0
Consulta Externa (CE)	3,0	4,0	6,0
Arquivo Lógico Interno (ALI)	7,0	10,0	15
Arquivo de Interface Externa (AIE)	5,0	7,0	10,0

Fonte: (ALBRECHT; GAFFNEY, 1983)

- **Arquivos de Interface Externa (AIE):** São arquivos de outra aplicação acessados pelo sistema em questão ou arquivos compartilhados por outros sistemas.
- **Funções de Entrada Externa (EE):** São classificadas como entradas externas as funções elementares que processam dados ou informações que venham de fora do controle da aplicação, como por exemplo documentos digitados, transações em discos, leituras ópticas.
- **Funções de Saída Externa (SE):** São processos elementares que enviam dados de controle para fora da fronteira da aplicação. As funções de saída devem ter para cada grupo de informações um formato diferente ou uma lógica de processamento diferente.
- **Funções de Consulta Externa (CE):** São processos responsáveis por enviar dados ou informações para fora da fronteira da aplicação. Diferenciam-se das saídas externas por não possuírem processamento intermediário.

A contagem desses itens elementares permite a contagem inicial dos pontos de função sem ajuste e, para tanto, utilizam-se fatores de peso de acordo com a complexidade analisada para cada tipo de função do sistema, conforme a tabela 3. É importante ressaltar que tal contagem deve sempre manter a perspectiva do usuário.

No segundo estágio o valor obtido na primeira etapa passa por reajustes baseados em características adicionais. Esses fatores não seriam facilmente concebidos na primeira etapa e possuem impacto na contagem final. Para cada característica adicional, é necessária sua classificação em: 0,0 - nenhuma influência; 1,0 - influência mínima; 2,0 - influência moderada; 3,0 - influência média; 4,0 - influência significante; 5,0 - influência forte.

As catorze características do sistema necessárias no ajuste dos pontos de caso de uso são:

- Comunicação de dados

- Funções distribuídas
- Desempenho
- Carga de máquina
- Volume de transações
- Atualização On-Line
- Facilidade de Operação
- Eficiência do usuário final
- Complexidade de Processamento
- Reusabilidade
- Facilidade de instalação
- Entrada de dados On-line
- Múltiplos locais
- Facilidade de Alteração

O cálculo final dos pontos de caso de uso ajustados deve ser realizado através da equação 2.1.

$$PFA = UFP * (0,65 + 0,01 * VAF) \quad (2.1)$$

Onde,

PFA: Pontos de Função Ajustados

UFP: Pontos de Função Não Ajustados

VAF: Value Adjustment Factor (somação de todos os graus de influência)

2.2.1.3 Contexto de Aplicação

Como relata Lokan (2005), apesar do uso da técnica ter decaído, a técnica Pontos de Função ainda é dominante no mercado, com exceção apenas do Reino Unido. O conhecimento e experiência do uso da técnica permitem que adaptações sejam realizadas com êxito e que várias ferramentas fossem criadas para auxílio de seus usuários. A técnica possui várias críticas relativas à subjetividade das medidas, aspectos não cobertos e dúvidas relativas a seus passos, no entanto seu tradicionalismo ainda é responsável por que muitas organizações optem por seu uso.

2.2.2 COCOMO

2.2.2.1 Introdução

Dois anos após a primeira apresentação da métrica de Pontos de Função foi apresentada uma nova técnica para estimativa de tamanho funcional. O COCOMO (*Constructive Cost Model*), desenvolvido por Dr. Barry Boehm em 1981 (MERLO; SCHETT, 2003), refletia o desenvolvimento de software praticado na época, onde as aplicações eram desenvolvidas seguindo o paradigma estrutural, o uso da internet não era difundido e os modelos de processos de software eram essencialmente baseados no modelo cascata. Desde então, os processos de desenvolvimento passaram por diversas modificações assim como os tipos de software desenvolvidos. Aplicações de tempo real passaram a ter significativa importância e processos de design e manutenção a ter tanto valor quanto a implementação de novos produtos, devido a necessidade de reusabilidade de software. A técnica COCOMO então teve se passar por adaptações, sendo que num esforço conjunto da Universidade da Califórnia, uma nova proposta foi apresentada para suprir as necessidades e lacunas existentes, surgindo então o COCOMO II (MERLO; SCHETT, 2003). Embora a métrica tenha passado por adaptações e sua segunda versão seja a que melhor se adeque aos processos de desenvolvimentos atuais, essa seção detalha sua versão original, que é mais simples e fornece o grau de aprofundamento cabível ao foco do presente trabalho.

O COCOMO é um modelo desenvolvido para estimar esforço, prazo, custo e tamanho da equipe para um projeto de software. O modelo básico calcula o esforço do desenvolvimento de software em função das linhas de código estimadas. O modelo intermediário em função do tamanho do programa e fatores que incluem avaliações subjetivas do produto, do hardware, do pessoal e atributos do projeto. O modelo avançado incorpora os fatores da versão intermediária incluindo avaliações dos impactos dos atributos do software da equipe em cada etapa do projeto (JUNIOR; SANCHES, 2006). Posteriormente esses modelos serão detalhados e suas equações apresentadas.

O modelo COCOMO utiliza-se de equações elaboradas por Boehm baseadas num estudo de centenas de projetos e considera fatores tangíveis / intangíveis e quantitativos / qualitativos no desenvolvimento desses projetos. A unidade de medida utilizada em suas equações é a KDSI (*Kilo Delivered Source Instructions*) que são as instruções de código entregues ao cliente medidas em milhares. Os Pontos de Função Desajustados podem ser convertidos em linhas de código de acordo com tabelas definidas para cada tipo de linguagem, dessa forma podem ser utilizados juntamente com o COCOMO. Os ajustes técnicos e de ambientes utilizados pela técnica de Pontos de Função, por possuírem pesos iguais não são contemplados pelo COCOMO, que

atribui pesos distintos para seus fatores. Essa constitui uma das principais diferenças entre as técnicas.

Ao contrário de outros modelos de estimativa de custos, o COCOMO é um modelo aberto, para que todos os detalhes sejam publicados, incluindo: as equações de estimativa de custos subjacentes, as suposições feitas no modelo, as definições dos termos utilizados, os custos incluídos na estimativa de maneira explícita".

O modelo original do COCOMO possuía duas equações principais para suas estimativas, sendo 2.2 utilizada na estimativa de esforço e 2.3 utilizada na estimativa de prazo ótimo.

$$E = (A * S^b) * F \quad (2.2)$$

Onde,

- **E** = Esforço medido em Horas/mês (H/m)
- **A** = Constante do modelo de acordo com o modelo utilizado
- **S** = Tamanho do sistema em KDSI (milhares de linha de código)
- **b** = Fatores de performance não-lineares
- **F** = Fatores de performance lineares - 15 direcionadores de esforço e custo utilizados nos modelos intermediários e avançados

$$PO = C * E^n \quad (2.3)$$

Onde,

- **PO** = Prazo Ótimo expresso em meses
- **C** = Constante do modelo de acordo com o modelo utilizado
- **E** = Esforço medido em Horas/mês (H/m)
- **n** = Fatores de performance não-lineares

2.2.2.2 Metodologia

A primeira versão do COCOMO possuía três níveis. O nível mais básico calcula a estimativa de esforço e consequente custo através de funções de tamanho de programa expressas em milhares de instruções de código geradas (equações 2.2 e 2.3). A versão intermediária fornece a estimativa de esforço através do tamanho do código e um conjunto de 15 direcionadores de custo que contém características de hardware, pessoais e de atributos de projeto. Já a versão avançada ou detalhada do COCOMO,

incorpora os elementos da versão intermediária com distinção para todas as etapas do ciclo de desenvolvimento, tais como Análise, Design, Desenvolvimento.

O COCOMO diferencia os tipos de software e seus ambientes de desenvolvimento, e de acordo com estes, constantes diferentes são utilizadas nas suas equações, alterando significativamente os resultados. Os modelos de aplicação reconhecidos pelo COCOMO são:

- **Modelo orgânico (convencional):** caracteriza-se por equipes pequenas, ambientes de desenvolvimento altamente familiares e estáveis, onde a maior parte das pessoas tem experiência com sistemas similares.
- **Modelo Restrito:** caracterizado por sistemas operando com grandes restrições e contexto complexo de hardware, software, regras e procedimentos operacionais.
- **Modelo Difuso:** a equipe possui nível intermediário de experiência com as aplicações e possui grande heterogeneidade.

As fórmulas de aplicação da técnica, adaptadas para cada modelo são:

Equações para se estimar o esforço (H/M) de desenvolvimento (em KDSI):

- **Orgânico:**

$$H/M = 2,4 * (KDSI)^{1,05} \quad (2.4)$$

- **Difuso:**

$$H/M = 3,0 * (KDSI)^{1,12} \quad (2.5)$$

- **Restrito:**

$$H/M = 3,6 * (KDSI)^{1,20} \quad (2.6)$$

Equações para se estimar o prazo ótimo (PO) de desenvolvimento (em H/M):

- **Orgânico:**

$$PO = 2,5 * (H/M)^{0,38} \quad (2.7)$$

- **Difuso:**

$$PO = 2,5 * (H/M)^{0,35} \quad (2.8)$$

- **Restrito:**

$$PO = 2,5 * (H/M)^{0,32} \quad (2.9)$$

A segunda versão da técnica COCOMO surgiu com o objetivo de suprir três necessidades principais: novos processos de software, necessidade de tomada de decisão baseada em informações incompletas e novos fenômenos como reuso e tamanho. Dentro as principais diferenças entre a primeira e segunda versão da técnica podemos citar o endereçamento de novas três fases do ciclo de vida: desenvolvimento, design e pós-arquitetura na segunda versão. Inclusão da volatilidade dos requisitos na segunda versão para cálculo da estimativa. Alteração no padrão de contagem de comandos lógicos. Determinação de equação exponencial da estimativa através de cinco fatores ao invés de três modelos de desenvolvimento.

2.2.2.3 Contexto de Aplicação do COCOMO

A técnica COCOMO ainda é utilizada amplamente atualmente, sendo essa associada a outras técnicas de estimativa como principalmente a de Pontos de Função para suprir deficiências que a aplicação individual da técnica possua (ROLLO, 2006).

Como relata Syavasya (2011) “O modelo de estimativa de custos COCOMO é usado por milhares de gerentes projeto de software”, e associado ao fato de ser um modelo aberto, incentiva pesquisadores e profissionais da comunidade de engenharia de software a utiliza-lo de forma independente. Por esses motivos existe uma maior facilidade para que empresas iniciantes possuem utilizar a técnica em seus processos de desenvolvimento de software.

2.2.3 Análise de Pontos de Caso de Uso

2.2.3.1 Introdução

Com o surgimento da UML e sua popularização, a utilização de casos de uso como forma de escrever requisitos passou a ser alta no segmento de desenvolvimento de software. Uma nova técnica de estimativa de tamanho de software chamada de Pontos de Casos de Uso foi apresentada então por Karner (1993). Essa seção descreve de maneira não aprofundada no que consiste a métrica e posteriormente, na seção 2.3.2, uma análise mais rigorosa é realizada a seu respeito. A técnica é baseada na análise de Pontos de Função, diferenciando-se por medir atores e caso de uso ao invés de dados e transações e por produzir como resultado uma estimativa de tamanho ao invés de esforço.

2.2.3.2 Metodologia

A técnica de Pontos de Casos de Uso é composta de três etapas, sendo essas: classificação da complexidade dos atores em simples, média e complexa, classifica-

ção da complexidade dos casos de uso de acordo com o número de transações, e ajuste dos pontos de caso de uso de acordo com fatores técnicos e de ambientes relevantes ao projeto.

2.2.3.3 Contexto de Aplicação

Estudos de caso mostram que a técnica de Pontos de Casos de Uso pode atingir altos índices de precisão de estimativas de esforço apesar de não ser amplamente utilizada (ANDA; DREIEM; SJOBERG, 2001). Ferramentas foram desenvolvidas para que ela seja automatizada, facilitando ainda sua aplicação.

A técnica está sujeita a fatores subjetivos como, por exemplo, o julgamento pessoal dos fatores técnicos e ambientais. No entanto, a análise de Pontos de Função também possui tal subjetividade, não caracterizando, portanto, tais fatores como os responsáveis pelas diferenças de resultados entre ambas as técnicas.

A necessidade da geração de artefatos específicos para a aplicação da técnica restringe sua aplicação em diversos contextos, sendo essa uma de suas principais desvantagens.

A abordagem de desenvolvimento de software baseada em casos de uso é adotada internacionalmente, sendo inclusive parte integrante do Processo Unificado. No entanto, para equipes de projeto que não utilizam tal técnica, a elaboração de tais documentos somente para estimativa de tamanho do software pode ser custosa e não precisa, devido à falta de padronização da escrita dos casos de uso.

A falta de padrões específicos para a escrita de casos de uso é um fato reconhecido internacionalmente como relata Fowler (2005):

Não existe uma maneira padrão para escrever o conteúdo de um caso de uso, diferentes formatos trabalham bem em casos diferentes.

Esse fato abre margem a múltiplas interpretações da definição de caso de uso e, consequentemente, múltiplas maneiras de escrevê-los, e resultados não precisos quando a técnica de pontos de caso de uso é aplicada, restringindo ainda mais a possibilidade de utilização da técnica.

2.2.4 Comparação entre as métricas

Com o objetivo de melhor avaliar as diferenças entre as métricas avaliadas, a tabela 4 contém características resumidas das métricas que implicam em suas vantagens e desvantagens.

Tabela 4 – Vantagens e Desvantagens das Métricas Avaliadas

Característica	Análise de Pontos de Função	COCOMO	Análise de Pontos de Caso de Uso
Momento da aplicação	Etapa de design (intermediária)	Etapa de design (intermediária)	Etapa de análise (inicial)
Amplamente difundida?	Sim	Sim	Não
Facilidade para automação	Pequena	Média	Grande
Padronização	Grande	Grande	Pequena

2.3 A métrica de Pontos de Caso de Uso

2.3.1 Introdução

A utilização da técnica de pontos de caso de uso em diferentes ambientes foi responsável por adaptações na métrica para que seu resultado chegasse mais próximo do real. Esses diferentes ambientes, caracterizados por processos de software distintos, diferentes maneiras de se escrever casos de uso e características tecnológicas distintas são responsáveis pelas adaptações que a técnica tem sofrido. Nesta seção a metodologia de aplicação da abordagem original será apresentada novamente e de maneira detalhada por ser o tema central desse trabalho. Embora uma apresentação da técnica já tenha sido realizada anteriormente na seção 2.2.3, o detalhamento de como ela deve ser aplicada é necessário para que as abordagens derivadas sejam entendidas com maior clareza.

Nessa seção serão introduzidas abordagens derivadas da métrica original, bem como sua metodologia para aplicação e os comentários e resultados obtidos pelos seus autores. A primeira abordagem derivada tem como objetivo simplificar a metodologia original sem que a precisão desta seja afetada, a segunda abordagem acrescenta aspectos dos casos de uso não contemplados anteriormente e a terceira acrescenta novos fatores de ambiente para obtenção da estimativa.

2.3.2 Abordagem Original

Essa seção detalha a abordagem original, proposta por Karner (1993), para a estimativa de tamanho de software baseada em casos de uso, onde a primeira sub-seção contém os detalhes necessários para a aplicação da técnica e a segunda sub-seção relata os resultados obtidos pós sua elaboração/aplicação quando comparados aos obtidos pela aplicação da técnica de Pontos De Função.

Tabela 5 – Pontuação dos atores

Complexidade	Definição	Peso
Simples	Um ator é considerado simples, caso represente outro sistema com uma interface de programação definida	1,0
Média	A complexidade média é atribuída a: 1. Interação com outro sistema através de um protocolo 2. Interação humana através de interface gráfica	2,0
Complexa	O ator é considerado complexo quando existe a interação através de interface gráfica	3,0

Fonte: (KARNER, 1993)

2.3.2.1 Metodologia para aplicação da métrica

A métrica original de Pontos de Caso de Uso, proposta por Karner em 1993, é uma adaptação da técnica de Pontos de Função, concebida originalmente para processos de desenvolvimento estruturados, – para paradigma de desenvolvimento orientado a objetos. A partir da medição das funcionalidades através dos casos de uso, obtém-se um valor de estimativa chamado de Ponto de Caso de Uso Desajustado (*Unadjusted Use Case Points - UUCP*), que de maneira similar a técnica de Pontos de Função passa por um ajuste baseado em fatores técnicos e posteriormente por um ajuste baseado em fatores de ambiente, que não eram contemplados pela técnica de Pontos de Função.

2.3.2.1.1 Pontos de Caso de Uso Desajustados

Para a obtenção da pontuação de casos de uso desajustada (UUCP) primeiramente é necessário obter a complexidade associada aos atores do caso de uso em questão. Somente os atores concretos são levados em consideração, ou seja, aqueles que representam uma população de usuários operacionalmente ligados como um grupo de atores que compartilham o mesmo conjunto básico de responsabilidades. Tais atores são classificados de acordo com a tabela 5.

Ao valor de complexidade obtido dos atores do caso de uso é adicionado ainda o valor da complexidade obtido pelos passos do cenário principal, de acordo com a tabela 6.

Karner (1993) diz em sua tese que se existem informações adicionais sobre o projeto, pode-se aprimorar o resultado com fatores técnicos e de ambiente, do contrário esse resultado pode ser utilizado como estimativa.

Tabela 6 – Peso dos casos de uso

Complexidade	Definição	Peso
Simples	O caso de uso é considerado simples se possui três transações ou menos, incluindo cursos alternativos.	5,0
Média	O caso de uso é considerado de complexidade média quando possui de 3 a 7 transações, incluindo fluxos alternativos.	10,0
Complexa	O caso de uso é considerado complexo quando possui mais de sete transações, incluindo fluxos alternativos	15,0

Fonte: (KARNER, 1993)

Tabela 7 – Complexidade de Fatores Técnicos

Fi	Fator contribuinte à complexidade	Wi
F1	Distribuição do sistema	2,0
F2	Desempenho da aplicação	1,0
F3	Eficiência do usuário final	1,0
F4	Complexidade do processamento interno	1,0
F5	Reusabilidade do código em outras aplicações	1,0
F6	Facilidade de instalação	0,5
F7	Usabilidade	0,5
F8	Portabilidade	2,0
F9	Facilidade de modificação	1,0
F10	Concorrência	1,0
F11	Funcionalidades especiais de segurança	1,0
F12	Acesso direto a dispositivos de terceiros	1,0
F13	Treinamento especial aos usuários	1,0

Fonte: (KARNER, 1993)

2.3.2.1.2 Fatores de complexidade técnica

Os fatores de complexidade técnica (TCF) da métrica de Pontos de Caso de Uso se referem à dificuldade que existirá para o sistema ser construído. Tais fatores são bastante similares a da técnica de Pontos de Função, sendo que seus valores e constantes foram propostos por Albrecht e Gaffney (1983) e são obtidos de acordo com a equação 2.10.

$$TCF = 0,6 + 0,01 * \sum_{i=1}^{13} F_i * W_i \quad (2.10)$$

Onde Wi podem ser obtidos da tabela 7 e Fi de acordo com a tabela 8.

Tabela 8 – Influência do Fator Técnico

Influência	Valor
Sem Influência	0,0
Incidental	1,0
Moderado	2,0
Médio	3,0
Significativo	4,0
Essencial	5,0

Tabela 9 – Complexidade de Fatores Ambientais

Fi	Fator contribuinte à complexidade	Wi
F1	Familiaridade com o processo de desenvolvimento	1,5
F2	Experiência na aplicação	0,5
F3	Experiência com orientação a objetos	1,0
F4	Experiência do líder do projeto	0,5
F5	Motivação	1,0
F6	Estabilidade dos requisitos	2,0
F7	Membros da equipe com dedicação parcial	-1,0
F8	Dificuldade da Linguagem de programação	-1,0

Fonte: (KARNER, 1993)

2.3.2.1.3 Fatores Ambientais

A medida dos fatores ambientais (EF), contidos na tabela 9, auxilia a estimativa da eficiência do projeto e é obtida de maneira semelhante aos fatores técnicos associados ao caso de uso, de acordo com a equação 2.11.

$$EF = 1,4 - 0,03 * \sum_{i=1}^8 Fi * Wi \quad (2.11)$$

Wi pode ser obtido de acordo com a tabela 9 e Fi de maneira idêntica aos Fatores de complexidade técnica.

Finalmente o valor final do caso de uso é obtido a partir da equação 2.12.

$$UCP = UUCP * TCF * EF \quad (2.12)$$

O resultado obtido (UCP) deve ser multiplicado pela quantia de recursos necessários gastos por caso de uso. Caso esses dados não estejam disponíveis, duas opções podem ser adotadas. Karner (1993) propõe um fator de 20 horas por caso de uso. Schneider e Winters (SCHNEIDER; WINTERS, 1998) propõem a seguinte metodologia:

NF = soma dos fatores ambientais de F1 a F6 que tem valor menor que 3 adicionado à soma dos fatores ambientais de F7 a F8 que tem valor maior que 3.

Se o valor de NF for menor ou igual a 2, então o valor de horas por caso de uso é igual a 20, caso NF seja igual a 3 ou 4, temos então, 28. Caso NF seja maior que 4 o tempo dispendido por caso de uso será de 36.

2.3.2.2 Comentários gerais sobre a técnica

Nos resultados apresentados por Karner foi demonstrada uma correlação entre a quantidade de recursos necessária ao projeto e os resultados dos Pontos de Caso de Uso. Karner ainda questionou a precisão da métrica se comparada a outras, por ter sido elaborada a partir de poucos projetos. A aplicação da métrica em um maior número de projetos foi realizada por outros estudiosos do assunto, respondendo ao questionamento inicial.

2.3.3 Análise de Pontos de Caso de Uso Simplificada

Uma segunda variante da métrica de Pontos de Caso de uso, desenvolvida por M. Ochodek, J. Nawrocki e K. Kwarciak (OCHODEK; NAWROCKI; KWARCIAK, 2011) na Polônia em 2010, possibilita a simplificação da abordagem tradicional, descartando aspectos que não impliquem em alterações significativas no resultado final obtido pela estimativa. A técnica foi proposta a partir da observação de um conjunto de catorze projetos sem e com as simplificações propostas pela abordagem.

2.3.3.1 Apresentação da abordagem

O processo de estimativa de tamanho de software, embora essencial, pode ser custoso quando utiliza-se de determinadas técnicas. Em algumas situações surge um *trade-off* entre o tempo dispendido e esforço na estimativa e o grau de precisão que ela pode oferecer. A técnica de simplificação de Pontos de Caso de Uso é justificável em tal contexto e tem por objetivos:

- Simplificar os métodos de estimativa de esforço sem comprometer sua acurácia.
- Fazer com que a estimativa de esforço seja mais acurada sem aumentar o custo e tempo gasto na técnica.

A abordagem proposta por Ochodek, Nawrocki e Kwarciak (2011) foi realizada baseada num conjunto de catorze projetos de diferentes tecnologias como Delphi, C-sharp, Java e Python e Asp.NET e diferentes domínios como Solução E-Commerce, Sistema de informações Bibliométricas, Sistema bancário de pagamento virtual.

A aplicação da técnica discutiu os seguintes aspectos relevantes na obtenção da estimativa: complexidade dos atores na técnica de Pontos de Caso de Uso, fatores

Tabela 10 – Diferenças entre as abordagens Original e Simplificada

	Pontos de Caso de Uso	Pontos de Caso de Uso Simplificada
Complexidade dos atores	Utiliza	Não utiliza
Passo vs. Transação	Indiferente	Passos
Fatores Técnicos	13 fatores	4 fatores
Fatores Ambientais	8 fatores	2 fatores

de ajuste da técnica, uso de passos e de transações. Tais discussões possibilitaram a adaptação da técnica tradicional e cada mudança será discutida individualmente; as diferenças entre a abordagem original e a abordagem simplificada da técnica de Pontos de Caso de Uso podem ser observadas na tabela 10.

Para cálculo da estimativa da métrica, os casos de uso que descreviam os projetos em questão passaram por revisão, em que alguns foram rejeitados de acordo com normas estabelecidas previamente, sendo descartados na utilização. Uma posterior contagem de passos e transações foi realizada de acordo com o método proposto por Robiolo, Badano e Orosco (2009). Por fim, foram realizadas revisões dos passos e transações e discussão dos resultados para que houvesse convergência nas estimativas finais. Após essas etapas, os fatores de ambiente e de complexidade técnica foram obtidos a partir de uma pesquisa e os Pontos de Caso de Uso finalmente calculados.

2.3.3.1.1 Complexidade dos atores na métrica Pontos de Caso de Uso

Segundo Ochodek, Nawrocki e Kwarciak (2011), os pesos dos atores na métrica de Pontos de Caso de Uso proposta por Karner não foram validados empiricamente e possuem alguns problemas com relação ao cálculo, tais como:

- Generalização dos atores, onde atores que poderiam ser substituídos por um específico são detalhados. Para melhor compreender tal problema, pode-se tomar o exemplo de um sistema onde atuam analistas júniores, sêniores e plenos. A generalização dos atores que implica em somente em 1 ator, o analista, faz com que o caso de uso possua uma menor pontuação. Já a especialização dos atores acarreta numa pontuação maior.
- Adição de atores que são ditados pelas propriedades particulares do projeto mas que poderiam ser substituídos por um ator não pertinente ao domínio mas que possuísse o interesse em comum no resultado do caso de uso. Para uma melhor compreensão desse problema pode-se tomar o exemplo duma rede de postos de

saúde onde os seguintes atores estão presentes no domínio: médico, gerente do posto e paciente. Os três atores são interessados na funcionalidade de exibição de relatório de consultas de paciente. Caso os três atores fossem substituídos por um não pertinente ao domínio mas que caracterizasse melhor o objetivo do caso de uso, o leitor, a complexidade de tal caso de uso seria reduzida e o caso de uso não perderia o seu sentido.

A técnica de simplificação de Pontos de Caso de uso descarta a pontuação associada aos atores para evitar os problemas citados anteriormente. Os dados dos projetos analisados demonstram que não houve alteração significativa na pontuação final da estimativa.

2.3.3.1.2 Fatores de Ajuste

A métrica original de pontos de caso de uso possui treze fatores de complexidade técnica e oito fatores de complexidade de ambiente, no entanto nenhum padrão para interpretação da escala que tais fatores podem assumir foi proposto, sendo que esses podem variar de 0 a 5 conforme tabela 8, mas com grande interferência de quem está aplicando a métrica. Os pesos relativos a tais fatores, contidos nas tabelas 7 e 9 também pouco foram discutidos após a proposta da métrica original.

A análise de dados obtidos dos projetos em questão permitiu observar que apenas quatro fatores de complexidade técnica e dois fatores de ambiente foram relevantes na pontuação final. Tais fatores técnicos com maior impacto final são eficiência, operabilidade, manutibilidade e interoperabilidade e de característica ambientais: experiência da equipe e coesão do time.

2.3.3.1.3 Passos vs. Transações

Uma das principais seções de um documento de caso de uso é a seção de Fluxo de Eventos, que descreve o curso de eventos ou cenário em que as interações entre usuário e sistema são realizadas. A sequência de eventos contida nessa seção usualmente é representada através de passos ou transações.

A diferença existente na pontuação do caso de uso se deve ao fato de passos serem elementos sintáticos e transações elementos semânticos. Um passo é usualmente representado para uma ação do ator ou do sistema em questão, enquanto uma transação pode ser entendida como uma parte do cenário de caso de uso, um passo, ou até mesmo uma simples frase. Karner (1993) define transação como um conjunto de atividades no cenário, executadas inteiramente ou não. Algumas abordagens pregam que a transação é a interação do usuário e a resposta do sistema a

essa interação enquanto outros autores como Jacobson (1992) definem cada passo do caso de uso como uma transação.

Autores como Ribu (2001) não fazem distinção entre passos e transações, diferentemente do que pregam abordagens mais tradicionais. A diferenciação entre passo e transação não é fácil e a utilização de transações ao invés de passos pode influenciar o resultado final da estimativa de maneira significativa, já que a definição de transação oferece margem a múltiplas interpretações.

Para simplificação da técnica o conceito de transação foi descartado e somente o número de passos foi levado em consideração para realização dos cálculos. Tal simplificação deve ser feita somente se uma padronização do nível de abstração e do estilo de escrita do caso de uso seja realizada.

2.3.3.2 Comentários gerais sobre a técnica por seus criadores

O processo de simplificação da métrica de Pontos de Caso de Uso analisou três pontos da técnica que foram propostos inicialmente, mas pouco discutidos ao longo dos anos após sua publicação. A omissão de alguns passos da técnica permitiu que o custo relativo à obtenção da estimativa final fosse reduzido sem que houvesse significativas alterações no resultado.

As principais limitações das observações realizadas neste estudo são relacionadas com o tamanho reduzido do conjunto considerado dos projetos e da sua heterogeneidade.

2.3.4 Pontos de Tamanho de Caso de Uso

Uma terceira abordagem da técnica de Pontos de Caso de Uso foi desenvolvida por Braz e Vergilio (2006) suprindo principalmente a deficiência em se calcular os pontos de caso de uso para grandes sistemas cuja partição não fosse homogênea, onde os casos de uso apresentam complexidade variável.

2.3.4.1 Apresentação da abordagem

A técnica chamada de USP (Use Case Size Point) mede a funcionalidade considerando a estrutura e seções de um caso de uso (BRAZ; VERGILIO, 2006). Ela contempla seções do caso de uso não tratadas pela abordagem original e que, segundo a investigação de seus criadores, afetam o resultado final das estimativas. Quando a técnica é utilizada, um caso de uso pode adquirir complexidade três vezes maior que outro.

Dentre as principais diferenças entre a abordagem e a proposta inicialmente por Kerner destacam-se: a contagem de pontos para pré-condições, pós-condições, ex-

Tabela 11 – Classificação dos Atores na métrica USP

Complexidade	Dados	CA
Simples	≤ 5	2,0
Média	6 a 10	4,0
Complexa	> 10	6,0

Fonte: (BRAZ; VERGILIO, 2006)

ceções e fluxos alternativos, que antes não eram avaliados na pontuação final. De maneira similar à métrica tradicional, a pontuação final do caso de uso é obtida através da multiplicação da pontuação desajustada do caso de uso por valores associados a fatores técnicos e de ambiente. A pontuação desajustada do caso de uso (UUSP) é obtida através da equação 2.13.

$$UUSP = TPA + TPPrC + PCP + TPCA + TPE + TPPoc \quad (2.13)$$

Onde:

- TPA equivale ao somatório da complexidade dos atores de cada caso de uso;
- TPPrC equivale ao somatório da complexidade das pré-condições de cada caso de uso;
- PCP equivale à soma do número de entidades e passos para o cenário principal;
- TPCA equivale ao somatório da complexidade dos cenários alternativos de cada caso de uso;
- TPE equivale ao somatório da complexidade das exceções de cada caso de uso;
- TPPoc equivale ao somatório da complexidade das pós-condições de cada caso de uso;

2.3.4.1.1 Classificação dos Atores

A complexidade total dos atores do caso de uso (TCA), demonstrada na equação 2.14 equivale ao somatório das complexidades individuais (CA) dos atores de caso de uso. Tais complexidades podem ser obtidas de acordo com a tabela 11.

$$TPA = \sum_{i=1}^n CAi \quad (2.14)$$

Tabela 12 – Classificação das pré-condições na métrica USP

Complexidade	Expressões Testadas	CPrC
Simples	1	1,0
Média	2 ou 3	2,0
Complexa	> 3	3,0

Fonte: (BRAZ; VERGILIO, 2006)

Tabela 13 – Classificação dos Cenários na métrica USP

Complexidade	Entidades e Passos	PCP
Muito Simples	≤ 5	4,0
Simples	6 a 10	6,0
Média	11 a 15	8,0
Complexa	16 a 20	12,0
Muito Complexa	> 20	16,0

Fonte: (BRAZ; VERGILIO, 2006)

2.3.4.1.2 Classificação das pré-condições

A complexidade das pré-condições não era contemplada inicialmente pela métrica de pontos de caso de uso. No entanto, pode ser relevante na pontuação final para medir as possíveis situações iniciais em que o sistema deve estar para que o caso de uso seja iniciado.

A complexidade de pré-condições de cada caso de uso (CPrC) é obtida através da quantidade de expressões lógicas, de acordo com a tabela 12. A complexidade total de pré-condições (TPPrC) é obtida pelo somatório das complexidades individuais de acordo com a equação 2.15.

$$TCPrC = \sum_{i=1}^n CPrCi \quad (2.15)$$

2.3.4.1.3 Classificação do cenário principal

A complexidade relativa ao cenário principal (PCP) é obtida a partir da soma do número de entidades e passos de acordo com a tabela 13.

2.3.4.1.4 Classificação dos cenários alternativos

Os cenários alternativos não eram contemplados inicialmente pela métrica proposta de Pontos de Caso de Uso por Kerner. Esses cenários alternativos podem adquirir complexidade muitas vezes maior que a do cenário principal, sendo justificável então a contagem de pontos para eles.

Tabela 14 – Classificação das exceções na métrica USP

Complexidade	Entidades	CE
Simples	≤ 3	1,0
Média	4 a 6	2,0
Complexa	> 6	3,0

Fonte: (BRAZ; VERGILIO, 2006)

Tabela 15 – Classificação das pós-condições na métrica USP

Complexidade	Entidades	CE
Simples	≤ 3	1,0
Média	4 a 6	2,0
Complexa	> 6	3,0

Fonte: (BRAZ; VERGILIO, 2006)

O cálculo total da complexidade dos cenários alternativos (TPCA) pode ser obtido de acordo com a equação 2.16, onde a complexidade de cada cenário alternativo (PCA) é obtido de maneira similar ao cenário principal, de acordo com a tabela 13.

$$TPCA = \sum_{i=1}^n PCA_i \quad (2.16)$$

2.3.4.1.5 Classificação das exceções

O cálculo de complexidade das exceções de caso de uso (CE), também não estava presente na versão inicial da métrica. O peso individual de cada exceção é obtido através do número de expressões lógicas testadas para a obtenção da exceção, de acordo com a tabela 14. O cálculo final da complexidade de exceções (TPE) é obtido através do somatório das complexidades individuais, de acordo com a equação 2.17.

$$TPE = \sum_{i=1}^n PE_i \quad (2.17)$$

2.3.4.1.6 Classificação das pós-condições

O cálculo individual da complexidade das pós-condições (CPoc) é obtido através do número de entidades de cada uma de acordo com a tabela 15. A complexidade total das pós-condições do caso de uso (TCPoc) é dada pelo somatório das complexidades individuais de acordo com a equação 2.18.

$$TCPoC = \sum_{i=1}^n CPoCi \quad (2.18)$$

2.3.4.1.7 Aplicação dos valores de ajuste

O valor de Pontos de Caso de Uso obtido anteriormente ainda precisa passar por um processo de ajuste derivado da métrica de Pontos de Função e Pontos de Caso de Uso, onde fatores técnicos e ambientais são analisados e multiplicados pela pontuação obtida.

2.3.4.2 Comentários pós-aplicação da técnica

Os resultados obtidos para essa métrica foram comparados aos resultados obtidos pelas métricas de Pontos de Caso de Uso e Pontos de Função, por Braz e Vergilio (2006) que considera que a métrica apresentou vantagens quando comparada à UCP tais como a possibilidade de se obter estimativas para cada Caso De Uso separadamente. Os resultados mostrados na avaliação empírica demonstraram ainda que a USP proporcionou melhores estimativas a UCP. Não houve diferenças entre as taxas de erro para FP e USP indicando uma correspondência entre as métricas.

2.3.5 Análise de Pontos de Caso de Uso com adição de novos fatores ambientais

2.3.5.1 Apresentação da abordagem

A última abordagem derivada da técnica de Pontos de Caso de Uso foi apresentada por Alwidian e Hadi (2012) na Arábia Saudita com o objetivo de tornar a técnica aplicável ao ambiente de desenvolvimento local através da adição de novos fatores ambientais. Ela se utilizava da abordagem original da métrica, proposta por Karner (1993), com adição dos fatores ambientais que podem ser encontrados junto a seus respectivos pesos na tabela 16. A adição de tais fatores de ambiente implica na alteração sua fórmula de obtenção, sendo este obtido pela equação 2.19.

$$ECF = C1 + C2 * \sum_{i=1}^{24} Wi * Vi / 3.48 \quad (2.19)$$

Onde,

- ECF: Complexidade de fatores de ambiente
- C1: 1,4
- C2: -0,03
- Wi: Multiplicador de esforço do fator
- Vi: Valor entre 1 e 5 que tal valor tem em importância no caso de uso

Tabela 16 – Fatores Ambientais Adicionais na estimativa por pontos de Caso de Uso

	Fator Sugerido	Multiplicador de Esforço
1	Recursos satisfeitos	0,9
2	Motivação financeira	1,2
3	Descrição de trabalho generalizada	0,9
4	Tipo de cliente	1,2
5	Eventos religiosos	1,2
7	Eventos não agendados	0,9
8	Treinamento contínuo	1,0
9	Parceiro imposto	1,2
10	Rotatividade de trabalho	1,2
11	Tomada de decisão	0,8
12	Aumento de tarefas	1,2
13	Estratégias de TI unificadas	0,9
14	Relação de emprego	0,7
15	Conceito de terminologias	0,8
16	Nível de renda	1,2

Fonte: (ALWIDIAN; HADI, 2012)

2.3.5.2 Comentários gerais pós-aplicação da técnica

A aplicação da técnica num conjunto de 14 projetos foi elaborada de maneira a se comparar o resultado obtido com os resultados obtidos pela técnica tradicional. Usando o modelo sugerido a taxa de erro calculada foi sempre menor ou igual à obtida pela abordagem tradicional, permitindo observar que essa foi adequada ao ambiente em questão.

2.4 Considerações do Capítulo

2.4.1 Síntese geral

O presente capítulo discutiu a importância das métricas estimativa de tamanho de software no mercado de desenvolvimento de sistemas, além das principais técnicas para estimativa de recursos e de tempo como COCOMO, Pontos de Função e Pontos de Caso de Uso. Essas três métricas foram apresentadas e sua metodologia de aplicação e contexto que são melhores aplicadas foram descritas de forma superficial.

A métrica de Pontos de Caso de Uso foi discutida detalhadamente, onde a abordagem original da técnica e outras três variantes foram apresentadas. A métrica originalmente proposta por Karner foi adaptada por autores de maneira a aumentar a precisão, sendo que algumas dessas adaptações tentam simplificá-la sem que seu resultado seja afetado de maneira significativa.

2.4.2 Determinação da abordagem mais complexa e mais simples para estudo de caso

Dentre as abordagens propostas para a determinação de esforço através de Pontos de Caso de uso, a técnica mais simples foi a desenvolvida por Ochodek, Nawrocki e Kwarciak (2011). A técnica de Pontos de Caso de Uso simplificada reduz o número de fatores técnicos e ambientais além de descartar a pontuação associada aos atores do caso de uso. Essa simplificação na técnica reduz o número de atividades para que o esforço associado ao projeto seja obtido, e por esse motivo a técnica se qualifica como a mais simples dentre as abordagens analisadas.

A técnica de Pontos de Tamanho de Caso de Uso, desenvolvida por Braz e Vergilio (2006) é utilizada empiricamente para validação de projetos. Por possuir um maior número de atividades para que a estimativa seja obtida, ela se qualifica como mais complexa por adicionar complexidades associadas a outros elementos do caso de uso, tais como pré-condições, pós-condições, fluxos alternativos e exceções.

Essas duas abordagens são utilizadas num estudo empírico, juntamente com a abordagem original, para avaliar a precisão que conseguem estimar o esforço necessário ao seu desenvolvimento.

3 APLICAÇÃO DAS ABORDAGENS

Esse capítulo descreve uma análise empírica realizada a partir de abordagens da métrica de Pontos de Caso de Uso. As abordagens, uma de maior complexidade (Pontos de Tamanho de Casos de Uso), uma de menor complexidade (Pontos de Caso de Uso Simplificada) e a abordagem original, descritas nas seções 2.3.4, 2.3.3 e 2.3.2, respectivamente, aplicadas a um conjunto de quinze documentos de caso de uso, são avaliadas em atributos de exatidão e complexidade. A exatidão, avaliada pelo quanto as estimativas se aproximaram do tempo real coletado para a implementação dos casos de uso e complexidade, avaliada pelo tempo necessário para que essas estimativas fossem realizadas. A seção 3.1 apresenta os passos necessários para a coleta dos resultados e a seção 3.2 apresenta os resultados obtidos, bem como as discussões sobre tais resultados.

Esse capítulo é dividido em duas seções, onde a primeira apresenta a metodologia para que a análise comparativa seja realizada e a segunda apresenta os resultados obtidos através da metodologia aplicada.

3.1 Metodologia

Essa seção apresenta a metodologia utilizada para que as métricas escolhidas sejam aplicadas de forma satisfatória aos objetos utilizados na comparação entre elas – os quinze casos de uso selecionados. Como nenhuma metodologia já existente foi encontrada na literatura para a comparação entre métricas, cada atividade dessa metodologia possui uma subseção com uma sucinta descrição e uma subseção justificando sua importância para sua aplicação como um todo. De maneira resumida, para que a metodologia fosse aplicada foi necessário: a seleção dos casos de uso, e uma posterior revisão e padronização. Após sua padronização, que consistiu na revisão e estabelecimento de uma maneira comum de se escrever os documentos, a etapa de contagem de atores e de passos foi realizada isoladamente para que os dados pudessem ser utilizados em mais de uma abordagem. Com os resultados obtidos no passo anterior, as abordagens foram aplicadas, isolando-se um conjunto de casos de uso para a descoberta da produtividade da equipe e outro conjunto para a avaliação das métricas com a produtividade já estabelecida. Através da figura 1 pode-se observar as atividades estabelecidas na metodologia, que são detalhadas posteriormente. É importante ressaltar que as atividades de seleção, revisão e padronização dos casos de uso são etapas preparatórias para a aplicação das métricas. Essas etapas preparatórias são importantes para que a aplicação das abordagens sejam realizadas

eficientemente. As etapas posteriores consistuem na aplicação das abordagens em si, ou de passos dessas abordagens.

Os quinze documentos de caso de uso utilizados para essa análise foram extraídos de um conjunto de cinco projetos reais de uma empresa do setor financeiro. Os tempos dispendidos para implementação de tais casos de uso são utilizados para analisar a precisão das métricas em questão, juntamente com os resultados obtidos com a aplicação da abordagem original.

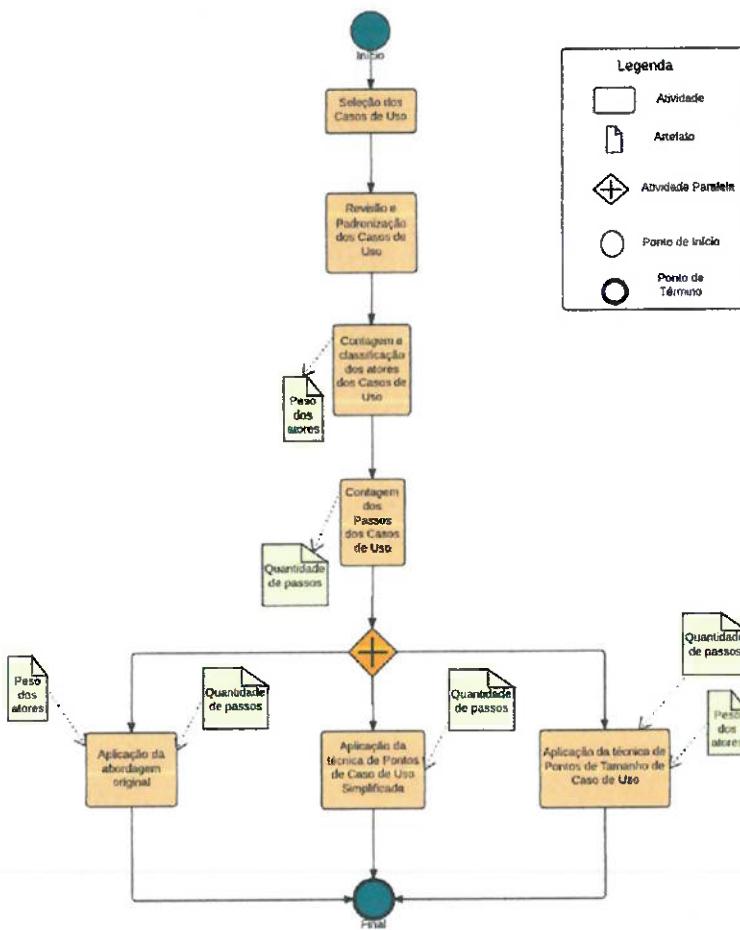


Figura 1 – Fluxo de Atividades para a aplicação da metodologia

3.1.1 Seleção de casos de uso

3.1.1.1 Descrição da etapa

Nessa etapa uma quantidade pré-definida de quinze casos de uso foi escolhida para análise a partir da base histórica de documentos disponíveis na empresa finan-

ceira à qual se tinha acesso. Os critérios para seleção dos documentos foram:

- **Casos de uso pertencentes a diferentes projetos:** a heterogeneidade dos casos de uso é importante para análise do grau de assertividade das métricas independentemente da maneira de se escrever o documento de caso de uso e de quem o escreveu.
- **Casos de uso já implementados:** é importante que o caso de uso já tenha sido implementado para que seja possível a comparação das estimativas realizadas com o esforço de desenvolvimento real registrado.
- **Casos de uso com alto nível de abstração:** uma das vantagens da utilização da métrica de análise de pontos de caso de uso é poder ser aplicada antes que a análise de pontos de função, já que não existe a necessidade de definição de um modelo de dados. O interesse em casos de uso com nível mais alto de abstração, aqueles que ignoram características menos importantes e concentram nos aspectos essenciais do contexto, é justificado por essa necessidade de que os envolvidos no projeto possuem, em estabelecer prazos o mais rápido possível.

3.1.1.2 Justificativa da etapa

A etapa de seleção de casos de uso foi proposta para que possíveis impactos como tempo de desenvolvimento indisponível e disparidade nos tempos de desenvolvimentos e dados viciados não afetassem a precisão da estimativa.

3.1.2 Revisão e padronização dos casos de uso

3.1.2.1 Descrição da etapa

A etapa de revisão e padronização utilizou como base a lista de erros mais comuns em casos de uso, proposta por Rosenberg e Scott (2001), para que uma maneira comum de escrever os casos de uso pudesse ser estabelecida. Tal lista de erros não captura todas as eventuais falhas que possam vir a ocorrer em um documento de caso de uso, mas é suficiente para que exista uma similaridade na escrita, essencial na análise. Os principais pontos possíveis de erros caso de uso são:

- Escrita de requisitos funcionais ao invés de cenários de uso;
- Descrição de atributos e métodos ao invés do uso destes;
- Casos de Uso escritos muito sucintamente;
- Abandono completo da interface de usuário;

- Escrita na voz passiva, utilizando perspectiva diferente da do usuário;
- Foco no que está "dentro" do caso de uso, a explicação de como o sistema responderá a uma ação;

3.1.2.2 Justificativa da etapa

A etapa de padronização foi necessária para que diferenças nas maneiras de se escrever não fossem responsáveis pela perda de precisão da estimativa. Dessa forma uma linguagem comum a todos projetos foi estabelecida e os casos de uso puderam ser utilizados.

3.1.3 Contagem e classificação dos atores

3.1.3.1 Descrição da etapa

A etapa de contagem da pontuação dos atores foi utilizada na abordagem original e na métrica de pontos de tamanho de caso de uso. Nessa etapa foram contados os atores do caso de uso e duas classificações foram realizadas, uma contemplando a abordagem original, conforme a tabela 5 e uma classificação contemplando a abordagem de pontos de tamanho de caso de uso, conforme a tabela 11.

3.1.3.2 Justificativa da etapa

Essa etapa, por ser comum a mais de uma abordagem da métrica (Pontos de Caso de Uso Original e Pontos de Tamanho de Caso de Uso), foi realizada anteriormente à aplicação das abordagens em si. A escolha de transformá-la em uma etapa individual foi motivada pelo fato dos resultados obtidos nessa etapa serem utilizados mais de uma vez.

3.1.4 Contagem dos passos dos casos de uso

3.1.4.1 Descrição da etapa

Para a realização da contagem, as seguintes regras sugeridas por Ribu (2001) foram utilizadas para todas as abordagens:

- Quando o mesmo passo ocorrer em diversos casos de uso, como ex: login ou procedimentos de segurança, o passo deve ser contado apenas uma vez porque a função é implementada uma vez e reutilizada em outros casos de uso;
- Contar os casos de uso estendidos e incluídos separadamente apenas uma vez e não como um passo de cada caso de uso que o utiliza;

- Identificar as classes de análise (não as de projeto) que implementam as funções do caso de uso para auxiliar na determinação da complexidade do caso de uso. As classes podem ser identificadas na descrição dos casos de uso, nos diagramas de sequência ou no diagrama de classes de análise;
- Comparar os passos (descritos no caso de uso) com os passos do diagrama de sequência e com o número de classes de análise utilizadas para implementar o caso de uso para verificar se a complexidade do caso de uso foi definida corretamente;
- Quando a documentação não apresentar a descrição dos casos de uso, deve-se utilizar o diagrama de sequência para contar os passos (apenas aquelas que indicam o que fazer e não as que indicam o como fazer) ou as classes que implementam o caso de uso.

3.1.4.2 Justificativa da etapa

A etapa de contagem de passos dos cenários principais dos casos de uso foi isolada da abordagem tradicional para uma posterior utilização nas três derivadas da métrica original (Simplificada, Pontos de Tamanho de Caso de Uso e Pontos de Caso de Uso Original). A classificação do caso de uso de acordo com os passos que possui não foi realizada nessa etapa pois as abordagens utilizadas possuem maneiras distintas de atribuir pesos a cada um deles.

3.1.5 Aplicação da abordagem original

3.1.5.1 Descrição da etapa

Para que a abordagem original fosse aplicada, a quantidade de passos obtida anteriormente foi classificada de acordo com a tabela 6. Ao valor obtido, adicionou-se a pontuação dos atores, que já foi classificada anteriormente conforme tabela 5, esses valores resultaram nos pontos de caso de uso desajustados, que ainda deveriam sofrer o ajuste baseado em fatores técnicos e ambientais. O ajuste de fatores foi realizado a partir das tabelas 8 e 9 sendo que os valores utilizados para que o ajuste fosse realizado foram encontrados em bases históricas. Para o ajuste baseado em fatores técnicos foi realizada a análise da arquitetura dos sistemas em questão. O fato de todos os sistemas avaliados apresentarem o mesmo modelo arquitetural e de nenhum requisito que afetasse a arquitetura do sistema ter sido explicitado nos documentos avaliados implicou em mesmos fatores de ajuste técnicos para todos os casos de uso. Para aqueles fatores técnicos de complexidade em que não foi possível uma avaliação

ou obtenção dos dados históricos foi atribuído o valor de complexidade 3, cujo valor não interfere no cálculo final.

Os valores associados aos fatores ambientais dos casos de uso foram obtidos por analistas de sistemas participantes do ciclo de desenvolvimento do projeto em questão. As regras utilizadas para a classificação dos fatores, variando de 0 a 5, foram:

- **Familiaridade com o processo de desenvolvimento** - Obtido a partir do tempo que o funcionário atua na empresa.
- **Experiência na aplicação** - Obtido a partir do tempo que o desenvolvedor atua no sistema em questão.
- **Experiência com orientação a objetos** - Obtido com base na experiência prévia do desenvolvedor em orientação a objetos.
- **Experiência do líder do projeto** - Valor obtido com base no tempo que o líder do projeto atua na área.
- **Motivação** - Valor obtido diretamente com líder de projeto, quando este disponível.
- **Estabilidade dos requisitos** - Valor obtido com base na quantidade de versões de documentos de especificação de requisitos que foram motivados por solicitação do cliente.
- **Membros da equipe com dedicação parcial** - Valor obtido com base na alocação do(s) desenvolvedor(es) em outros projetos.
- **Dificuldade da Linguagem de programação** - Valor fixo aplicado para todos casos de uso.

Após a obtenção dos pontos de caso de uso ajustados, um conjunto pré-determinado de cinco casos de uso foi separado para que a produtividade da equipe pudesse ser obtida. Uma relação de produtividade foi obtida para casos de uso de manutenção e outra para casos de uso de novas implementações. Com a produtividade já obtida, outro conjunto de dez casos de uso foi selecionado para que estimativas de tempo pudessem ser estabelecidas e dessa maneira comparar os resultados obtidos com os reais.

3.1.5.2 Justificativa da etapa

Uma etapa contendo a aplicação da abordagem original foi proposta devido à importância que a métrica adquiriu no segmento de software. Como essa métrica serviu de base para que adaptações fossem realizadas, é de grande interesse comparar os resultados obtidos pelas abordagens com o da métrica original.

3.1.6 Aplicação da métrica de Pontos de Caso de Uso Simplificada

3.1.6.1 Descrição da etapa

A aplicação da métrica de Pontos de Uso Simplificada utilizou a classificação de passos obtida na aplicação da abordagem original sem a adição da complexidade de atores. É importante ressaltar que, para todas as abordagens, utilizou-se o conceito de passo ao invés de transação, não existindo então a necessidade de conversão de acordo com a abordagem específica.

Os fatores de ajuste da Análise de Pontos de Caso de Uso Simplificada constituíram-se em apenas quatro fatores técnicos: eficiência, operabilidade, manutecibilidade e interoperabilidade e dois ambientais: experiência da equipe e coesão do time.

Novamente um conjunto de cinco casos de uso foi separado para o estabelecimento da produtividade da equipe, e com essas relações de produtividade obtidas foi-se aplicado em outro conjunto de dez casos de uso para a análise dos resultados.

3.1.6.2 Justificativa da etapa

Uma etapa contendo a metodologia de aplicação da versão simplificada da métrica foi proposta para que os tempos previstos por ela fossem comparados com os tempos efetivos da implementação.

3.1.7 Aplicação da métrica de Pontos de Tamanho de Caso de Uso

3.1.7.1 Descrição da etapa

A aplicação da abordagem utilizou os resultados obtidos na contagem de passos para o cenário principal do caso de uso, sendo estes classificados de acordo com a tabela 13. Os cenários alternativos ainda foram classificados de acordo com tal tabela. A métrica utilizou ainda a pontuação dos atores obtidas anteriormente juntamente com a complexidade das pré-condições, exceções e pós-condições, obtidas através das tabelas 12, 14 e 15 respectivamente.

Para o ajuste dos Pontos de Caso de Uso foram utilizadas as tabelas 8 e 9, de maneira idêntica à abordagem original.

A métrica seguiu os mesmos passos das outras abordagens para que a análise de precisão fosse feita, sendo esses obtenção da produtividade da equipe através de um conjunto de cinco casos de uso e uma posterior aplicação em um outro conjunto de casos de uso para que a análise fosse realizada.

3.1.7.2 Justificativa da etapa

Uma etapa contendo a metodologia de aplicação da métrica de Pontos de Tamanho de Caso de Uso foi proposta para que os tempos previstos por ela fossem comparados com os tempos efetivos da implementação.

3.1.8 Metodologia de análise dos resultados

3.1.8.1 Descrição da etapa

Os resultados obtidos pela aplicação das abordagens foram discutidos de acordo com as seguintes características:

- **Tempo para a aplicação da abordagem:** Esse atributo é responsável por avaliar o tempo dispendido para que a estimativa seja realizada e é medido em horas/homem. É de extrema importância que o tempo gasto para que a estimativa seja feita não seja alto.
- **Precisão da abordagem:** Característica que nos fornece o quanto a estimativa se aproximou do resultado real. É obtido através do comparativo percentual entre a quantidade de horas gastas para implementação do caso de uso e a quantidade de horas estimada. Para esse trabalho, inicialmente será considerada a análise de Pontos de Caso de Uso Original e a partir desta a relação Tempo/ Caso de Uso será obtida, com essa relação será possível avaliar a precisão das distintas abordagens.

3.1.8.2 Justificativa da etapa

Uma etapa contendo a metodologia de análise de resultados foi proposta para que todos os dados necessários à análise fossem coletados previamente e para que houvesse um acordo quanto a maneira que esses dados fossem analisados. Dessa forma, evitou-se uma análise incompleta devido a dados que não pudessem ser obtidos posteriormente.

3.2 Execução da metodologia proposta

3.2.1 Seleção, revisão e padronização dos casos de uso

Essa seção descreve as observações realizadas nas primeiras duas etapas da metodologia, a de Seleção de casos de uso e a de Revisão e Padronização dos casos de uso. Na etapa de seleção de casos de uso foram selecionados dezenove casos de uso de um conjunto de vinte e três casos de uso disponíveis. Esse conjunto foi estabelecido inicialmente pois foram implementados recentemente e as informações sobre fatores ambientais estavam disponíveis com maior facilidade. Os casos de uso descartados não atenderam principalmente ao requisito de pertencerem a diferentes projetos, onde muitos descreviam funcionalidades similares. Na etapa de revisão e padronização foram descartados os quatro casos de uso muito sucintos e que abandonaram completamente a interface com usuário; dos restantes, quatro sofreram alterações no seu conteúdo para atender o requisito de escrita na voz ativa.

3.2.2 Contagem de atores e passos

Nessa subseção são apresentados os resultados de duas etapas da metodologia proposta: a de contagem e classificação dos atores, cujos resultados estão dispostos na tabela 17 e foram utilizados na abordagem de Pontos de Tamanho de Caso de Uso e na abordagem original; e a de classificação dos cenários principais, utilizados na abordagem original, simplificada e Pontos de Tamanho de Caso de Uso. Para fins de análise, todos casos de uso foram classificados como casos de uso de manutenção ou de nova implementação, tal distinção também pode ser observada na tabela.

3.2.3 Aplicação da abordagem original

Para a aplicação da abordagem original, inicialmente a pontuação dos atores e passos, obtida anteriormente, passou pelos ajustes técnicos e ambientais previstos pela métrica, cujos pesos podem ser observados nas tabelas 18 e 19, respectivamente. Esses pesos foram utilizados ainda na aplicação da Análise de Pontos de Tamanho de Caso de Uso.

Onde,

- **F1:** Familiaridade com o processo de desenvolvimento
- **F2:** Experiência na aplicação
- **F3:** Experiência com orientação a objetos
- **F4:** Experiência do líder do projeto

Tabela 17 – Classificação dos atores e cenário principal do caso de uso

Caso de Uso	Categoria	Peso dos atores	Classificação do Cenário Principal	Total
UC01	Nova Implementação	2,0	5,0	7,0
UC02	Nova Implementação	1,0	10,0	11,0
UC03	Nova Implementação	2,0	5,0	7,0
UC04	Manutenção	1,0	5,0	6,0
UC05	Nova Implementação	4,0	30,0	34,0
UC06	Nova Implementação	2,0	10,0	12,0
UC07	Nova Implementação	2,0	10,0	12,0
UC08	Nova Implementação	4,0	15,0	19,0
UC09	Manutenção	2,0	10,0	12,0
UC10	Nova Implementação	1,0	5,0	6,0
UC11	Nova Implementação	1,0	5,0	6,0
UC12	Nova Implementação	6,0	20,0	26,0
UC13	Manutenção	6,0	15,0	21,0
UC14	Manutenção	1,0	5,0	6,0
UC15	Manutenção	1,0	5,0	6,0

Tabela 18 – Aplicação dos Fatores de ajuste técnicos

Fator técnico	Resultado Obtido
Distribuição do sistema	4,0
Desempenho da aplicação	4,0
Eficiência do usuário final	5,0
Complexidade do processamento interno	1,0
Reusabilidade do código em outras aplicações	5,0
Facilidade de instalação	2,0
Usabilidade	3,0
Portabilidade	2,0
Facilidade de modificação	2,0
Concorrência	1,0
Funcionalidades especiais de segurança	2,0
Acesso direto a dispositivos de terceiros	3,0
Treinamento especial aos usuários	3,0

Tabela 19 – Aplicação dos Fatores de ajuste ambientais

Caso de Uso	F1	F2	F3	F4	F5	F6	F7	F8
UC01	5,0	5,0	3,0	4,0	3,0	5,0	1,0	2,0
UC02	4,0	5,0	4,0	5,0	3,0	5,0	1,0	2,0
UC03	5,0	5,0	4,0	5,0	2,0	5,0	1,0	2,0
UC04	4,0	5,0	4,0	5,0	3,0	5,0	1,0	2,0
UC05	4,0	4,0	5,0	3,0	3,0	2,0	2,0	2,0
UC06	5,0	5,0	4,0	4,0	3,0	5,0	1,0	2,0
UC07	5,0	5,0	3,0	4,0	2,0	5,0	1,0	2,0
UC08	5,0	5,0	4,0	5,0	3,0	4,0	1,0	2,0
UC09	5,0	5,0	4,0	5,0	3,0	5,0	2,0	2,0
UC10	5,0	5,0	3,0	4,0	3,0	4,0	1,0	2,0
UC11	5,0	5,0	3,0	4,0	3,0	4,0	1,0	2,0
UC12	5,0	5,0	4,0	5,0	3,0	5,0	1,0	2,0
UC13	4,0	4,0	3,0	4,0	2,0	5,0	1,0	2,0
UC14	3,0	4,0	5,0	3,0	3,0	5,0	1,0	2,0
UC15	4,0	4,0	3,0	4,0	3,0	4,0	2,0	2,0

- **F5:**Motivação
- **F6:**Estabilidade dos requisitos
- **F7:**Membros da equipe com dedicação parcial
- **F8:**Dificuldade da Linguagem de programação

Os valores de pontos de caso de uso, antes e depois do ajuste ser realizado, podem ser observados na tabela 20 juntamente com o tempo de implementação real coletado na instituição.

Para a obtenção da produtividade da equipe, foram selecionados aleatoriamente os casos de uso UC01, UC04, UC06, UC09 e UC10. Através da equação 3.1, foi estabelecida a produtividade da equipe para os casos de uso em questão, cujos valores podem ser observados na tabela 21.

$$\text{Produtividade} = \text{Tempo}/\text{UCP} \quad (3.1)$$

Calculando-se a média aritmética das produtividades encontradas, obteve-se os valores de:

- 5h06 por ponto de caso de uso, para casos de uso de nova implementação;
- 1h40 por ponto de caso de uso, para casos de uso de manutenção;

Tabela 20 – Ajuste de fatores - Abordagem Original

Caso de Uso	Categoria	Pontos de Caso de Uso Desajustados	Pontos de Caso de Uso Ajustados	Tempo Implementação
UC01	Nova Implementação	7,0	4,57	31h40
UC02	Nova Implementação	11,0	7,19	42h14
UC03	Nova Implementação	7,0	4,47	15h21
UC04	Manutenção	6,0	3,92	06h14
UC05	Nova Implementação	34,0	29,90	192h43
UC06	Nova Implementação	12,0	7,48	37h12
UC07	Nova Implementação	12,0	8,20	30h43
UC08	Nova Implementação	19,0	12,70	55h10
UC09	Manutenção	12,0	7,66	13h35
UC10	Nova Implementação	6,0	4,28	14h38
UC11	Nova Implementação	6,0	4,28	15h36
UC12	Nova Implementação	26,0	15,81	88h15
UC13	Manutenção	21,0	15,62	46h33
UC14	Manutenção	6,0	4,28	6h14
UC15	Manutenção	6,0	4,82	08h09

Tabela 21 – Obtenção da Produtividade - Abordagem Original

Caso de Uso	Categoria	Pontos de Caso de Uso	Tempo Implementação	Produtividade da Equipe
UC01	Nova Implementação	4,57	31h40	6h55
UC04	Manutenção	3,92	6h14	1h35
UC06	Nova Implementação	7,48	37h12	4h58
UC09	Manutenção	7,66	13h35	1h46
UC10	Nova Implementação	4,28	14h38	3h25

Tais valores foram multiplicados pelos pontos de caso de uso ajustados de outro conjunto de projetos, e a partir do tempo real de desenvolvimento o erro relativo das estimativas foi calculado, tais resultados estão dispostos na tabela 22.

O tempo médio para levantamento da estimativa baseada em pontos de caso de uso foi de aproximadamente vinte e três minutos e o erro médio obtido foi de aproximadamente 24%.

3.2.4 Aplicação da métrica de Pontos de Caso de Uso Simplificada

Para a análise de resultados da métrica de Pontos de Caso de Uso Simplificada foi necessária somente a aplicação dos fatores de ajuste adaptados na pontuação dos passos obtida previamente, já que ela despreza a pontuação associada aos atores do caso de uso. A tabela 23 contempla o ajuste de fatores técnicos e ambientais para a

Tabela 22 – Cálculo das estimativas - Abordagem Original

Caso de Uso	Pontos de Caso de Uso Ajustados	Tempo Efetivo	Tempo Previsto	Erro Relativo (%)
UC02	7,19	42h14	36h39	13,26
UC03	4,47	15h21	22h47	48,62
UC05	29,90	192h43	152h29	20,88
UC07	8,20	30h43	41h49	36,15
UC08	12,70	55h10	64h46	17,38
UC11	4,28	15h36	21h50	39,97
UC12	15,81	88h15	80h37	8,64
UC13	15,62	46h33	26h02	44,07
UC14	4,28	6h14	7h08	14,56
UC15	4,82	8h09	8h02	1,33

Tabela 23 – Ajuste de fatores - Pontos de Caso de Uso Simplificada

Caso de Uso	Categoria	Pontos de Caso de Uso Desajustados	Pontos de Caso de Uso Ajustados	Tempo Efetivo
UC01	Nova Implementação	5,0	4,49	31h40
UC02	Nova Implementação	10,0	8,98	42h14
UC03	Nova Implementação	5,0	4,18	15h21
UC04	Manutenção	5,0	4,18	06h14
UC05	Nova Implementação	30,0	30,60	192h43
UC06	Nova Implementação	10,0	8,98	37h12
UC07	Nova Implementação	10,0	8,98	30h43
UC08	Nova Implementação	15,0	12,55	55h10
UC09	Manutenção	10,0	8,36	13h35
UC10	Nova Implementação	5,0	4,18	14h38
UC11	Nova Implementação	5,0	4,18	15h36
UC12	Nova Implementação	20,0	17,95	88h15
UC13	Manutenção	15,0	15,76	46h33
UC14	Manutenção	5,0	4,18	6h14
UC15	Manutenção	5,0	4,49	08h09

abordagem, juntamente com o tempo efetivo para a sua implementação.

De maneira similar à abordagem tradicional, um conjunto de cinco casos de uso foi isolado para o estabelecimento da produtividade da equipe, e calculado conforme equação 3.1. Os resultados obtidos estão contidos na tabela 24.

Por meio do cálculo da média aritmética foram obtidos os valores de produtividade de:

- 4h54 por ponto de caso de uso, para casos de uso de nova implementação;
- 1h34 por ponto de caso de uso, para casos de uso de manutenção;

Tabela 24 – Obtenção da Produtividade - Pontos de Caso de Uso Simplificada

Caso de Uso	Categoria	Pontos de Caso de Uso	Tempo Implementação	Produtividade da Equipe
UC01	Nova Implementação	4,49	31h40	7h03
UC04	Manutenção	4,18	6h14	1h29
UC06	Nova Implementação	8,98	37h12	4h09
UC09	Manutenção	8,36	13h35	1h37
UC10	Nova Implementação	4,18	14h38	3h30

Tabela 25 – Cálculo das estimativas - Pontos de Caso de Uso Simplificada

Caso de Uso	Pontos de Caso de Uso Ajustados	Tempo Efetivo	Tempo Previsto	Erro Relativo (%)
UC02	8,98	42h14	43h59	4,10
UC03	4,18	15h21	20h29	33,66
UC05	30,60	192h43	149h56	22,20
UC07	8,98	30h43	43h59	43,17
UC08	12,55	55h10	61h28	11,42
UC11	4,18	15h36	20h29	31,35
UC12	17,95	88h15	87h58	0,33
UC13	15,76	46h33	24h32	47,30
UC14	4,18	6h14	6h31	4,49
UC15	4,49	8h09	6h59	14,28

Aplicando-se os valores de produtividade em outro conjunto de casos de uso foi possível realizar as estimativas de tempo dos projetos, e comparar os resultados com os tempos efetivos, tais resultados estão dispostos na tabela 25.

O tempo gasto para levantamento da estimativa foi em média de dezoito minutos e a média de erro obtido foi de aproximadamente 24%, o mesmo obtido na aplicação da abordagem original.

3.2.5 Aplicação da métrica de Pontos de Tamanho de Caso de Uso

A aplicação da abordagem de Pontos de Tamanho de Caso de Uso foi realizada em sete etapas, para cada caso de uso em questão:

1. Obtenção da complexidade dos atores
2. Obtenção da complexidade do cenário principal
3. Obtenção da complexidade dos cenários alternativos
4. Obtenção da complexidade das exceções
5. Obtenção da complexidade das pré-condições

Tabela 26 – Obtenção dos Pontos de Caso de Uso Desajustados - Pontos de Tamanho de Caso de Uso

Caso de Uso	Categoria	TPA	PCP	TPCA	TPPrC	TTPoc	TPE
UC01	Nova Implementação	2,0	4,0	0,0	1,0	1,0	1,0
UC02	Nova Implementação	2,0	6,0	0,0	1,0	1,0	1,0
UC03	Nova Implementação	2,0	4,0	0,0	1,0	0,0	0,0
UC04	Manutenção	2,0	4,0	0,0	1,0	0,0	1,0
UC05	Nova Implementação	2,0	12,0	0,0	4,0	0,0	1,0
UC06	Nova Implementação	2,0	4,0	4,0	2,0	1,0	1,0
UC07	Nova Implementação	2,0	6,0	0,0	2,0	0,0	0,0
UC08	Nova Implementação	2,0	8,0	0,0	4,0	0,0	2,0
UC09	Manutenção	2,0	6,0	0,0	2,0	0,0	1,0
UC10	Nova Implementação	2,0	4,0	0,0	1,0	0,0	0,0
UC11	Nova Implementação	2,0	4,0	0,0	0,0	0,0	1,0
UC12	Nova Implementação	2,0	16,0	0,0	6,0	6,0	2,0
UC13	Manutenção	2,0	12,0	4,0	2,0	3,0	3,0
UC14	Manutenção	2,0	4,0	0,0	1,0	1,0	0,0
UC15	Manutenção	2,0	4,0	4,0	1,0	0,0	0,0

6. Obtenção da complexidade das pós-condições

7. Ajuste dos pontos de caso de uso com base em ajustes técnicos e ambientais

Os resultados coletados nas seis primeiras etapas da análise estão dispostos na tabela 26.

Após a obtenção dos pontos de caso de uso desajustados foi realizado o ajuste com base nos fatores técnicos e ambientais, no qual utilizou-se a mesma classificação que a usada na abordagem original. Os resultados podem ser observados na tabela 27.

Através do conjunto de cinco casos de uso estabelecido na etapa de análise da abordagem original, foi possível a obtenção da produtividade da equipe, que pode ser observada na tabela 28.

Os valores médios de produtividade, obtidos por meio da tabela 28, foram de: Por meio do cálculo da média aritmética foram obtidos os valores de produtividade de:

- 3h38 por ponto de caso de uso, para casos de uso de nova implementação;
- 1h20 por ponto de caso de uso, para casos de uso de manutenção;

A tabela 29 mostra os resultados obtidos, quando as relações de produtividade calculadas anteriormente foram aplicadas em outro conjunto de projetos.

O tempo médio para obtenção da estimativa foi de 32 minutos e o erro médio da abordagem de 31%.

Tabela 27 – Ajuste de Fatores - Pontos de Tamanho de Caso de Uso

Caso de Uso	Categoria	Pontos de Caso de Uso Desajustados	Pontos de Caso de Uso Ajustados	Tempo Efetivo
UC01	Nova Implementação	9,0	5,88	31h40
UC02	Nova Implementação	11,0	7,19	42h14
UC03	Nova Implementação	7,0	4,47	15h21
UC04	Manutenção	8,0	5,23	6h14
UC05	Nova Implementação	18,0	15,83	192h43
UC06	Nova Implementação	14,0	8,72	37h12
UC07	Nova Implementação	10,0	6,83	30h43
UC08	Nova Implementação	16,0	10,69	55h10
UC09	Manutenção	11,0	7,02	13h35
UC10	Nova Implementação	7,0	4,99	14h38
UC11	Nova Implementação	9,0	6,42	15h36
UC12	Nova Implementação	22,0	13,38	88h15
UC13	Manutenção	13,0	9,67	46h33
UC14	Manutenção	9,0	6,42	06h14
UC15	Manutenção	11,0	8,84	08h09

Tabela 28 – Obtenção da Produtividade - Pontos de Tamanho de Caso de Uso

Caso de Uso	Categoria	Pontos de Caso de Uso	Tempo Implementação	Produtividade da Equipe
UC01	Nova Implementação	6,80	31h40	4h40
UC04	Manutenção	6,04	6h14	1h02
UC06	Nova Implementação	10,43	37h12	3h34
UC09	Manutenção	8,25	13h35	1h39
UC10	Nova Implementação	5,43	14h38	2h42

Tabela 29 – Cálculo das estimativas - Pontos de Tamanho de Caso de Uso

Caso de Uso	Pontos de Caso de Uso Ajustados	Tempo Efetivo	Tempo Previsto	Erro Relativo (%)
UC02	8,31	42h14	30h15	28,42
UC03	5,25	15h21	19h07	24,71
UC05	19,09	192h43	69h31	63,93
UC07	7,65	30h43	27h51	9,32
UC08	12,16	55h10	44h17	19,74
UC11	5,43	15h36	19h45	26,63
UC12	20,72	88h15	75h27	14,50
UC13	20,41	46h33	27h19	41,30
UC14	6,20	6h14	8h18	33,23
UC15	8,86	8h09	11h51	45,46

Tabela 30 – Comparativo de tempo estimado nas abordagens avaliadas e o tempo efetivo

Caso de Uso	Tempo estimado - Pontos de Caso de Uso Original	Tempo estimado - Pontos de Caso de Uso Simplificada	Tempo estimado - Pontos de Tamanho de Caso de Uso	Tempo Efetivo para a implementação dos casos de uso
UC02	36h39	43h59	30h15	42h15
UC03	22h47	20h29	19h7	15h20
UC05	152h29	149h56	69h31	192h43
UC07	41h49	43h59	27h51	30h43
UC08	64h46	61h28	44h17	55h10
UC11	21h50	20h29	19h45	15h36
UC12	80h37	87h58	75h27	88h15
UC13	26h02	24h32	27h19	46h33
UC14	7h08	6h31	8h18	6h14
UC15	8h02	6h59	11h51	8h09

3.2.6 Discussão dos resultados

Os resultados obtidos com a aplicação das métricas foram avaliados com base no tempo utilizado para que a estimativa fosse feita e com base na precisão em que chegaram próximas do resultado original. Através da tabela 30 pode-se observar os tempos estimados para as métricas e o quanto se diferem do valor real. Os resultados foram plotados num gráfico, observado na figura 2.

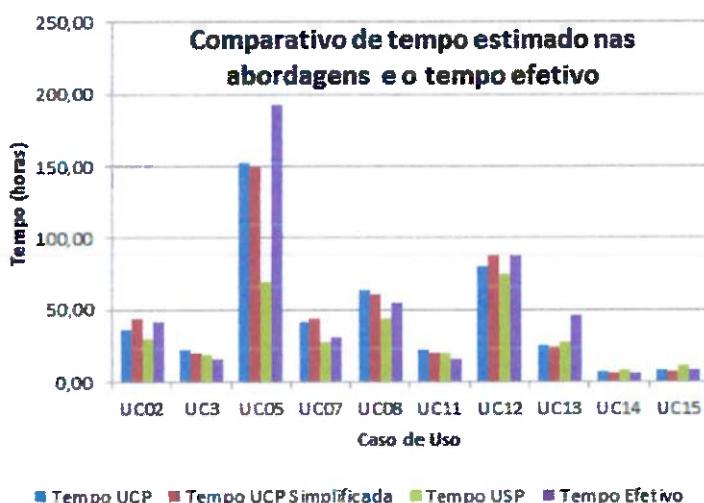


Figura 2 – Comparativo de tempo entre as abordagens estudadas

Os erros relativos encontrados na aplicação das abordagens estão dispostos na tabela 31 e a figura 3 representa os dados de forma gráfica para uma melhor observação.

Tabela 31 – Erro Relativo das abordagens avaliadas

Caso de Uso	Pontos de Caso de Uso (%)	Pontos de Caso de Uso Simplificada (%)	Pontos de Tamanho de Caso de Uso (%)
UC02	13,26	4,10	28,42
UC03	48,62	33,66	24,71
UC05	20,88	22,20	63,93
UC07	36,15	43,17	9,32
UC08	17,38	11,42	19,74
UC11	39,97	31,35	26,63
UC12	8,64	0,33	14,50
UC13	44,07	47,30	41,30
UC14	14,56	4,49	33,23
UC15	1,33	14,28	45,46

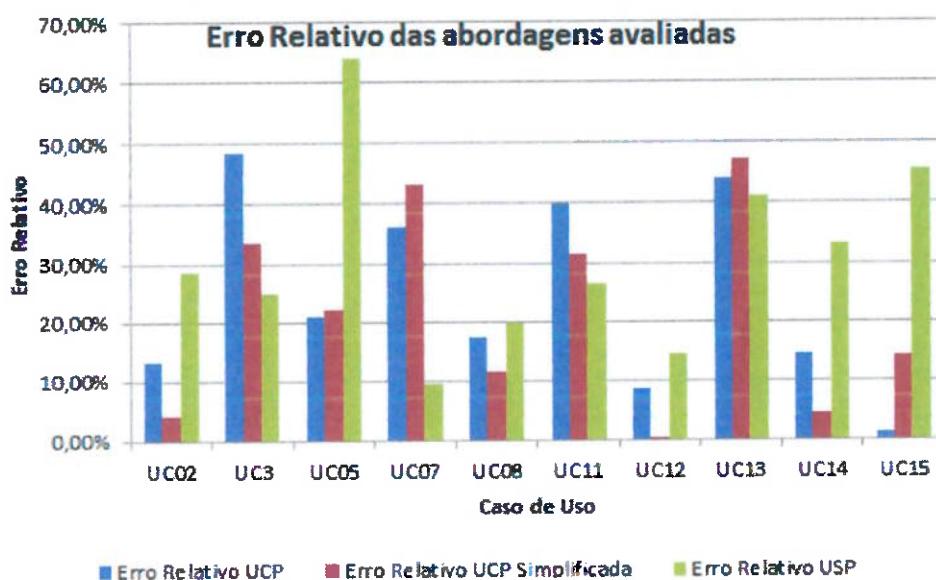


Figura 3 – Erro relativo das abordagens estudadas

A análise dos gráficos e dados coletados permitiu destacar a abordagem simplificada como a que possui o melhor índice de precisão com o menor custo para o levantamento da estimativa. Essa relação era esperada quando comparada à abordagem original, já que os estudos anteriores demonstraram que a abordagem obteve resultados semelhantes e que descartou alguns fatores para o levantamento da estimativa. Um estudo detalhado é realizado acerca dos resultados obtidos no capítulo 4 desse trabalho.

3.3 Considerações do Capítulo

Neste capítulo, foram descritas a metodologia para aplicação das abordagens da métrica de Pontos de Caso de Uso e os resultados obtidos através da execução desta

metodologia. Os resultados encontrados nesse capítulo foram analisados e as considerações acerca desses resultados estão dispostas no capítulo 4, juntamente com as conclusões e perspectivas de novos trabalhos que essa análise possibilitou observar.

4 CONSIDERAÇÕES FINAIS

4.1 Conclusões

O gerenciamento de tempo e custo são disciplinas essenciais para uma boa gestão de projeto. Essas disciplinas estão intimamente relacionadas, já que os custos para o desenvolvimento de software dependem de informações de recursos alocados e tempo que tais recursos necessitam para o desenvolvimento.

A disciplina de Engenharia de Software, relativamente recente se comparada a engenharias tradicionais como civil, mecânica, tem se preocupado em fornecer meios de controlar os custos e tempo do projeto. Técnicas de estimativa vêm sendo utilizadas como auxílio a engenheiros de software e gerentes de projeto para que o tamanho do software possa ser estimado e consequentemente o tempo para sua construção. Algumas técnicas como as métricas de Análise de Pontos de Função, COCOMO e Pontos de Caso de Uso adquiriram notoriedade para a estimativa de tamanho de softwares e são utilizadas com frequência no segmento de Tecnologia da Informação. Uma dessas técnicas, a estimativa por Ponto de Caso de Uso se adequa principalmente a sistemas desenvolvidos utilizando o paradigma de desenvolvimento orientado a objetos. Essa técnica apresenta vantagens na utilização como por exemplo sua facilidade para automatização e aplicação no início do projeto. Por esse motivo foi escolhida como tema central desse trabalho. No entanto, uma das principais deficiências da técnica consiste na falta de padronização existente na escrita dos documentos de caso de uso. Devido a essa possibilidade de diferentes escritas desse tipo de documento, algumas distintas abordagens da técnica de Pontos de Caso de Uso se tornaram relevantes e se demonstraram adequadas a determinadas características do caso de uso.

A partir de uma revisão literária foi possível o levantamento de vantagens e desvantagens na utilização de algumas técnicas tradicionais de estimativa de tempo de software, como COCOMO, Análise de Pontos de Função e Análise de Pontos de Caso de Uso. Essa revisão apontou a precocidade com que a estimativa pode ser realizada como principal vantagem e a falta de padronização de documentos de caso de uso como a principal desvantagem para a técnica de Análise de Pontos de Caso de Uso. Partindo desse princípio da falta de padronização na escrita de casos de uso, foi realizado um aprofundamento no estudo da técnica e identificou-se e se avaliou diferentes maneiras de aplicá-la, que se constituem em abordagens distintas. As abordagens estudadas nesse trabalho foram Análise de Pontos de Caso de Uso Original, Análise de Pontos de Caso de Uso Simplificada, Análise de Pontos de Tamanho de Caso de Uso e Análise de Pontos de Caso de Uso com Adição de Fatores Ambientais.

O estudo literário permitiu observar que a técnica de Análise de Pontos de Caso de Uso Simplificada foi a abordagem de menor complexidade dentre as avaliadas e que a técnica de Análise de Pontos de Tamanho de Caso de Uso foi a abordagem de maior complexidade. Elas foram escolhidas então para uma análise empírica. Na primeira seção desse capítulo pode-se observar como esse estudo pôde colaborar na gestão de projetos através da escolha de uma métrica de estimativa adequada. Uma discussão acerca dos resultados obtidos é realizada para que tal objetivo seja atendido.

Algumas perspectivas de melhorias e projeções de novos trabalhos observadas no decorrer desse estudo são apresentadas na segunda seção do capítulo.

4.2 Contribuições do trabalho

O estudo empírico realizado a partir dos casos de uso da instituição apontou a métrica de Pontos de Caso de Uso Simplificada como a mais eficiente dentre as avaliadas, sendo que essa eficiência foi medida através da relação de erro que a métrica obteve e o tempo para elaboração da estimativa. Tal resultado era esperado na comparação com a abordagem original da métrica, já que a abordagem simplificada descarta o levantamento de alguns fatores, reduzindo o tempo para elaboração, e segundo os estudos originais apresenta o mesmo índice de erros.

O erro relativo encontrado para a métrica de Pontos de Tamanho de Caso de Uso foi inesperado quando comparado à abordagem original, já que a métrica é uma melhoria desta e os resultados encontrados pelos autores da derivada foram melhores que os obtidos a partir da aplicação da original. Esse índice de erro maior pode ser justificado pela pouca variação nos tempos de implementação dos casos de uso. A técnica adaptada utiliza uma nova abordagem para a contagem de passos dos cenários, sendo que esta apresenta uma variação maior no resultado final, conforme a quantidade de passos aumenta. No entanto, os casos de uso utilizados nesse trabalho apresentaram tempos com uma variação pequena, conforme a quantidade de passos aumentou. Tal fator favoreceu as abordagens original e simplificada, que possuem uma tabela de classificação de passos mais simples que a abordagem de Pontos de Tamanho de Caso de Uso.

O erro relativo médio encontrado na aplicação da técnica de Pontos de Tamanho de Caso de Uso por Braz e Vergilio (2006) foi de 30,17%, enquanto o obtido nesse trabalho foi de 31%, resultado extremamente satisfatório. Para a abordagem simplificada, o erro relativo obtido pelos autores não estava disponível. A limitação na quantidade de casos de usos disponíveis impossibilitou uma análise estatística mais aprofundada, no entanto, permitiu uma análise individual dos casos de uso quando necessário. Através

dessa análise constatou-se que 2 casos de uso foram responsáveis pelo aumento de 7% de erro na métrica de Pontos de Tamanho de Caso de Uso, e esses erros foram causados devido à nova classificação de passos proposta na técnica, que para os casos de uso em questão, não colaboraram para o aumento de precisão.

A análise realizada através dos gráficos permitiu observar que as três técnicas obtiveram um índice de erro próximo para casos de uso de manutenção e de novas implementações, caracterizando as três abordagens como próprias para ambos ambientes.

4.3 Trabalhos Futuros

Uma das principais deficiências encontradas no decorrer desse trabalho foi a pequena quantidade de casos de uso disponíveis para análise. Essa pequena quantidade e principalmente a similaridade entre os casos de uso, por serem oriundos de uma mesma instituição, acarretam em dados viciosos e consequentemente numa análise não genérica. Uma forma pela qual o trabalho conseguiria melhores resultados seria através da coleta de dados em instituições de diferentes segmentos de mercado e de uma análise de um maior número de casos de uso.

A análise de uma grande quantidade de casos de uso inviabilizaria a análise manual dos casos de uso para a obtenção dos resultados desajustados, já que essa etapa é de fácil automatização. O estudo para criação de um sistema para automatização do processo de obtenção dos Pontos de Caso De Uso Desajustados seria de grande importância para a minimização do esforço para o processo como um todo. Atualmente já existem ferramentas que automatizem a estimativa baseada em Pontos de Caso de Uso, porém só a abordagem tradicional descrita por Karner é utilizada nessas ferramentas. Com a criação de ferramentas que auxiliem no cálculo para abordagens mais simples e complexas, os envolvidos na no cálculo da estimativa do tamanho do software poderiam optar entre essas abordagens de acordo com suas necessidades.

REFERÊNCIAS

20926:2009, A. S. I. Software and systems engineering - software measurement - ifpug functional size measurement method 2009. In: . [S.I.]: ASTM International, 2009.

AGUIAR, M. Function points or use case points? *IFPUG Metric Views*, Junho 2003.

ALBRECHT, A.; GAFFNEY, J. E. Software function, source lines of code, and development effort prediction: A software science validation. *Software Engineering, IEEE Transactions on*, SE-9, n. 6, p. 639–648, Nov 1983.

ALWIDIAN, J.; HADI, W. Enhancing the results of ucp in cost estimation using new external environmental factors. *International Conference on Information Technology and e-Services*, 2012.

ANDA, B.; DREIEM, H.; SJOBERG, D. I. Estimating software development based on use cases - experience from industry. p. 487–504, Outubro 2001.

BRAZ, M. R.; VERGILIO, S. R. Software effort estimation based on use cases. *Computer Software and Applications Conference, 2006. COMPSAC '06. 30th Annual International*, p. 221–228, Setembro 2006.

BUNDSCUH, M.; DEKKERS, C. *The IT Measurement Compendium: Estimating and Benchmarking Success with Functional Size Measurement*. 1. ed. [S.I.]: Springer Publishing Company, Incorporated, 2008. ISBN 3540681876, 9783540681878.

FENTON, N. E.; PFLEIDERER, S. L. *Software Metrics: A Rigorous and Practical Approach, Revised*. 2a.. ed. [S.I.]: Boston EUA: PWS Publishing Company, 1997.

FOWLER, M. *UML Essencial: Um Breve Guia para Linguagem Padrao*. 3a.. ed. [S.I.]: Bookman, 2005.

GOODMAN, P. *SOFTWARE METRICS Best Practices for Successful IT Management*. 1a.. ed. [S.I.]: Brookfield, 2004.

GOPU, A. Software metrics & associated issues. a literature survey. *Indiana University*, 2003.

HEEMSTRA, F. J. Software cost estimation. *Information and software technology*, Elsevier, v. 34, n. 10, p. 627–639, 1992.

JACOBSON, I. *Object Oriented Software Engineering, A Use Case Driven Approach*. 1a.. ed. [S.I.]: Addison-Wesley Professional, 1992.

JUNIOR, W. T.; SANCHES, R. *Modelos de Estimativas de Custo de Software COCOMO and COCOMO II*. Dissertação (Mestrado) — Universidade de São Paulo, São Carlos, Abril 2006.

KARNER, G. Resource estimation for objectory projects. *Objective Systems SF AB*, Setembro 1993.

LOKAN, C. J. Function points. *Advances in Computers*, 2005.

MERLO, N.; SCHETT. Cocomo, constructive cost model. *Seminar on Software Cost Estimation*, 2003.

MISHRA, S.; HAZRA, K.; MALL, R. A survey of metrics for software development effort estimation. *International Journal of Research and Reviews in Computer Science*, p. 1199–1204, Outubro 2011.

MUKHOPADHYAY, T.; VICINANZA, S. S.; PRIETULA, M. J. Examining the feasibility of a case-based reasoning model for software effort estimation. *MIS quarterly*, JSTOR, p. 155–171, 1992.

OCHODEK, M.; NAWROCKI, J.; KWARCIAK, K. Simplifying effort estimation based on use case points. *Information and Software Technology Journal*, p. 200–213, 2011.

PRESSMAN, R. S. *Software Engineering: A Practitioner's Approach*. 6a.. ed. [S.I.]: McGraw-Hill, Inc, 2005.

RIBU, K. Estimating object-oriented software projects with use cases. Master's thesis, University of Oslo, Department of Informatics, 2001.

ROBIOLO, G.; BADANO, C.; OROSCO, R. Transactions and paths: two use case based metrics which improve the early effort estimation. *Third International Symposium on Empirical Software Engineering and Measurement*, p. 422–425, 2009.

ROLLO, A. L. Functional size measurement and cocomo – a synergistic approach. *Proc. of Software Measurement European Forum*, p. 259–267, 2006.

ROSENBERG, D.; SCOTT, K. Top ten use case mistakes. *SOFTWARE DEVELOPMENT-SAN FRANCISCO-*, v. 9, n. 2, p. 52–56, 2001.

SCHNEIDER, G.; WINTERS, J. P. *Applying Use Cases – A Practical Guide*. 2a.. ed. [S.I.]: Addison-Wesley, 1998.

SCHOFIELD, J.; ARMEMTROUT, A.; TRUJILLO, R. Function points, use case points, story points - observations from a case study. *CrossTalk – the Journal of Defense Software Engineering*, p. 23–27, 2013.

SINGH, G.; SINGH, D.; SINGH, V. A study of software metrics. *International Journal of Computational Engineering and Management*, p. 23–27, Janeiro 2011.

SYAVASYA, C. An approach to find maintenance costs using cost drivers of cocomo intermediate model. *International Journal Of Computational Engineering Research*, Março 2011.

TRAINNING EDUCATION SERVICES. APF - *Análise de Pontos de Função*. [S.I.], 2013.