

UNIVERSIDADE PAULISTA

**PROCESSO DE ESTIMATIVA DE SOFTWARE COM A MÉTRICA *USE*
CASE POINTS, PMBOK E RUP**

EVERTON CASTELÃO TETILA

Dissertação apresentada ao Programa de Pós Graduação em Engenharia de Produção da Universidade Paulista, para obtenção do título de Mestre.

SÃO PAULO
2007

UNIVERSIDADE PAULISTA

**PROCESSO DE ESTIMATIVA DE SOFTWARE COM A MÉTRICA *USE*
CASE POINTS, PMBOK E RUP**

EVERTON CASTELÃO TETILA

Orientador: Prof. Dr. Ivanir Costa

Área de Concentração: Engenharia de
Produção

Dissertação apresentada ao Programa de
Pós Graduação em Engenharia de
Produção da Universidade Paulista, para
obtenção do título de Mestre.

SÃO PAULO
2007

TETILA, Everton Castelão.

Processo de estimativa de software com a métrica *use case points*, PMBOK e RUP./

Everton Castelão Tetila. São Paulo, 2007.

139 p.

Dissertação (Mestrado) – Universidade Paulista, 2007.

Área de Concentração: Engenharia de Produção.

Orientador: Prof. Dr. Ivanir Costa.

1. Gerência de Projetos.
2. Métricas de Software.
3. Processo de Estimativa de Software.

Dedicatória

Aos meus pais, nos quais se doaram inteiros e renunciaram aos seus sonhos, para que, muitas vezes, pudessem realizar os meus.

Agradecimentos

Ao Prof. Dr. Ivanir Costa, que é fonte contínua de inspiração e apoio moral à pesquisa de excelência.

Ao Prof. Dr. Mauro de Mesquita Espínola, cuja sábia orientação deu forma ao meu conhecimento dos problemas computacionais.

Ao Prof. Dr. Cesar Augusto Cardoso Caetano e ao Prof. Dr. Antonio Roberto Albuquerque que fizeram sugestões e contribuições indispensáveis.

As revisoras que garantiram maior rigor técnico ao trabalho, eu devo agradecimentos especiais a Josiane Queiroz da Silva e a minha noiva Juliana Queiroz da Silva, por terem cuidado com os detalhes.

Epígrafe

“De tudo ficaram três coisas: a certeza de que estava sempre começando, a certeza de que era preciso continuar e a certeza de que seria interrompido antes de terminar. Fazer da interrupção um caminho novo. Fazer da queda, um passo de dança, do medo, uma escada, do sonho, uma ponte, da procura, um encontro”.

FERNANDO PESSOA

Índice

Resumo.....	VIII
Abstract.....	IX
Lista de Abreviaturas	X
Lista de Ilustrações.....	XI
Lista de Quadros	XII
Lista de Tabelas.....	XIII
Lista de Anexos.....	XIV
Lista de Apêndices	XV
Capítulo 1 – Introdução	16
1.1 Motivação e Relevância do Problema.....	16
1.2 Objetivos e Suposições	19
1.2.1 Objetivo geral.....	19
1.2.2 Objetivos específicos.....	20
1.2.3 Hipóteses	20
1.3 Justificativa	20
1.4 Metodologia	21
1.5 Estrutura	23
Capítulo 2 – Contribuições da Engenharia de Software	25
2.1 Introdução	25
2.2 Gerência de Projetos (PMBOK).....	25
2.2.1 Introdução.....	25
2.2.2 Projeto	26
2.2.3 Gerência de Projetos.....	28
2.3 Processo de Desenvolvimento de Software (RUP).....	34
2.3.1 Rational Unified Process (RUP).....	34
2.3.2 Disciplinas do RUP	35
2.3.3 Fases do RUP.....	39
2.4 Considerações Finais do Capítulo.....	42
Capítulo 3 – Métricas de Software.....	44
3.1 Introdução	44
3.2 Medição de Software	45
3.3 Métricas de Estimativa de Tamanho de Software.....	47
3.3.1 Linhas de Código – LOC	48
3.3.2 <i>Function Points</i> – FP	50
3.3.3 <i>Use Case Points</i> – UCP	52
3.4 Considerações Finais do Capítulo.....	62
Capítulo 4 – Processo de Estimativa de Software	64
4.1 Introdução	64
4.2 Mapeamento dos processos de gestão de estimativas propostos pelo PMBOK	

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Índice

para as atividades de desenvolvimento de estimativa propostas pelo RUP	64
4.3 Processo de Estimativa de software	67
4.4 Considerações Finais do Capítulo.....	81
Capítulo 5 – Pesquisa-ação	83
5.1 Introdução	83
5.2 Etapas da Pesquisa-ação.....	83
5.2.1 Seleção da Amostra	83
5.2.2 Coleta de Dados.....	84
5.2.3 Análise e Interpretação dos Dados.....	85
5.2.4 Plano de Ação	86
5.2.5 Divulgação dos Resultados.....	87
5.3 Considerações Finais do Capítulo.....	96
Capítulo 6 – Conclusões e Perspectivas Futuras	98
Referências Bibliográficas	99
Glossário.....	103
Anexos.....	108
Apêndices	119

Resumo

TETILA, E. C. **Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.** 2007. Dissertação (Engenharia de Produção) - Universidade Paulista, São Paulo, 2007.

Palavras-chave: gerência de projetos, métricas de software, PMBOK, RUP, *Use Case Points*.

A competição entre organizações que desenvolvem software vem aumentando com o crescimento do mercado de tecnologia da informação, como consequência as organizações têm se preocupado cada vez mais com a melhoria da qualidade de seus produtos de software, com os custos efetivos e com o cumprimento de prazos de seus projetos. Para obter essas características, os processos de desenvolvimento de softwares vêm exigindo um gerenciamento mais efetivo, com um plano de projeto bem definido, baseado em estimativas mais precisas. Dentre as métricas que servem de apoio à gerência de projetos de desenvolvimento de software, destaca-se a estimativa de tamanho na qual, a partir desta dimensão, permite-se definir o esforço, o prazo e os custos necessários para o desenvolvimento do software. Em relação a projetos de softwares orientados a objetos, abordagem bastante utilizada no mercado atual, a estimativa de tamanho é melhor realizada pela métrica *Use Case Points*, ou Pontos de Casos de Uso, proposta em 1993 por Gustav Karner. Dessa forma, este trabalho propõe por meio de levantamento bibliográfico e pesquisa-ação criar um Processo de Estimativa de Software para apoiar a gerência de projetos durante o ciclo de desenvolvimento. Este processo consiste no uso da métrica *Use Case Points* como ferramenta de estimativa; nos processos de gestão de estimativas propostos pelo PMBOK; e nas atividades de desenvolvimento de estimativas propostas pelo RUP. Os resultados da pesquisa-ação demonstraram que, em média, o Processo de Estimativa de Software com *Use Case Points* é 4,3% mais preciso que as métricas de estimativas tradicionais (*Function Points* e *Use Case Points*) sem processo formal de estimativas.

Abstract

TETILA, E. C. **Software Estimative trial with the Use Case Points Metric, PMBOK and RUP.** 2007. 100 f. Dissertation (Production Engineering) - Paulista University, São Paulo, 2007.

Keywords: management of projects, metrics of software, PMBOK, RUP, *Use Case Points*.

The competition between organizations that develop software is increasing with the growth of the market in technology of the information, as consequence the organizations has concerned more and more with the improvement of the quality of its products of software, with the effective costs and with the fulfillment of terms of its projects. To obtain these characteristics, the software development trials require a more effective management, with a well definite project plan based in accurate estimative. Among the meters that serve as support to the software development projects managers, detaches-itself the estimate of size in which, from this dimension, allows define the effort, the term and the necessary costs for the development of the software. Regarding to the software projects oriented to objects, a common approach utilized in the present market, the estimate of size is better carried out through the metric Uses Case Points, proposed in 1993 by Gustav Karner. In that way, this work proposes through bibliographical hoist and action research, create a Software Estimative Trial to support the management of projects during the cycle of development. This trial consists the Use Case Points Metric as a tool of estimative; in the estimative management trials proposed by the PMBOK; and in the estimative of development activities proposed by the RUP. The results of the action research demonstrate that, in average, the Software Estimative Trial with *Use Case Points* is 4.3% more precise than the traditional estimative metrics (Function Points and Use Case Points) without estimative formal trial.

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Lista de Abreviaturas

Lista de Abreviaturas

Abreviatura	Significado
CGS	Características Gerais do Sistema
CPM	<i>Counting Practices Manual</i>
EAP	Estrutura Analítica do Projeto (WBS)
FP	<i>Function Points</i>
GQM	<i>Goal Question Metric</i>
IFPUG	<i>International Function Point Users Group</i>
IP	Índice de Produtividade
KLOC	<i>Short for thousands (Kilo) of Lines of Code</i>
LOC	<i>Lines of Code</i>
OO	Orientado a Objetos
PCU	Pontos de Casos de Uso
PF	Pontos de Função
PMBOK	<i>Project Management Body of Knowledge</i>
PMI	<i>Project Management Institute</i>
RUP	<i>Rational Unified Process</i>
TI	Tecnologia de Informação
UCP	<i>Use Case Points</i>
UML	<i>Unified Modeling Language</i>
WBS	<i>Work Breakdown Structure</i>

Lista de Ilustrações

Figura 1 - Relação entre recursos e riscos durante o ciclo de vida do projeto.....	28
Figura 2 - Ligações entre os grupos de processo em cada fase.....	29
Figura 3 - Sobreposição dos grupos de processos em cada fase	30
Figura 4 - Arquitetura Geral do RUP	35
Figura 5 - As fases e os marcos de um projeto de software.....	39
Figura 6 - Os ciclos de evolução de um projeto de software	40
Figura 7 - Diagrama de casos de uso de um catálogo telefônico	54
Figura 8 - Diagrama de casos de uso de um sistema de saque em conta	59
Figura 9 - Mapeamento das atividades de desenvolvimento de estimativas do RUP na disciplina Gerenciamento de Projeto	66
Figura 10 - Representação gráfica do Processo de Estimativa de Software	68
Figura 11 - Erros de custos obtidos pelas métricas <i>Function Points</i> , <i>Use Case Points</i> e Processo de Estimativa de Software, no <i>Projeto 1</i>	89
Figura 12 - Erros de custos obtidos pelas métricas <i>Function Points</i> , <i>Use Case Points</i> e Processo de Estimativa de Software, no <i>Projeto 2</i>	91
Figura 13 - Erros de custos obtidos pelas métricas <i>Function Points</i> , <i>Use Case Points</i> e Processo de Estimativa de Software, no <i>Projeto 3</i>	92

Lista de Quadros

Quadro 1 - Mapeamento dos processos de gestão de projeto em grupos de processos e áreas de conhecimento	31
Quadro 2 - Mapeamento dos processos de gestão de estimativas propostos pelo PMBOK	65
Quadro 3 - Mapeamento dos processos de gestão de estimativas propostos pelo PMBOK para as atividades de desenvolvimento de estimativas propostas pelo RUP ...	67

Lista de Tabelas

Tabela 1 - Distribuição de esforço e programação de um ciclo de desenvolvimento inicial típico para um projeto de médio porte	40
Tabela 2 - Peso dos Atores.....	56
Tabela 3 - Peso dos casos de uso: opção por número de transações.....	57
Tabela 4 - Peso dos casos de uso: opção por número de classes de objetos	57
Tabela 5 - Peso dos casos de uso: opção por classificação simples da lógica de processamento	57
Tabela 6 - Fatores de Complexidade Técnica.....	58
Tabela 7 - Fatores de Complexidade Ambiental.....	58
Tabela 8 - Perspectivas para a inflação	90
Tabela 9 - Valor absoluto do percentual de erro obtido pelas métricas <i>Function Points</i> , <i>Use Case Points</i> e Processo de Estimativa de Software nos projetos 1, 2 e 3	95

Lista de Anexos

Anexo 1 - Questionário explicativo para identificar os níveis de influências dos fatores de complexidade técnica e ambiental.....	108
Anexo 2 - Lista das instituições de software que receberam o trabalho via e-mail..	114

Lista de Apêndices

Apêndice A - Questionário que utiliza a métrica <i>Function Points</i> para estimar o custo do <i>Projeto 1</i> , da Instituição privada	119
Apêndice B - Questionário que utiliza a métrica <i>Use Case Points</i> para estimar o custo do <i>Projeto 1</i> , da Instituição privada	121
Apêndice C - Questionário que utiliza o Processo de Estimativa de Software para estimar o custo do <i>Projeto 1</i> , da Instituição privada.....	123
Apêndice D - Questionário que utiliza a métrica <i>Function Points</i> para estimar o custo do <i>Projeto 2</i> , da Instituição Pública	125
Apêndice E - Questionário que utiliza a métrica <i>Use Case Points</i> para estimar o custo do <i>Projeto 2</i> , da Instituição Pública	127
Apêndice F - Questionário que utiliza o Processo de Estimativa de Software para estimar o custo do <i>Projeto 2</i> , da Instituição Pública.....	129
Apêndice G - Questionário que utiliza métrica <i>Function Points</i> para estimar o custo do <i>Projeto 3</i> , da Instituição Pública	131
Apêndice H - Questionário que utiliza métrica <i>Use Case Points</i> para estimar o custo do <i>Projeto 3</i> , da Instituição Pública	133
Apêndice I - Questionário que utiliza o Processo de Estimativa de Software para estimar o custo do <i>Projeto 3</i> , da Instituição Pública.....	135
Apêndice J - Quadro comparativo das métricas de estimativas utilizadas no <i>Projeto 1</i> , da Instituição privada	137
Apêndice K - Quadro comparativo das métricas de estimativas utilizadas no <i>Projeto 2</i> , da Instituição Pública	138
Apêndice L - Quadro comparativo das métricas de estimativas utilizadas no <i>Projeto 3</i> , da Instituição Pública	139

1 – Introdução

1.1 Motivação e Relevância do Problema

Conforme dados disponibilizados na página do Ministério da Ciência e Tecnologia (MCT, 2006), o Brasil possui grandes chances de se tornar um competidor importante na área de TI. Um extenso trabalho realizado pela consultoria AT Kearney, sob supervisão do governo federal e da Brasscom, para avaliar as condições do mercado global de serviços de TI compara a situação do Brasil com outros nove mercados. A relação inclui países que marcaram a primeira fase da exportação desses serviços, como o Canadá, até concorrentes mais recentes, que estão crescendo rapidamente no rastro da expansão dos contratos internacionais, como a Índia, que se tornou um vasto celeiro de centros de produção de software sob encomenda nos últimos anos. Segundo o relatório, a oportunidade está longe de se esgotar. A estimativa é de que, entre 2004 e 2008, os gastos totais com serviços de TI vão subir de US\$ 1,081 trilhão para US\$ 1,233 trilhão, um crescimento de 3%.

Embora seja uma evolução discreta, o que importa para o Brasil são os gastos com serviços terceirizados feitos por grandes empresas em países emergentes, cujos custos são menores, e vão aumentar 40% no mesmo período, saindo de US\$ 18 bilhões para US\$ 70 bilhões. É neste mercado que as empresas brasileiras interessam-se (MCT, 2006).

A despeito de suas virtudes, porém, o Brasil ainda tem muitos desafios para tornar-se mais atraente que seus principais competidores. Segundo a empresa de consultoria AT Kearney, conforme notícia disponibilizada na página do Ministério da Ciência e Tecnologia (MCT, 2006), a carga tributária pesa no quesito custo. O

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Capítulo - 1

país apresenta a terceira mais pesada entre os países pesquisados, perdendo apenas para a República Tcheca e a China. Além disso, o país depara-se com outras fragilidades, como o número restrito de empresas com certificação internacional em software, parques tecnológicos com recursos limitados e a falta de identidade do país no mercado internacional de serviços de TI.

Para 2007, as recomendações incluem a ampliação do número de empresas certificadas, além da oferta de linhas de financiamento para preencher lacunas mais críticas na oferta de serviços. A meta do projeto é colocar o Brasil entre os países detentores das cinco mais abrangentes ofertas de serviços terceirizados de TI no mundo (MCT, 2006).

Em consequência desta realidade, as organizações que desenvolvem software têm buscado desenvolver sistemas dentro do prazo e do orçamento previstos e com um nível de qualidade adequada. Essas características, por sua vez, são dependentes de um processo de gerenciamento de desenvolvimento de software confiável. Nesse gerenciamento, é essencial a adoção de guias que sirvam de apoio a um plano de projeto de desenvolvimento efetivo, que englobe os requisitos de qualidade do produto de software. Além de um plano baseado em estimativas e medições precisas de tamanho, esforço, prazos e custos (PMBOK, 2004).

O tamanho do software é um indicador da quantidade de trabalho a ser executado no desenvolvimento de um projeto. Esta dimensão constitui a base para a derivação das estimativas de esforço, custo e prazo necessários para a definição do plano de desenvolvimento do software (CMMI, 2006). Além de subsidiar o planejamento do projeto, a estimativa de tamanho facilita o relacionamento entre cliente e fornecedor; permite o gerenciamento de riscos; o controle do cronograma e

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Capítulo - 1

possibilita o conhecimento da produtividade da equipe, o que beneficia a gerência e a qualidade dos contratos de projetos de software (PRESSMAN, 2002; SOMMERVILLE, 2003; SWEBOK, 2004).

Pesquisa recente da qualidade e produtividade no setor de software brasileiro, realizada pela Secretaria de Política de Informática do Ministério da Ciência e Tecnologia em 2005 (MCT, 2005), demonstraram que a estimativa de tamanho de software ainda está em evolução nas empresas brasileiras. Em uma amostra de 446 organizações do mercado de software brasileiro, em 2001, apenas 21% das organizações detinham algum tipo de métrica para medir a produtividade e a qualidade dos processos de software, sendo que 8% das organizações utilizavam as métricas de forma sistemática (em todos os projetos) e 13% das organizações utilizavam em projetos específicos.

Em uma nova amostra em 2004, com 701 organizações, houve um crescimento de 21% para 39%, sendo 17% fazendo uso das métricas de forma sistemática e 22% em projetos específicos. Dentre as métricas mais utilizadas pelas organizações, 13% utilizavam LOC (*Lines of Code*), 16% *Function Points*, 4% *COSMIC Full Function Point* e 19% *Use Case Points* (MCT, 2005).

É lícito supor que o aumento do uso da tecnologia orientada a objetos (OO) para o desenvolvimento de projetos de software, contribuiu enormemente para a expansão da métrica *Use Case Points* (UCP) no mercado atual, visto que esta foi proposta em 1993 para estimar tamanho de softwares OO.

Além disso, os UCPs colaboram para diminuir a resistência do mercado em adotar métricas de estimativas, pois a métrica utiliza-se dos modelos de casos de uso,

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Capítulo - 1

durante a fase de levantamento de requisitos, o que permite obter estimativas precisas logo no início do ciclo de desenvolvimento do projeto. Outras métricas, porém, requerem informações que nem sempre estão disponíveis no início dos projetos (exemplos: número de linhas de código, número de operadores e operandos, número de pontos de função). Essas informações são essenciais para realizar as estimativas, pois elas que vão auxiliar a discussão de contratos ou a determinação da viabilidade do projeto em termos da análise de prazos, custos e recursos.

Uma outra questão diz respeito à falta de padrões quando da aplicação das estimativas. Não se tem conhecimento na literatura sobre a existência de algum modelo, guia ou processo que auxilie o trabalho de executar estimativas; que avalie o percentual de erro das estimativas em relação aos resultados obtidos; e que oriente a melhoria das estimativas ao longo do tempo.

Nesse sentido, o presente trabalho define um Processo de Estimativa de Software, baseado na métrica UCP, que auxilie o gerente a administrar o ciclo de desenvolvimento do projeto. Isto permite subsidiar o planejamento, comparar e avaliar estimativas, controlar o projeto com mais segurança e providenciar ações de ajustes no plano e no cronograma. Dessa forma, esperam-se reduzir os problemas de gestão, tais como: altos custos, atrasos no cronograma, insatisfação do usuário, dificuldades de medição do progresso do projeto e quebras de contratos.

1.2 Objetivos e Suposições

1.2.1 Objetivo Geral

Este trabalho visa construir um Processo de Estimativa de Software para

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Capítulo - 1

padronizar o trabalho de executar estimativas. Espera-se que à medida que novos artefatos sejam criados durante o projeto, uma melhor forma de estimar, comparar e avaliar as estimativas esteja disponível dentro de um processo padronizado.

1.2.2 Objetivos Específicos

Para atender o objetivo geral, faz-se necessário:

- Mapear a relação entre os processos de gestão de estimativas propostos pelo PMBOK; as atividades de desenvolvimento de estimativas propostas pelo RUP; e a métrica UCP (ferramenta de estimativa do processo).
- Identificar as ações gerenciais a serem tomadas pelo gerente de projeto durante o Processo de Estimativa de Software.

1.2.3 Hipóteses

As hipóteses a serem investigadas nesta pesquisa são:

- Existe relação entre os guias PMBOK, RUP e a métrica UCP?
- Os guias PMBOK, RUP e a métrica UCP podem ser usados de forma combinada?

1.3 Justificativa

- Considerando que existe uma grande oportunidade no mercado de TI;
- Considerando que existe uma busca contínua em desenvolver sistemas dentro do prazo e do orçamento previstos e com um nível de qualidade adequada;

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Capítulo - 1

- Considerando que apenas 39% das organizações detêm algum tipo de métrica para medir a produtividade e a qualidade dos processos de software (MCT, 2005);
- Considerando que organizações têm buscado padronizar os processos utilizados, assim como as práticas relativas ao gerenciamento de projetos;
- Considerando que se esperam reduzir problemas de gestão, tais como: altos custos, atrasos no cronograma, insatisfação do usuário, dificuldades de medição do progresso do projeto e quebras de contratos;

O presente trabalho propõe um Processo de Estimativa de Software que auxilie o gerente de projeto a identificar as ações a serem tomadas durante o trabalho de executar estimativas.

1.4 Metodologia

Conforme ANDRADE (2003), a presente pesquisa caracteriza-se quanto à sua natureza como científico original, pois é uma pesquisa realizada pela primeira vez, que vem contribuir com novas conquistas e descobertas para a evolução do conhecimento científico.

Um Processo de Estimativa de Software foi proposto para a pesquisa de excelência. Dez etapas foram sugeridas para a composição deste processo. As etapas foram definidas com base nos processos de gestão de estimativas propostos pelo PMBOK; nas atividades de desenvolvimento de estimativas propostas pelo RUP; e na métrica UCP, utilizada como ferramenta do processo. Para tal, foi investigada a relação existente entre os guias e a métrica; e identificadas às ações gerenciais a

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Capítulo - 1

serem tomadas pelo gerente de projeto durante o Processo de Estimativa de Software.

Quanto aos objetivos, segundo GIL (2002), uma pesquisa pode ser classificada em: exploratória, descritiva ou explicativa.

A pesquisa exploratória, aplicada neste trabalho, envolve o levantamento bibliográfico, o que proporciona maior familiaridade com o problema, a fim de torná-lo mais explícito.

Com base em procedimentos técnicos, GIL (2002) classifica uma pesquisa de várias formas. Neste trabalho são utilizadas as seguintes pesquisas:

Pesquisa bibliográfica – É desenvolvida com base em material já elaborado, constituído principalmente de livros e artigos científicos. Esta pesquisa é a base para a obtenção dos conhecimentos científicos e técnicos para desenvolver o Processo de Estimativa de Software, proposto por este trabalho.

Pesquisa-ação – Consiste em um tipo de pesquisa com base empírica que é concebida e realizada em estreita associação com uma ação ou com a resolução de um problema coletivo e no qual os pesquisadores e participantes representativos da situação ou do problema estão envolvidos de modo cooperativo ou participativo (THIOLLENT, 1985 *apud* GIL, 2002).

Andrade (2003) informa que uma pesquisa quanto ao objeto pode ser classificada em: bibliográfica, de laboratório e de campo. Uma pesquisa de campo foi realizada com duas instituições de software brasileiras, uma pública e outra privada, para mostrar os conceitos, experimentar as ações do processo e relatar a experiência. Os participantes do estudo desta pesquisa são: os gerentes de projetos das instituições e o aluno Everton Castelão Tetila do curso de Pós-Graduação em

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Capítulo - 1

Engenharia de Produção da UNIP. Os dados coletados foram baseados em questionários utilizados para estimar os custos dos projetos de software; e os resultados foram analisados com base na experiência pessoal dos participantes do estudo.

1.5 Estrutura

Este trabalho consiste de cinco capítulos, incluindo este que é a introdução.

No capítulo 2, são apresentados os guias relacionados à gerência de projetos e ao processo de desenvolvimento de software, de modo que estes auxiliem o gerente a administrar o ciclo de desenvolvimento do projeto.

No capítulo 3, são apresentados os conceitos relacionados à medição de software e as principais métricas de estimativa de tamanho de software existente no mercado.

O capítulo 4 descreve as etapas do Processo de Estimativa de Software, que engloba: mapeamento da relação entre os guias PMBOK, RUP e a métrica UCP; e identificação das ações gerenciais a serem tomadas pelo gerente de projeto durante Processo de Estimativa de Software.

O capítulo 5 relata a pesquisa-ação realizada com duas instituições de software brasileiras.

No capítulo 6, são apresentadas as principais contribuições deste trabalho e as perspectivas de trabalhos futuros.

Finalmente, são apresentados: o questionário explicativo para identificar os níveis de influências dos fatores de complexidade técnica e ambiental (ANEXO 1); a

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Capítulo - 1

lista das instituições de software que receberam o trabalho via e-mail (ANEXO 2); e os questionários utilizados para estimar os custos dos projetos de software (APÊNDICES).

2 – Contribuições da Engenharia de Software

2.1 Introdução

Este capítulo apresenta dois guias para apoiar o Processo de Estimativa de Software. A seção 2.1 descreve de forma sucinta a área de gerência de projetos, com base no *Project Management Body of Knowledge* (PMBOK). A seção 2.2 aborda de forma resumida o *Rational Unified Process* (RUP) e a *Unified Modeling Language* (UML), que foram desenvolvidos para facilitar o planejamento, a execução e o controle de projetos de desenvolvimento de software.

2.2 Gerência de Projetos (*PMBOK*)

2.2.1 Introdução

A área de gerência de projetos teve sua origem na indústria bélica e aeroespacial americana na década de 60, sendo posteriormente adotada na construção civil e em outras áreas da engenharia. Na engenharia de software, a necessidade de aumento da eficácia em suas atividades, possibilitou a propagação do seu conceito e sua utilização. O *Project Management Institute* (PMI) é pioneiro na regulamentação e distribuição deste conhecimento.

O PMI é uma entidade internacional sem fins lucrativos que congrega os profissionais de áreas relacionadas à Gerência de Projetos. Foi fundado em 1969 nos EUA e atualmente está presente em todo o mundo, incluindo o Brasil, onde tem associações locais (*chapter*) em São Paulo, Rio de Janeiro e Minas Gerais. Sua

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Capítulo - 2

missão é promover o profissionalismo e desenvolver o “estado da arte” na gerência de projetos, provendo aos seus associados, serviços e produtos e estabelecendo a aceitação do gerenciamento de projetos como disciplina e profissão.

O PMI especificou um conjunto de procedimentos a fim de padronizar a teoria relacionada à gerência de projetos em um documento denominado *Project Management Body of Knowledge* (PMBOK), disponível em (PMBOK, 2004). Este documento serve como um guia de orientação no apoio à gerência de projetos, cujo propósito é identificar e descrever conceitos e práticas de gerenciamento e padronizar a terminologia e os processos utilizados. Pelo uso deste, é possível objetivar a qualidade, a produtividade e a redução de riscos no decurso do planejamento e controle do projeto, aumentando-se assim, a satisfação do cliente.

A seguir é apresentada uma visão geral do PMBOK com base no *PMBOK Guide*, Terceira Edição. Maiores detalhes podem ser observados em (PMBOK, 2004).

2.2.2 Projeto

Um projeto consiste numa organização sistemática de atividades para a criação de um produto, bem ou serviço. Como em qualquer empreendimento, as atividades devem ser planejadas, programadas e, durante a execução, precisam ser controladas. Pontos claros do início ao fim precisam ser estabelecidos para o alcance de um conjunto de objetivos específicos, incluindo limitações de tempo, custos e recursos, de acordo com o cronograma e o orçamento previstos.

Em (PMBOK, 2004) é definido projeto como “*um empreendimento temporário com o objetivo de criar um produto ou serviço único*”. **Temporário** porque cada projeto tem um começo e um fim bem definidos, e **único** porque o

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Capítulo - 2

produto ou serviço produzido é de alguma forma diferente de todos os outros produtos ou serviços semelhantes.

Algumas características são muito presentes na composição de projetos como: a relação entre cliente e fornecedor; a incerteza da maioria dos projetos em prever o futuro; a necessidade em utilizar técnicas específicas para obter maior possibilidade de êxito. Outras características variam em relação ao desenvolvimento de projetos: podem envolver uma única pessoa ou milhares delas; podem durar poucas semanas ou levar anos; podem compreender apenas uma única organização, ou muitas outras.

Abaixo seguem alguns exemplos de projetos:

- Desenvolver um novo produto ou serviço;
- Implementar uma mudança organizacional em nível de estrutura, de pessoas ou de estilo gerencial;
- Planejar um novo veículo de transporte;
- Desenvolver ou adquirir um sistema de informação novo ou modificado;
- Construir um prédio ou instalações.

Um projeto é um empreendimento que tem um ciclo de vida definido, composto por início, etapas intermediárias e término. Os ciclos definem a natureza dos trabalhos a serem realizados entre as fases, sendo que cada fase do ciclo de vida gera insumos para as fases seguintes. Por exemplo, os requisitos são insumos para a fase de *design* (desenho) do produto a ser construído. Geralmente, os custos e a quantidade de recursos alocados são baixos no início do projeto e aumentam à medida que o projeto se desenvolve, conforme ilustra a Figura 1. Por outro lado, o risco é bastante elevado no início, pois a incerteza no começo do projeto é alta, e

diminui progressivamente à medida que se adquire mais conhecimento.

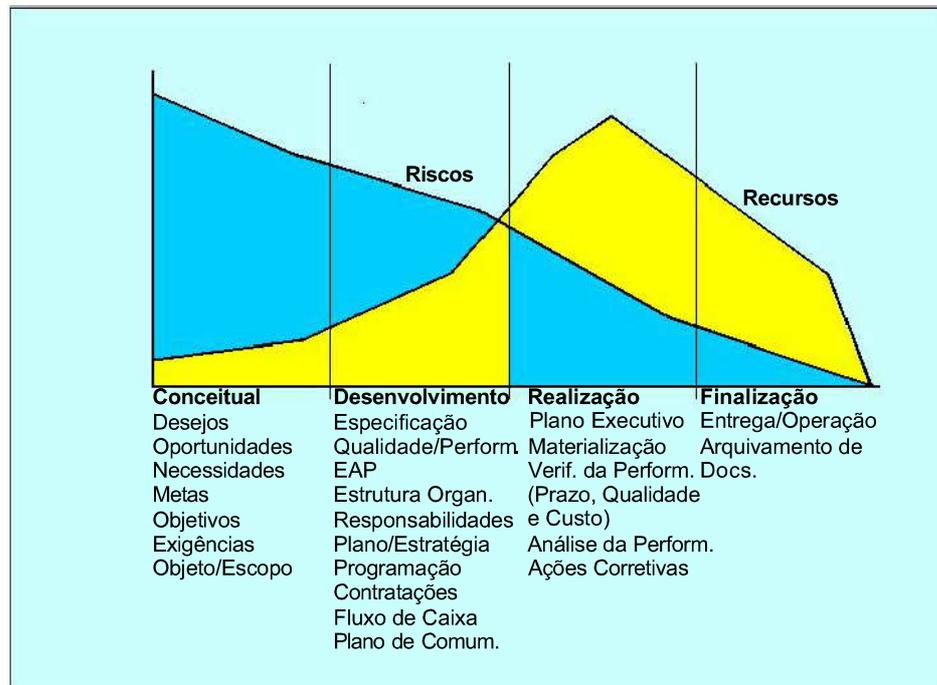


Figura 1 – Relação entre recursos e riscos durante o ciclo de vida do projeto
(Fonte: Adaptado de PMBOK, 2004).

Em geral, o resultado de um projeto é um único produto gerado, mas esse produto pode incluir outras unidades com seu próprio escopo, interdependentes entre si. Para definir o grau de sucesso de um projeto, é preciso verificar se critérios foram atendidos, como o alcance dos objetivos especificados, o cumprimento do prazo previsto, o consumo dos recursos alocados e, principalmente, o nível de satisfação do cliente.

2.2.3 Gerência de Projetos

Conforme em (PMBOK, 2004), Gerência de Projetos é a aplicação de conhecimentos, habilidades e técnicas para projetar atividades que visem atingir os requisitos do projeto. O gerenciamento do projeto é composto por processos, e cada processo organiza e complementa as atividades do projeto, conforme ilustra a Figura

2. Os processos são organizados em cinco grupos:

- **Processos de Iniciação:** estudo da viabilidade e autorização para início;
- **Processos de Planejamento:** definição dos objetivos e da estratégia de implementação, assim como programação das atividades, prazos, custos, riscos e formação da equipe;
- **Processos de Execução:** coordenação das pessoas e recursos para execução do plano do projeto;
- **Processos de Controle:** medição do processo do projeto visando identificar desvios para implementar ações corretivas;
- **Processos de Encerramento:** entrega do produto e formalização da aceitação do trabalho executado.

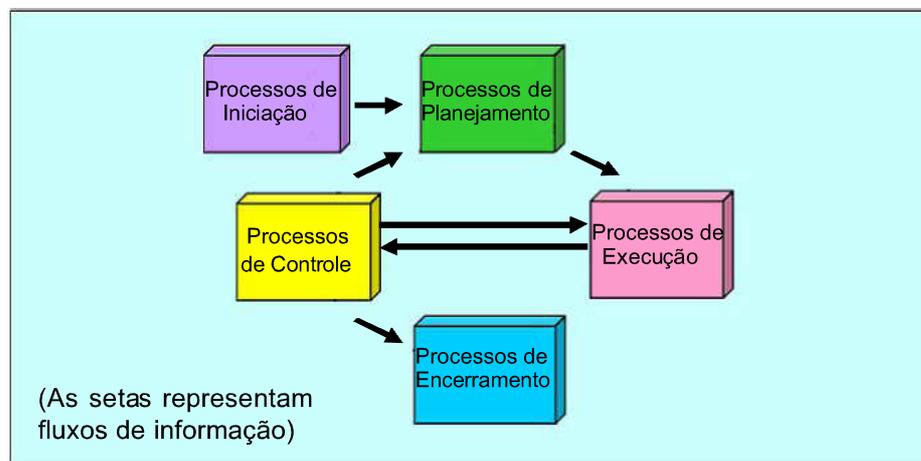


Figura 2 – Ligações entre os grupos de processo em cada fase (Fonte: PMBOK, 2004).

Os grupos de processos estão interligados entre si pelos resultados que produzem, ou seja, a saída de um processo é a entrada do outro. No processo de iniciação são definidos o objetivo e as principais premissas e restrições para o planejamento. Este, por sua vez, fornece à execução um plano de projeto definido. Já

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Capítulo - 2

os processos de controle fornecem o retorno dos trabalhos executados para os demais processos poderem ser ajustados, caso necessário. Contudo, embora representados em seqüência, na prática os processos se sobrepõem e ocorrem com diferentes intensidades durante o ciclo de vida do projeto, conforme ilustra a Figura 3.

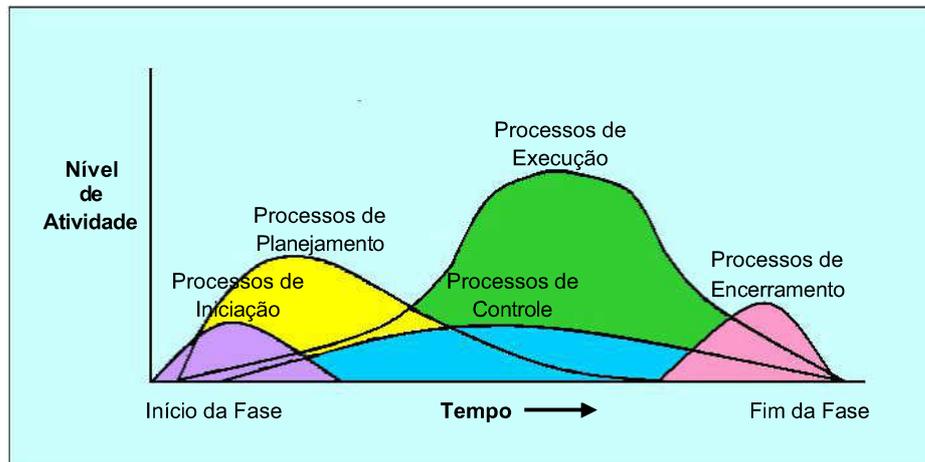


Figura 3 – Sobreposição dos grupos de processos em cada fase (Fonte: PMBOK, 2004).

Os grupos de processos estão organizados em áreas de conhecimento que os compõem e não por fases de projeto. Estas áreas descrevem os conhecimentos e as práticas da gerência de projetos. O Quadro 1 apresenta o mapeamento dos processos entre as áreas de conhecimento que os compõem. Os números indicam os capítulos do PMBOK.

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Capítulo - 2

Quadro 1 - - Mapeamento dos processos de gestão de projeto em grupos de processos e áreas de conhecimento (Fonte: PMBOK, 2004).

GRUPOS DE PROCESSOS / ÁREAS DE CONHECIMENTO	INICIAÇÃO	PLANEJAMENTO	EXECUÇÃO	CONTROLE	ENCERRAMENTO
4. Gerenciamento da integração do projeto	4.1 Desenvolver o termo de abertura do projeto 4.2 Desenvolver a declaração do escopo preliminar do projeto	4.3 Desenvolver o plano de gerenciamento do projeto	4.4 Orientar e gerenciar a execução do Projeto	4.5 Monitorar e controlar o trabalho do projeto 4.6 Controlar mudanças e integração	4.7 Encerrar o projeto
5. Gerenciamento do escopo do projeto		5.1 Planejar o escopo 5.2 Definir o escopo 5.3 Criar EAP		5.4 Verificar o escopo 5.5 Controlar o escopo	
6. Gerenciamento do tempo do projeto		6.1 Definir atividade 6.2 Sequenciar as atividades 6.3 Estimar os recursos da atividade 6.4 Estimar a duração da atividade 6.5 Desenvolver o Cronograma		6.6 Controlar o cronograma	
7. Gerenciamento dos custos do projeto		7.1 Estimar os custos 7.2 Desenvolver a orçamentação		7.3 Controlar os custos	
8. Gerenciamento da qualidade do projeto		8.1 Planejar a qualidade	8.2 Realizar a garantia da qualidade	8.3 Realizar o controle da qualidade	
9. Gerenciamento dos recursos humanos do projeto		9.1 Planejar os recursos humanos	9.2 Contratar ou mobilizar a equipe do projeto 9.3 Desenvolver a equipe do projeto	9.4 Gerenciar a equipe do projeto	
10. Gerenciamento das comunicações do projeto		10.1 Planejar as comunicações	10.2 Distribuir as informações	10.3 Desenvolver o relatório de desempenho 10.4 Gerenciar as partes interessadas	
11. Gerenciamento dos riscos do projeto		11.1 Planejar o gerenciamento de riscos 11.2 Identificar os riscos 11.3 Analisar qualitativamente os riscos 11.4 Analisar quantitativamente os riscos 11.5 Planejar as respostas a riscos		11.6 Monitorar e controlar os riscos	
12. Gerenciamento das aquisições do projeto		12.1 Planejar compras e aquisições 12.2 Planejar contratações	12.3 Solicitar respostas de fornecedores 12.4 Selecionar fornecedores	12.5 Administrar o contrato	12.6 Encerrar o contrato

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Capítulo - 2

O **Gerenciamento da integração do projeto** descreve os processos e as atividades que integram os diversos elementos do gerenciamento de projetos, que são identificados, definidos, combinados, unificados e coordenados dentro dos grupos de processos. Ele consiste nos processos de gestão de projeto: Desenvolver o termo de abertura do projeto, Desenvolver a declaração do escopo preliminar do projeto, Desenvolver o plano de gerenciamento do projeto, Orientar e gerenciar a execução do projeto, Monitorar e controlar o trabalho do projeto, Controlar mudanças e integração e Encerrar o projeto.

O **Gerenciamento do escopo do projeto** descreve os processos na verificação de que o projeto inclui todo o trabalho necessário para que seja concluído com sucesso. Ele consiste nos processos de gestão de projeto: Planejar o escopo, Definir o escopo, Criar EAP (entradas e saídas), Verificar o escopo e Controlar o escopo.

O **Gerenciamento do tempo do projeto** descreve os processos relativos ao término do projeto no prazo correto. Ele consiste nos processos de gestão de projeto: Definir a atividade, Seqüenciar as atividades, Estimar os recursos da atividade, Estimar a duração da atividade, Desenvolver o cronograma e Controlar o cronograma.

O **Gerenciamento dos custos do projeto** descreve os processos envolvidos em planejamento, estimativa, orçamentação e controle de custos, de modo que o projeto termine dentro do orçamento aprovado. Ele consiste nos processos de gestão de projeto: Estimar os custos, Desenvolver a orçamentação e Controlar os custos.

O **Gerenciamento da qualidade do projeto** descreve os processos envolvidos na garantia de que o projeto irá satisfazer os objetivos para os quais foi

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Capítulo - 2

realizado. Ele consiste nos processos de gestão de projeto: Planejar a qualidade, Realizar a garantia da qualidade e Realizar o controle da qualidade.

O **Gerenciamento dos recursos humanos do projeto** descreve os processos que organizam e gerenciam a equipe do projeto. Ele consiste nos processos de gestão de projeto: Planejar os recursos humanos, Contratar ou mobilizar a equipe do projeto, Desenvolver a equipe do projeto e Gerenciar a equipe do projeto.

O **Gerenciamento das comunicações do projeto** descreve os processos relativos à geração, coleta, disseminação, armazenamento e destinação final das informações do projeto de forma oportuna e adequada. Ele consiste nos processos de gestão de projeto: Planejar as comunicações, Distribuir as informações, Desenvolver o relatório de desempenho e Gerenciar as partes interessadas.

O **Gerenciamento dos riscos do projeto** descreve os processos relativos à realização do gerenciamento de riscos em um projeto. Ele consiste nos processos de gestão de projeto: Planejar o gerenciamento de riscos, Identificar os riscos, Analisar qualitativamente os riscos, Analisar quantitativamente os riscos, Planejar as respostas a riscos e Monitorar e controlar os riscos.

O **Gerenciamento das aquisições do projeto** descreve os processos que compram ou adquirem produtos, serviços ou resultados, além dos processos de gestão de contratos. Ele consiste nos processos de gestão de projeto: Planejar compras e aquisições, Planejar contratações, Solicitar respostas de fornecedores, Selecionar fornecedores, Administrar o contrato e Encerrar o contrato.

Maiores detalhes sobre os grupos de processos e as áreas de conhecimento que os compõem podem ser observados em (PMBOK, 2004).

2.3 Processo de Desenvolvimento de Software (RUP)

Atualmente, as ferramentas mais populares para o processo de desenvolvimento de software são a UML e o RUP. A UML é uma linguagem padrão para documentar projetos de software. O RUP é um processo padrão de desenvolvimento de software que utiliza a UML para especificação de sistemas.

A UML é uma linguagem e, como tal, provê um vocabulário e um conjunto de regras para a padronização de uma notação. Faz uso de cinco fases de desenvolvimento, cinco tipos de visões, nove tipos de diagramas e vários modelos de elementos que serão utilizados na criação de diagramas e mecanismos gerais. Todos em conjunto especificam e exemplificam a definição de um sistema OO.

O RUP é um processo padrão de desenvolvimento de software no qual define as etapas do projeto e quem executará cada atividade, o que é executado, e quando e como o trabalho será feito, de modo a atingir os objetivos finais, que é criar um novo software ou evoluir um já existente.

A seguir serão apresentadas as várias facetas do RUP. Para melhor compreensão do texto, é necessário o conhecimento prévio da UML. Uma boa referência sobre essa abordagem pode ser encontrada em (UML, 2005).

2.3.1 *Rational Unified Process (RUP)*

O RUP oferece uma abordagem baseada em disciplinas para atribuir tarefas e responsabilidades dentro de uma organização de desenvolvimento. Sua meta é garantir a produção de software de alta qualidade que atenda às necessidades dos usuários dentro de um cronograma e orçamento previsíveis. A Figura 4 mostra a

arquitetura geral do RUP.

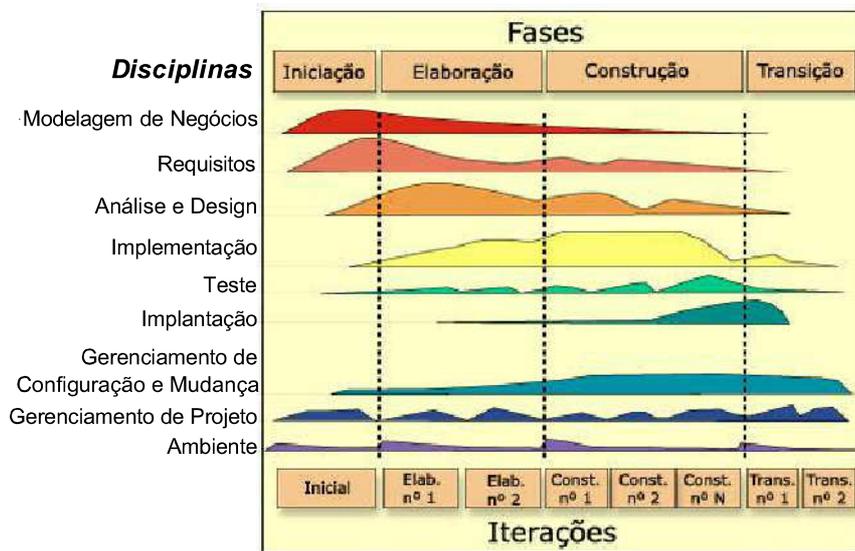


Figura 4 – Arquitetura geral do RUP (Fonte: RUP, 2002).

Conforme a Figura 4, a arquitetura do RUP é composta por duas dimensões. O eixo horizontal representa o tempo em relação aos aspectos do ciclo de vida do processo à medida que se desenvolve. O eixo vertical representa as disciplinas, que agrupam as atividades de maneira lógica, por natureza. A primeira dimensão representa o aspecto dinâmico do processo quando ele é aprovado, expresso em termos de fases, iterações e marcos. A segunda dimensão representa o aspecto estático do processo, como ele é descrito em termos de componentes, disciplinas, atividades, fluxos de trabalho, artefatos e papéis do processo.

O gráfico mostra como a ênfase varia através do tempo. Nas iterações iniciais, maior parte do tempo é dedicado aos requisitos. Já nas iterações posteriores, maior parte do tempo é consumido com a implementação.

2.3.2 Disciplinas do RUP

As disciplinas são procedimentos compostos de atividades logicamente

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Capítulo - 2

seqüenciadas e têm objetivos específicos em relação ao projeto. Cada atividade da disciplina tem a finalidade de criar ou atualizar um ou mais artefatos. O RUP possui nove disciplinas: modelagem de negócio, requisitos, análise e *design*, implementação, teste, gerenciamento de configuração e mudança, gerenciamento de projeto, ambiente e implantação. Como o foco de cada iteração, em cada fase do projeto, muda em função dos seus objetivos, dá-se maior e menor ênfase a cada uma das disciplinas em cada momento do projeto.

A seguir é apresentada uma breve descrição de cada uma das nove disciplinas do RUP. Uma descrição detalhada de cada disciplina pode ser observada em (RUP, 2002).

➤ ***Modelagem de Negócio***

A disciplina Modelagem de Negócio descreve como desenvolver uma visão da organização-alvo e, com base nesta visão, define os processos, os papéis e as responsabilidades dessa organização em um modelo de casos de uso de negócios e em um modelo de objetos de negócios.

➤ ***Requisitos***

A disciplina Requisitos tem vários objetivos como: estabelecer e manter um acordo com o cliente e com os *stakeholders* quanto às funções que o sistema deve desempenhar; esboçar a interface de usuário; fornecer à equipe de desenvolvimento uma tradução clara dos requisitos do sistema; e prover informações para o planejamento do projeto, como conteúdo técnico abordado nas iterações e subsídios para estimativas de prazo e custo.

➤ ***Análise e Design***

O objetivo desta disciplina é traduzir os requisitos numa especificação que indique como o sistema deve ser implementado. Para tal, é definida a arquitetura selecionada, o ambiente operacional, a escalabilidade, a performance, dentre outras características.

➤ ***Implementação***

Esta disciplina tem quatro objetivos principais:

- Definir a organização do código em termos de subsistemas de implementação organizados em camadas;
- Implementar classes, objetos em termos de componentes (arquivos-fonte, binários, executáveis e outros);
- Testar os componentes desenvolvidos como unidades;
- Integrar os resultados produzidos pelos programadores ao sistema executável.

➤ ***Teste***

A disciplina de Teste se concentra em avaliar a qualidade do produto, verificando e corrigindo problemas e má interpretação dos requisitos. São verificados: a integração e a interação entre os componentes, se todos os requisitos foram implementados conforme especificados e se todos os *bugs* localizados foram corrigidos.

➤ ***Gerenciamento de configuração e mudança***

O Gerenciamento de Configuração se concentra em registrar e manter uma

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Capítulo - 2

trilha das mudanças e da evolução dos artefatos produzidos pelo projeto. O Gerenciamento de Mudança, por sua vez, provê um processo estruturado para controlar as mudanças solicitadas para os artefatos do projeto, decorrentes de correções de falhas, melhoria de qualidade e inclusão de novos requisitos.

➤ ***Gerenciamento de projeto***

O Gerenciamento de Projeto é a arte de confrontar os objetivos da concorrência, gerenciar riscos e superar obstáculos para liberar com êxito um produto que atenda às necessidades dos clientes (que pagaram por ele) e dos usuários.

➤ ***Ambiente***

A disciplina Ambiente concentra-se nas atividades necessárias à configuração do processo para um projeto. Ela descreve as atividades para o desenvolvimento das diretrizes de suporte de um projeto. A meta das atividades dessa disciplina é oferecer à organização o ambiente de desenvolvimento de software, em nível de processos e ferramentas, que dará suporte à equipe de desenvolvimento.

➤ ***Implantação***

O objetivo da disciplina Implantação é disponibilizar o sistema para os usuários, o que inclui:

- Teste do sistema no ambiente de produção;
- Empacotamento do software para distribuição;
- Distribuição do software;
- Instalação do software;

- Treinamento dos usuários e equipe comercial;
- Migração de dados para novo sistema.

2.3.3 Fases do RUP

No RUP, o ciclo de vida do software é dividido em quatro fases seqüenciais: **Iniciação, Elaboração, Construção e Transição**, sendo cada uma concluída por um marco principal. Esse marco consiste em revisar o estado do projeto no final de uma fase para determinar se os objetivos foram alcançados. Uma avaliação satisfatória permite que o projeto avance para a próxima fase. Cada fase é basicamente um intervalo de tempo entre dois marcos principais, conforme ilustra a Figura 5.

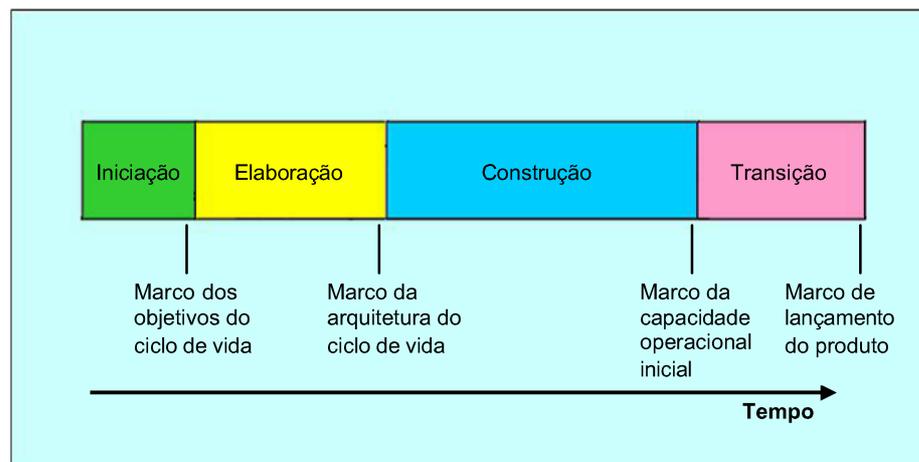


Figura 5 – As fases e os marcos de um projeto de software (Fonte: RUP, 2002).

Uma passagem pelas quatro fases é um **ciclo de desenvolvimento**. Um ciclo de desenvolvimento produz uma **geração** do software. À medida que o produto atravessa vários ciclos, novas gerações vão sendo produzidas, repetindo a mesma seqüência das fases: iniciação, elaboração, construção e transição. Os ciclos subseqüentes são chamados de **ciclos de evolução**, conforme ilustra a Figura 6.

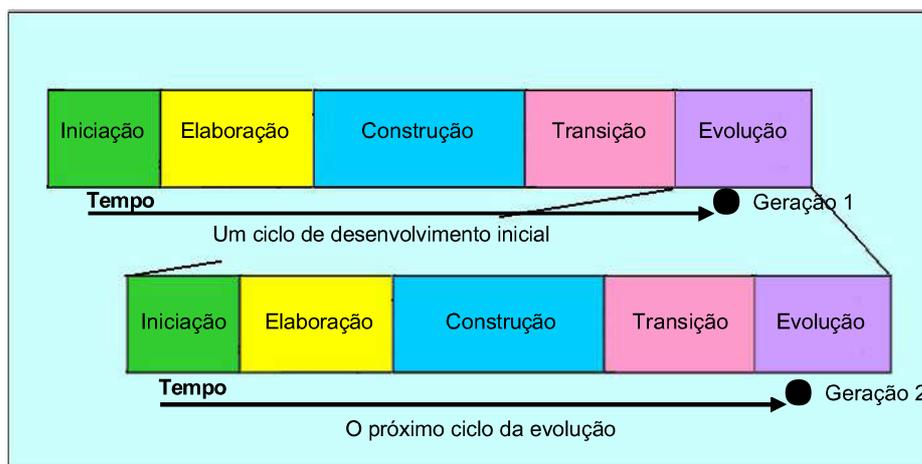


Figura 6 – Os ciclos de evolução de um projeto de software (Fonte: RUP, 2002).

Os ciclos de evolução foram concebidos para serem dinâmicos e adaptativos, podendo acomodar mudanças muito comuns nos projetos de software, como melhorias sugeridas pelos usuários, mudanças no contexto do usuário, mudanças na tecnologia subjacente, reação à concorrência, entre outros. As fases de Iniciação e Elaboração são normalmente menores que as fases de Construção e Transição nos ciclos de evolução, pois a definição e a arquitetura básicas do produto destas fases já foram determinadas nos ciclos de desenvolvimento anteriores. No entanto, são exceções a essa regra, os ciclos de evolução em que ocorre uma redefinição significativa do produto ou da arquitetura.

Segundo o RUP (2002), as fases não são idênticas em termos de programação e esforço. Embora varie muito de acordo com o projeto, um ciclo de desenvolvimento inicial típico para um projeto de médio porte deve prever a seguinte distribuição de esforço e programação (Tabela 1):

	Iniciação	Elaboração	Construção	Transição
Esforço	~5 %	20 %	65 %	10%
Programação	10 %	30 %	50 %	10%

Tabela 1 – Distribuição de esforço e programação de um ciclo de desenvolvimento inicial típico para um projeto de médio porte (Fonte: RUP, 2002).

A seguir é apresentada uma breve descrição de cada uma das quatro fases do RUP. Uma descrição detalhada de cada fase pode ser observada em (RUP, 2002).

➤ **Iniciação**

Também conhecida por Concepção, a meta dominante desta fase é atingir o consenso entre todos os envolvidos sobre os objetivos do ciclo de vida do projeto. A fase de iniciação tem muita importância principalmente para os esforços dos desenvolvimentos novos, nos quais há muitos riscos de negócios e de requisitos que precisam ser tratados para que o projeto possa prosseguir. Para projetos que visam melhorias em um sistema existente, a fase de iniciação é mais rápida, mas ainda se concentra em assegurar que o projeto seja compensatório e que seja possível fazê-lo.

➤ **Elaboração**

Na fase de Elaboração, a meta é criar a *baseline* para a arquitetura do sistema a fim de fornecer uma base estável para o esforço da fase de construção. A arquitetura se desenvolve a partir de um exame dos requisitos mais significativos (aqueles que têm grande impacto na arquitetura do sistema) e de uma avaliação de risco. A estabilidade da arquitetura é avaliada por meio de um ou mais protótipos de arquitetura.

➤ **Construção**

Na fase de Construção, a meta é esclarecer os requisitos restantes e concluir o desenvolvimento do sistema com base na arquitetura da *baseline*. A fase de Construção basicamente consiste num processo de manufatura, em que a ênfase está no gerenciamento de recursos e controle de operações para otimizar custos,

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Capítulo - 2

programações e qualidade. Nesse sentido, a mentalidade do gerenciamento passa por uma transição do desenvolvimento da propriedade intelectual durante a iniciação e elaboração, para o desenvolvimento dos produtos que podem ser implantados durante a construção e transição.

➤ *Transição*

Na fase de Transição, a meta é assegurar que o software esteja disponível para seus usuários finais. A Fase de Transição pode atravessar várias iterações e inclui testar o produto em preparação para *release* e ajustes pequenos com base no *feedback* do usuário. Nesse momento do ciclo de vida, o *feedback* do usuário deve priorizar o ajuste fino do produto, a configuração, a instalação e os problemas de usabilidade; todos os problemas estruturais mais graves devem ter sido trabalhados anteriormente no ciclo de vida do projeto.

2.4 Considerações Finais do Capítulo

Neste capítulo foram apresentados dois guias para apoiar o Processo de Estimativa de Software: PMBOK e RUP.

O RUP é um processo padrão de desenvolvimento de software bastante consolidado e validado comercialmente. Uma de suas principais características é o uso da abordagem iterativa que proporciona melhor entendimento dos problemas por meio de refinamentos sucessivos. O uso da orientação a objetos e da notação UML também contribuem para esse fim pela utilização de modelos visuais. Em relação à adoção da arquitetura baseada em componentes, esta por sua vez, viabiliza o reuso e a utilização de componentes em larga escala, garantindo a extensibilidade do sistema.

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Capítulo - 2

Além disso, o RUP utiliza-se de uma abordagem disciplinada para dividir atividades e responsabilidades numa organização, o que possibilita a produção de software com um nível de qualidade adequada, atendendo às necessidades dos clientes. No entanto, não são abordadas com grande clareza questões referentes à contratação de pessoal, treinamento, gerenciamento do orçamento e gerenciamento de contratos.

O PMBOK, por sua vez, engloba todas as áreas de conhecimento que regem as regras da Gerência de Projetos, sendo o mais importante guia do mundo nessa área. É um material genérico que serve para todas as áreas de conhecimento, ou seja, tanto para construção de edifício ou processo de fabricação industrial quanto para a produção de software. Porém, como não se trata de um modelo específico para a produção de software, necessita ser customizado para atender as características do seu desenvolvimento.

3 – Métricas de Software

3.1 Introdução

Dentro do contexto da gerência de projetos, medidas de atributos específicos do processo, projeto e produto são usadas para calcular métricas de software. Essas métricas podem ser analisadas para obterem indicadores que orientam as ações gerenciais e técnicas.

Os indicadores de processo determinam a eficácia de um processo existente, permitindo ao gerente de projeto avaliar os pontos fortes e fracos da organização. O modelo de processo, as tarefas de engenharia, os produtos de trabalho e os marcos de tempo, são alguns exemplos de indicadores de processo. Por outro lado, os indicadores de projeto permitem ao gerente de projeto avaliar o estado de um projeto em andamento, acompanhar riscos potenciais, descobrir áreas-problema antes que estas se tornem críticas, ajustar o fluxo de trabalho e avaliar a capacidade da equipe de projeto em controlar a qualidade dos produtos obtidos. Já os indicadores de produto definem as características do produto em termos de desempenho, confiabilidade, eficiência, usabilidade, nível de manutenibilidade, nível de portabilidade e nível de reutilização (PRESSMAN, 2002).

Com a necessidade de medidas que informem a eficiência do desenvolvimento de software, diversas métricas ou métodos de medição foram propostos a fim de minimizar os fracassos dos projetos obtidos, principalmente em relação às falhas no cronograma e orçamento previstos.

Este capítulo apresenta os conceitos relacionados à medição de software e as

principais métricas de estimativa de tamanho de software existente no mercado.

3.2 Medição de Software

Conforme PRESSMAN (2002), a medição é fundamental para qualquer atividade de engenharia, ela permite obter-se, através dos números, o entendimento do comportamento daquilo que está sendo medido, oferecendo um mecanismo para avaliação objetiva. Lord Kelvin em (TUNBRIDGE, 1992) ressalta que “quando é possível medir o que se fala e expressá-lo em números, você sabe algo sobre isso; mas quando não é possível medir e quando não se consegue expressar em números, seu conhecimento é insatisfatório; pode ser o início do aprendizado, mas você, em seus pensamentos, pouco avançou ao estágio da ciência”.

A medição tomou proporções tamanhas na engenharia de software devido à necessidade de se trabalhar com orçamentos definitivos, prazos estipulados, esforços sob domínio, produção com planejamento e capacidade de gerenciamento. Esses elementos são essenciais para um desenvolvimento profissional e são possíveis somente com instrumentos relevantes de mensuração. Dessa forma, a medição é indispensável para obter o entendimento do desenvolvimento, operação e manutenção do software, oferecendo um mecanismo para avaliação objetiva.

PRESSMAN (2002), SOMMERVILLE (2003) e SOWEBOK (2004) concordam que para ser efetiva a medição deve focar em objetivos específicos, ser aplicada durante todo o ciclo de desenvolvimento do software de modo que soluções possam ser desenvolvidas e aperfeiçoamentos reais possam ser conseguidos ao longo do tempo, com base na caracterização e no entendimento do contexto organizacional.

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Capítulo - 3

No entanto, medição resulta em mudança cultural numa organização devido a três elementos-chave que devem ser considerados para implementar um programa de medição: (1) coletar dados; (2) calcular; e (3) analisar métricas.

Primeiro deve-se definir um conjunto limitado de medições de processo, projeto e produto que sejam fáceis de coletar. Em geral, uma abordagem orientada a metas, como o paradigma de GQM (*goal-question-metric* – objetivo-questão-métrica), de BASILI *et al.* (1994) ajuda a organização a focalizar as métricas adequadas para o negócio. A chave para a seleção das características cuja medição seja relevante está na identificação dos objetivos estabelecidos tanto para o processo de desenvolvimento e manutenção de sistemas quanto para o produto gerado em si. Essas medições devem ser calculadas e normalizadas por meio de métricas específicas, como métricas orientadas a tamanho ou função. Logo, é possível analisar e comparar o resultado com médias anteriormente obtidas de projetos semelhantes desenvolvidos dentro da organização. Desta forma, tendências podem ser avaliadas e conclusões podem ser geradas.

Além disso, a medição é mais efetiva quando implementada de forma integrada com as atividades técnicas e de gestão que definem os projetos de software, pois a medição fornece informações objetivas relacionadas aos riscos e problemas que podem ter impacto nos objetivos dos projetos definidos.

Contudo, o uso da medição e de métricas sistemáticas de software ainda é relativamente incomum em organizações que desenvolvem software. “Existe uma relutância em introduzir a medição, porque os benefícios não são bem definidos. Uma razão para isso é que, em muitas empresas, os processos de software utilizados ainda são organizados de maneira precária e não estão suficientemente aperfeiçoados

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Capítulo - 3

para fazer uso de medições. Outra razão é que não há padrões para as métricas, e daí decorre o suporte limitado para a coleta e análise de dados. A maioria das empresas não estará preparada para introduzir medições até que esses padrões e essas ferramentas estejam disponíveis” (SOMMERVILLE, 2003).

3.3 Métricas de Estimativa de Tamanho de Software

A estimativa de tamanho de software é um processo pelo qual uma pessoa ou um grupo de pessoas estima o tamanho de um produto de software (MCPHEE, 1999). As estimativas de tamanho constituem a base para a derivação das estimativas de esforço, prazo e custos (CMMI, 2006). O tamanho, geralmente, tem impacto na solução técnica e na gestão do projeto já que, estimativas imprecisas podem levar ao fracasso do projeto (ROSS, 1999).

O tamanho do software é um indicador da quantidade de trabalho a ser executado no desenvolvimento de um projeto. Um projeto é estimado de acordo com o tamanho físico e de acordo com o grau de reuso de componentes. O tamanho físico é estimado por meio da medição da especificação de requisitos, análise, projeto e código, com base nas funções que o usuário obtém, na complexidade do problema que o software irá resolver. O grau de reuso de componentes do projeto mede o quanto o produto será copiado ou modificado a partir de um outro produto existente (FENTON & PFLEEGER, 1998).

Para subsidiar o desenvolvimento ou a manutenção de um projeto de software, as empresas requerem, no início do projeto, estimativas de tamanho de softwares para determinar o faturamento dos serviços cobrados e tornar possível

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Capítulo - 3

fechar o plano de trabalho. No entanto, a precisão de estimativas realizadas num nível abstrato de informações não é grande, precisando ser refinada à medida que se obtêm maiores detalhes do projeto, para uma medição mais confiável. Existe, portanto, uma demanda por métricas de software que proporcionem estimativas com maior precisão, sob um nível mais abstrato para a construção de tal produto.

Várias métricas de estimativa de tamanho de software foram desenvolvidas e melhoradas desde o final da década de 1960 dentre elas:

- Linhas de Código, disponível em (PRESSMAN, 2002).
- Delphi, disponível em (DIETZ, 1987).
- Halstead, disponível em (HALTEAD, 1977).
- Complexidade Ciclomática, disponível em (MCCABE, 1976).
- Regressão Linear, disponível em (FERNANDES, 1995).
- Putnam's Slim Model, disponível em (PUTNAM, 1978).
- Function Points, disponível em (IFPUG, 2004).
- COCOMO II, disponível em (BOEHM *et al.*, 2000).
- Feature Points, disponível em (JONES, 1996).
- Use Case Points, disponível em (KARNER, 1993).

Esta seção apresenta três das principais métricas de estimativa de tamanho de software existente no mercado.

3.3.1 Linhas de Código – LOC

As primeiras tentativas de se medir o tamanho de um sistema, conforme BRAGA (1996), levou em consideração as Linhas de Código - LOC (*Lines of Code*). Não se trata propriamente de uma métrica de estimativa, mas de uma técnica de contagem empírica, que conseguiu traduzir dimensão em número de linhas de instruções, por meio de linguagens específicas de programação. É baseada na idéia de que “certo sistema possui um número maior de LOCs que um outro, portanto, é maior e mais complexo”(BRAGA, 1996).

Como técnica, serve à mensuração de programas já construídos para estabelecer elementos de previsão de dimensões de programas futuros, como erros por KLOC (milhares de linhas de código), defeitos por KLOC, custo por LOC, páginas de documentação por KLOC, entre outros. Exige uma cultura de documentação histórica de dados, constantemente reavaliados e atualizados, alta padronização dos tipos de rotinas e trabalho com linguagens específicas e não mutáveis. A exigência pela manutenção de históricos sobre as dimensões já realizadas, devidamente tratadas em suas inferências é essencial para que seja possível utilizar tais informações como base nas estimativas dos próximos projetos.

Como pode ser percebido, apesar de aparentemente intuitiva, a contagem de LOC pode envolver uma série de regras para ser realizada. E não há um padrão de fato para essa contagem.

BRAGA (1996), PRESSMAN (2002) e VAZQUEZ (2005) concordam que existem inúmeros reveses na utilização de LOC como unidade para medir tamanho de sistemas como:

- Falta de padronização;
- Dificuldade de aplicação nas fases iniciais do ciclo de vida;
- Dependência da linguagem de programação utilizada;
- Não considera as condições ambientais de trabalho;
- Não considera os recursos computacionais utilizados;
- Penaliza programas curtos, mas bem projetados.

3.3.2 Function Points – FP

Os *Function Points* (FP), ou Pontos de Função (PF), foram inicialmente concebidos por Allan Albrecht em 1979 e, posteriormente, com o advento da divulgação da métrica, tornou-se necessário definir um guia que interpretasse as regras originais para os novos ambientes. Devido a essa necessidade, em 1986 foi criado o *International Function Point Users Group* (IFPUG), Grupo Internacional de Usuários de Pontos por Função (LONGSTREET, 2004). Em 2002 os FP passaram à condição de padrão internacional pela norma ISO/IEC 20926 (AGUIAR, 2003).

Os FP visam estabelecer uma medida de “tamanho” do software, em pontos por função, por meio da quantificação e complexidade das funcionalidades desempenhadas pelo software, independentemente de metodologia, tecnologia e plataforma utilizadas no desenvolvimento do sistema (HAZAN, 2000; LONGSTREET, 2002). Baseia-se na visão do usuário e mede o **que** é o sistema, ou seja, o tamanho funcional¹ da aplicação, não considerando **como** o sistema atua para

¹ Tamanho funcional é a medida de tamanho de software, baseada em uma avaliação padronizada dos requisitos lógicos dos usuários (DEKKERS, 1998).

produzir os resultados.

A contagem dos FP se dá sobre cinco elementos básicos de um sistema modelado: Arquivo Lógico Interno (ALI), Arquivo de Interface Externa (AIE), Entrada Externa (EE), Saída Externa (SE) e Consulta Externa (CE). Cada um desses elementos, representativos da base de dados ou das transações, terá um peso na complexidade do sistema, de acordo com o seu nível de influência. O valor total desses pesos determina o valor total dos **PF não-ajustados**.

No entanto, esse valor deve ser ajustado em relação às 14 características gerais do sistema (CGS), que refletem funções que afetam a aplicação de maneira geral. Cada uma dessas características possui um nível de influência sobre a aplicação, que pode variar em um intervalo discreto de zero a cinco: comunicação de dados, desempenho, volume de transações e reusabilidade são alguns exemplos. Após determinar o fator de ajuste, baseado nas 14 características gerais do sistema, este deve ser multiplicado pelos PF não-Ajustados. Logo então, são determinados os **PF ajustados** do sistema, que determinarão as estimativas de esforço, prazo e custos do projeto.

Apesar dos FP serem muito utilizados no mercado, autores como CALDIERA *et al.* (1998), DEKKERS (1998) e FREIRE (2003) afirmam que existem desvantagens no uso da métrica como:

- É subjetiva e possui diferença entre contadores;
 - Exige experiência para efetuar uma contagem acurada;
 - É pouco flexível às mudanças de requisitos;
-

- Possui bons resultados somente à medida que artefatos da fase de análise e projeto vão sendo gerados;
- É dependente da história da produtividade da própria empresa, ou de um rol de outras anteriores conhecidas;

Para estimar o tamanho do software, o IFPUG definiu os procedimentos de contagem dos FP, descritos no Manual de Práticas de Contagem (CPM - *Counting Practices Manual*), disponível em (IFPUG, 2004). Atualmente a versão mais recente do CPM é a *Release 4.2*.

3.3.3 Use Case Points – UCP

3.3.3.1 Origem e Definição

Os *Use Case Points* (UCP), ou Pontos de Casos de Uso (PCU), foram propostos em 1993, por Gustav Karner, com base nos FP, *Mark II*, e no Modelo de Casos de Uso para determinar estimativas de tamanho de softwares OO. Os UCP visam estabelecer uma medida de “tamanho” do software, em pontos por casos de uso, por meio da quantificação e complexidade das funcionalidades desempenhadas pelo software. Baseia-se na visão do usuário e tem como proposta ser utilizado logo no início do ciclo de desenvolvimento, na fase de definição dos requisitos, com base no modelo de casos de uso. Dessa forma, a métrica trabalha num nível mais abstrato que os FP.

Nesta métrica, KARNER (1993) substitui alguns fatores técnicos propostos pelos FP; cria os fatores ambientais; propõe uma estimativa de produtividade de 20 homens/hora por PCU e explora a medição da funcionalidade do sistema baseado no

modelo de casos de uso.

O **modelo de casos de uso** utiliza a UML como linguagem padrão para elaboração de modelagem de softwares OO e tem sido muito utilizado no mercado atualmente. Trata-se de uma técnica utilizada para capturar e descrever os requisitos funcionais de um sistema por meio de diagramas e descrições de casos de uso (ANDA, 2002).

Conforme em (UML, 2005), os elementos básicos associados com o diagrama de casos de uso são:

- **Ator:** representa o papel de uma entidade externa que interage com o sistema modelado, como um usuário, um hardware, ou um outro sistema. Graficamente, um ator é representado por um “homem palito” com o seu nome especificado acima ou abaixo do ícone;
- **Caso de Uso:** representa o comportamento de um sistema, ou parte dele, pelo conjunto de seqüências de ações executadas dentro do contexto deste sistema e que produz um resultado de valor observável por um ator. Graficamente, um caso de uso é representado por uma elipse com linhas contíguas, com o seu nome especificado ao centro, ou abaixo da elipse;
- **Sistema:** representa o contexto sob a consideração a que um conjunto de objetos se aplica, apresentando, desta forma, o comportamento destes objetos dentro deste contexto. Graficamente, um sistema é representado por um retângulo com o seu nome especificado em seu interior.

Em essência, um diagrama de casos de uso é utilizado para descrever a funcionalidade da aplicação e a sua interação com os usuários. Em sua forma mais

simples, um diagrama de casos de uso apresenta três elementos básicos de modelo de casos de uso: ator, caso de uso e sistema. A Figura 7 mostra a notação básica da UML para diagramas de casos de uso.

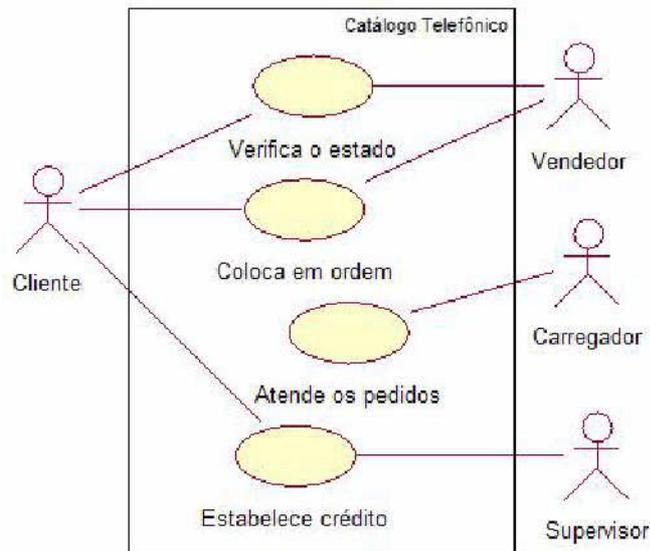


Figura 7 – Diagrama de casos de uso de um catálogo telefônico.

De fato, outros elementos de modelo podem ser usados em diagramas de casos de uso, tais como associações de dependência entre casos de uso e relações de generalização entre casos de uso e entre atores. Maiores detalhes sobre a técnica de modelagem de casos de uso podem ser observados em (UML, 2005).

A contagem dos UCP se dá sobre dois elementos básicos de um sistema modelado: **atores** e **casos de uso**. Cada um desses elementos terá um peso na complexidade do sistema, de acordo com o seu nível de influência. O valor total desses pesos determina o valor total dos **PCU não-ajustados**.

No entanto, esse valor deve ser ajustado em relação aos fatores de complexidade técnica e ambiental que refletem funções que afetam a aplicação de maneira geral. Os fatores de complexidade técnica variam numa escala de 0 a 5, de acordo com o grau de dificuldade do sistema a ser construído: desempenho da

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Capítulo - 3

aplicação, portabilidade e facilidade de manutenção, são alguns exemplos. Os fatores de complexidade ambiental indicam a eficiência do projeto, numa escala de 0 a 5, e estão relacionados ao nível de experiência dos profissionais e às condições ambientais e de trabalho, como a capacidade do líder de projeto, a motivação da equipe e a experiência com a aplicação de desenvolvimento. Após determinar os fatores de complexidade técnica e ambiental, estes devem ser multiplicados pelos PCU não-ajustados. Logo então, são determinados os **PCU ajustados** do sistema que determinarão as estimativas de esforço, prazo e custos do projeto.

Uma vantagem evidente da métrica UCP sobre os FP é que ela utiliza-se de um tipo de documento essencial em metodologias dirigidas por casos de uso. Neste sentido, destaca-se o RUP, que utiliza a UML como linguagem padrão para elaboração da modelagem de softwares OO. Dessa forma, é possível calcular prontamente mudanças nas estimativas do sistema a cada pequena alteração de requisitos, refazendo-se apenas alguns cálculos. Os FP, ao contrário, exigem que novos documentos para o cálculo das estimativas sejam adicionados ao sistema a cada pequena mudança no orçamento, prazo ou requisitos, sendo, dessa forma, pouco flexível a essas mudanças.

Uma outra razão, é que “os UCP são baseados em casos de uso e centrados no usuário, sendo mais adequado para sistemas orientados a objetos. Ele descreve a visão do sistema, independente como atua para produzir os resultados. Desse modo, os UCP são mais robustos e menos mutáveis que os FP ou LOC” (DAMODARAN & WASHINGTON, 2002).

Além disso, “os UCP contribuem para a diminuição de algumas dificuldades impostas pelo mercado em relação à resistência de adoção de métricas de estimativa,

porque é um método simples, fácil de usar e rápido de se aplicar” (DAMODARAN & WASHINGTON, 2002).

Para estimar o tamanho do software de acordo com os PCU, Karner definiu os processos de contagem conforme descritos a seguir.

3.3.3.2 Processo de Contagem de PCU

O processo de contagem dos PCU compõe-se de seis etapas (KARNER, 1993):

- **Contar os atores e atribuir o grau de complexidade:** identificar e multiplicar o total de atores, de acordo com o tipo de complexidade (simples, médio ou complexo), pelo seu respectivo peso (1, 2 ou 3), conforme a Tabela 2. Somar os produtos para obter o total de atores não ajustados.

Complexidade	Definição	Peso
SIMPLES	Representa um outro sistema com Interface definida de Programas.	1
MÉDIO	Representa um outro sistema que interage por meio de protocolos ou quando o ator representa uma pessoa e utiliza terminal sem capacidade gráfica.	2
COMPLEXO	Interage por meio de uma Interface Gráfica ou página Web.	3

Tabela 2 – Peso dos Atores (Fonte: KARNER, 1993).

- **Contar os casos de uso e atribuir o grau de complexidade:** a complexidade de cada caso de uso pode ser definida de acordo com três opções de classificação: (1) Número de transações; (2) Número de classes de objetos; ou (3) Classificação simples da lógica de processamento. De acordo com a opção escolhida, deve-se multiplicar cada caso de uso pelo seu respectivo peso, conforme a Tabela 3 (Número de transações), a Tabela 4 (Número de classes de objetos), ou a Tabela 5 (Classificação simples da lógica de

processamento).

Complexidade	Definição	Peso
SIMPLES	O caso de uso possui 3 ou menos transações incluindo fluxos alternativos.	5
MÉDIO	O caso de uso possui de 4 a 7 transações incluindo fluxos alternativos.	10
COMPLEXO	O caso de uso possui acima de 7 transações incluindo fluxos alternativos.	15

Tabela 3 – Peso dos casos de uso: opção por número de transações (Fonte: KARNER, 1993).

Complexidade	Definição	Peso
SIMPLES	O caso de uso pode ser realizado com menos de 5 objetos de análise.	5
MÉDIO	O caso de uso pode ser realizado com 5 a 10 objetos de análise.	10
COMPLEXO	O caso de uso necessita ser realizado com pelo menos 10 objetos de análise.	15

Tabela 4 – Peso dos casos de uso: opção por número de classes de objetos (Fonte: KARNER, 1993).

Complexidade	Definição	Peso
SIMPLES	O caso de uso possui uma interface simples com o usuário e utiliza apenas uma entidade em um banco de dados.	5
MÉDIO	O caso de uso possui uma interface mais trabalhada e utiliza duas ou mais entidades de banco de dados.	10
COMPLEXO	O caso de uso envolve três ou mais entidades em um banco de dados e contém uma interface complexa.	15

Tabela 5 – Peso dos casos de uso: opção por classificação simples da lógica de processamento (Fonte: KARNER, 1993).

- **Calcular os PCU não ajustados:** de acordo com a seguinte fórmula:

$$\text{PCU não ajustados} = ? \text{ Pontos de Atores} + ? \text{ PCU}$$

- **Determinar o fator de complexidade técnica:** Os fatores de complexidade técnica variam numa escala de 0 a 5, de acordo com o grau de dificuldade do sistema a ser construído. O valor 0 indica pouca criticidade e baixa complexidade (irrelevante para o projeto); e o valor 5 indica muita criticidade e alta complexidade (essencial para o projeto). O questionário para identificar

o nível de influência dos fatores de complexidade técnica pode ser observado no Anexo 1. Após determinar o valor dos fatores, conforme a Tabela 6, multiplicar pelo seu respectivo peso, somar o total e aplicar a seguinte fórmula:

$$\text{Fator de Complexidade Técnica} = 0.6 + (0.01 * \sum \text{Fator Técnico}(T_i))$$

T_i	Fatores de Complexidade Técnica	Peso
T_1	Sistemas Distribuídos	2
T_2	Desempenho da aplicação	1
T_3	Eficiência do usuário final (on-line)	1
T_4	Processamento interno complexo	1
T_5	Reusabilidade do código em outras aplicações	1
T_6	Facilidade de instalação	0,5
T_7	Usabilidade (facilidade operacional)	0,5
T_8	Portabilidade	2
T_9	Facilidade de manutenção	1
T_{10}	Concorrência	1
T_{11}	Características especiais de segurança	1
T_{12}	Acesso direto para terceiros	1
T_{13}	Facilidades especiais de treinamento	1

Tabela 6 – Fatores de Complexidade Técnica (Fonte: KARNER, 1993).

- **Determinar a eficiência do fator ambiental:** Os fatores ambientais indicam a eficiência do projeto e estão relacionados ao nível de experiência dos profissionais e às condições ambientais e de trabalho.

E_i	Fatores de Complexidade Ambiental	Peso
E_1	Familiaridade com o processo formal de desenvolvimento	1.5
E_2	Trabalhadores com dedicação parcial	-1
E_3	Capacidade do líder de projeto	0.5
E_4	Experiência com a aplicação de desenvolvimento	0.5
E_5	Experiência em orientação a objetos	1
E_6	Motivação	1
E_7	Dificuldade da linguagem de programação	-1
E_8	Requisitos estáveis	2

Tabela 7 – Fatores de Complexidade Ambiental (Fonte: KARNER, 1993).

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Capítulo - 3

Esses fatores (Tabela 7) são determinados através da escala de 0 a 5, onde 0 indica baixa habilidade (pouca experiência no domínio da aplicação), 3 indica média experiência e 5 indica alta experiência. O questionário para identificar o nível de influência dos fatores de complexidade ambiental pode ser observado no Anexo 1. Após determinar o valor de cada fator, multiplicar pelo seu respectivo peso e somar o total dos valores. Em seguida, aplicar a seguinte fórmula:

$$\text{Fator de Complexidade Ambiental} = 1.4 + (-0.03 * \text{Fator Ambiental } (E_i))$$

- **Calcular os PCU ajustados:** Esse cálculo é realizado com base na multiplicação dos PCU não ajustados, na complexidade técnica e na complexidade ambiental pela seguinte fórmula:

$$\text{PCU Ajustados} = \text{PCU não ajustados} * \text{Fator de complexidade técnica} * \text{Fator de complexidade ambiental}$$

Uma vez calculados os PCU é possível determinar as estimativas de esforço, prazo e custos para o projeto de software. Além disso, medidas de produtividade, qualidade e outros atributos podem ser normalizados.

3.3.3.3 Estudo de Caso

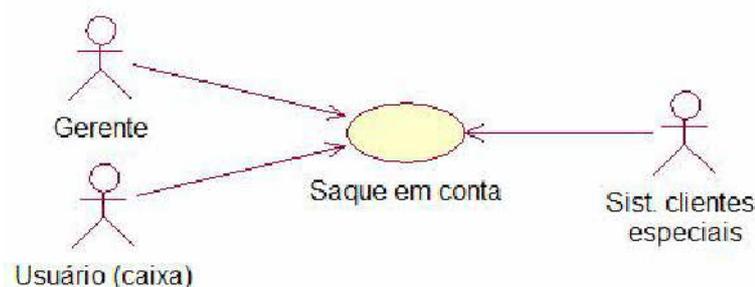


Figura 8 – Diagrama de casos de uso de um sistema de saque em conta (Fonte: COSTA, 2003).

Considere o sistema de saque em conta, conforme a Figura 8. Para estabelecer

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Capítulo - 3

as estimativas de esforço, prazo e custos é necessário determinar o tamanho do sistema, em PCU.

Para isso, deve-se identificar e multiplicar o total dos atores pelo seu respectivo peso (1, 2 ou 3), de acordo com a sua complexidade (simples, médio ou complexo) (Tabela 2). Dessa forma, é possível obter a soma total dos atores não ajustados (? Pontos de Atores).

Para o exemplo, temos que:

? Pontos de Atores = (2 atores (pessoas interagindo com o sistema) * 3 (peso)) + (1 ator (outro sistema) * 2 (peso)) = 8 pontos.

Logo após, é preciso identificar e multiplicar o total de casos de uso pelo seu respectivo peso (5, 10 ou 15), de acordo com a sua complexidade (simples, médio ou complexo), conforme a opção escolhida de classificação (Tabela 3, 4, ou 5). Dessa forma, é possível obter a soma total dos casos de uso não ajustados (? PCU).

Assumindo a opção classificação por transação (Tabela 3), temos que para o exemplo (Figura 8) existe apenas um caso de uso que, por sua vez, envolve no mínimo seis transações, como identificação e validação do cliente, validação da conta, aceite dos dados do saque, consulta ao sistema externo, liberação de dinheiro e emissão de recibo. Dessa forma, o peso do caso de uso é igual a 10, ou seja, a complexidade é média (veja Tabela 3).

? PCU = 1 * 10 (complexidade média) = 10 pontos.

Depois disso, soma-se os pontos dos atores com os PCU para determinar o total de PCU não ajustados.

PCU não ajustados = ? Pontos de Atores + ? PCU

PCU não ajustados = 8 + 10 = 18 pontos.

A seguir, é preciso determinar os fatores de complexidade técnica e ambiental.

O fator de complexidade técnica é dado pela fórmula:

Fator de Complexidade Técnica = $0.6 + (0.01 * \text{?Fator Técnico}(T_i))$

Para o exemplo, considerando o ?Fator Técnico(T_i) = 35, temos que:

Fator de Complexidade Técnica = $0.6 + (0.01 * 35) = 0,95$.

O fator de complexidade ambiental é dado pela fórmula:

Fator de Complexidade Ambiental = $1.4 + (-0.03 * \text{?Fator Ambiental}(E_i))$

Para o exemplo, considerando o ?Fator Ambiental (E_i) = 23,5, temos que:

Fator de Complexidade Ambiental = $1.4 + (-0.03 * 23,5) = 0,695$.

Finalmente, é possível calcular os PCU ajustados com base na multiplicação dos PCU não ajustados, na complexidade técnica e na complexidade ambiental pela fórmula:

PCU Ajustados = PCU não ajustados * Fator de complexidade técnica * Fator de complexidade ambiental.

Logo:

PCU Ajustados = $18 * 0,95 * 0,695 = 11,88$ pontos.

Considerando a estimativa de produtividade de 20 homens/hora por PCU, proposta por KARNER (1993), para determinar a estimativa de esforço, em homem/hora, temos que:

$11,88 \text{ PCU} * 20 \text{ horas} = 237,6 \text{ homens/hora}$.

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Capítulo - 3

Em relação à estimativa de prazo, considere o valor de 132 horas para o número de horas produtivas por mês, assumindo que cada profissional não é produtivo 8 horas/dia, devido a férias, imprevistos, doenças, etc. A estimativa de prazo, em mês, é calculada por meio da divisão do total do esforço em homem/hora pelo número de horas produtivas por mês. Portanto:

$$237,7 \div 132 = 1,8 \text{ mês.}$$

Nesse caso o prazo refere-se a 1 (um) profissional que faz todo o levantamento do sistema, modela os requisitos, analisa o sistema, projeta a arquitetura, faz o protótipo, implementa, testa e implanta o produto.

A estimativa de custos do projeto é estabelecida multiplicando-se o total do esforço em homem/hora pela média dos valores cobrados por hora de todos os profissionais do projeto. Considerando a média dos valores cobrado igual a R\$ 65,00 (valor hipotético), finalmente tem-se que:

$$237,6 * R\$ 65,00 = R\$ 15.444,00.$$

3.4 Considerações Finais do Capítulo

Este capítulo contribui na medida que dissemina um conjunto de informações necessárias à formação de uma massa de conhecimento sobre métricas de software; o que permite, e até motiva a implementação de iniciativas de utilização de medições para dominar os riscos inerentes de todo e qualquer projeto.

Nesse intuito, foi discutido sobre a necessidade de se medir com métricas de estimativa de tamanho. Um conjunto dessas métricas foi apresentado e criticado, mostrando que opções existem e são capazes de cobrir toda a forma aplicável de

desenvolvimento.

Além dessa compilação de informações sobre métricas disponíveis, antiga e moderna, procurou-se mostrar que muito tem sido feito nessa área. As pesquisas tem sido mantidas, ocasionando a reciclagem de algumas e a criação de outras, procurando atender à evolução tecnológica que a indústria de software experimenta.

Ainda no intuito de contribuir, foram avaliados três tipos de métricas de estimativas existente no mercado, a fim de que mensurar software com mais precisão se torne uma atividade possível e mais comum entre desenvolvedores. Dentre as três métricas apresentadas, ficou evidente que a estimativa de tamanho de software OO é melhor realizada por meio da métrica UCP, por proporcionar estimativas com maior precisão, sob um nível mais alto de abstração para a construção de tal produto.

Desta forma, para o contexto deste trabalho, foi sugerido o uso da métrica UCP para ser a ferramenta de estimativa do Processo de Estimativa de Software.

4 – Processo de Estimativa de Software

4.1 Introdução

Este capítulo descreve as etapas propostas para Processo de Estimativa de Software. Este processo consiste no uso dos guias PMBOK, RUP e da métrica UCP. Dessa forma, para alcançar este objetivo, é necessário realizar os seguintes passos:

- Mapear a relação entre os processos de gestão de estimativas propostos pelo PMBOK; as atividades de desenvolvimento de estimativa propostas pelo RUP; e a métrica UCP (ferramenta de estimativa do processo).
- Identificar as ações gerenciais a serem tomadas pelo gerente de projeto durante o Processo de Estimativa de Software.

A partir do Processo de Estimativa de Software, o gerente terá informações suficientes sobre as estimativas de tempo e custos do projeto, possibilitando o rastreamento do cronograma e do orçamento previstos. Dessa forma, conhecendo a situação do projeto, o gerente poderá tomar decisões de ajustes no Plano de Desenvolvimento do Software, com base nos indicadores de andamento do projeto.

4.2 Mapeamento dos processos de gestão de estimativas propostos pelo PMBOK para as atividades de desenvolvimento de estimativas propostas pelo RUP

Esta seção apresenta o mapeamento dos processos de gestão de estimativas propostos pelo PMBOK para as atividades de desenvolvimento de estimativas propostas pelo RUP.

No PMBOK, os processos de gestão de estimativas estão concentrados nas

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Capítulo - 4

áreas de conhecimento: Gerenciamento do tempo do projeto e Gerenciamento dos custos do projeto; e nos grupos de processos: Planejamento e Controle, conforme destacados no Quadro 2.

Quadro 2 -- Mapeamento dos processos de gestão de estimativas propostos pelo PMBOK (Fonte: PMBOK, 2004).

GRUPOS DE PROCESSOS / ÁREAS DE CONHECIMENTO	INICIAÇÃO	PLANEJAMENTO	EXECUÇÃO	CONTROLE	ENCERRAMENTO
4. Gerenciamento da integração do projeto	4.1 Desenvolver o termo de abertura do projeto 4.2 Desenvolver a declaração do escopo preliminar do projeto	4.3 Desenvolver o plano de gerenciamento do projeto	4.4 Orientar e gerenciar a execução do Projeto	4.5 Monitorar e controlar o trabalho do projeto 4.6 Controlar mudanças e integração	4.7 Encerrar o projeto
5. Gerenciamento do escopo do projeto		5.1 Planejar o escopo 5.2 Definir o escopo 5.3 Criar EAP		5.4 Verificar o escopo 5.5 Controlar o escopo	
6. Gerenciamento do tempo do projeto		6.1 Definir atividade 6.2 Seqüenciar as atividades 6.3 Estimar os recursos da atividade 6.4 Estimar a duração da atividade 6.5 Desenvolver o cronograma		6.6 Controlar o cronograma	
7. Gerenciamento dos custos do projeto		7.1 Estimar os custos 7.2 Desenvolver a orçamentação		7.3 Controlar os custos	
8. Gerenciamento da qualidade do projeto		8.1 Planejar a qualidade	8.2 Realizar a garantia da qualidade	8.3 Realizar o controle da qualidade	
9. Gerenciamento dos recursos humanos do projeto		9.1 Planejar os recursos humanos	9.2 Contratar ou mobilizar a equipe do projeto 9.3 Desenvolver a equipe do projeto	9.4 Gerenciar a equipe do projeto	
10. Gerenciamento das comunicações do projeto		10.1 Planejar as comunicações	10.2 Distribuir as informações	10.3 Desenvolver o relatório de desempenho 10.4 Gerenciar as partes interessadas	
11. Gerenciamento dos riscos do projeto		11.1 Planejar o gerenciamento de riscos 11.2 Identificar os riscos 11.3 Analisar qualitativamente os riscos 11.4 Analisar		11.6 Monitorar e controlar os riscos	

		quantitativamente os riscos 11.5 Planejar as respostas a riscos			
12. Gerenciamento das aquisições do projeto		12.1 Planejar compras e aquisições 12.2 Planejar contratações	12.3 Solicitar respostas de fornecedores 12.4 Selecionar fornecedores	12.5 Administrar o contrato	12.6 Encerrar o contrato

Já no RUP, as atividades de desenvolvimento de estimativas estão concentradas na disciplina Gerenciamento de Projeto, conforme destacado na Figura 9.

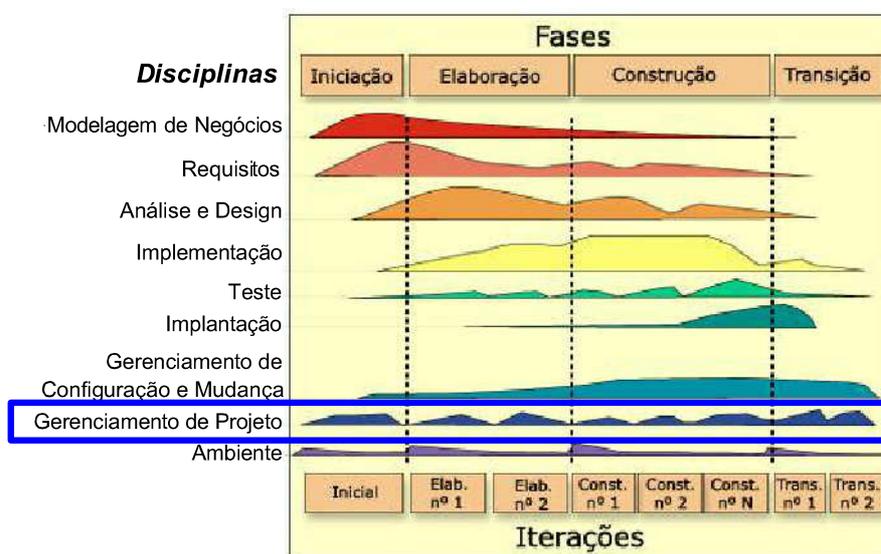


Figura 9 – Mapeamento das atividades de desenvolvimento de estimativas do RUP na disciplina Gerenciamento de Projeto (Fonte: RUP, 2002).

Dessa forma, é possível mapear os processos de gestão de estimativas propostos pelo PMBOK para as atividades de desenvolvimento de estimativas propostas pelo RUP, conforme o Quadro 3. Este mapeamento foi baseado em (CHARBONNEAU, 2004).

Quadro 3 -- Mapeamento dos processos de gestão de estimativas propostos pelo PMBOK para as atividades de desenvolvimento de estimativas propostas pelo RUP (Fonte: Adaptado de CHARBONNEAU, 2004).

PMBOK: Processos de Gestão de estimativas	RUP: Atividades de Desenvolvimento de Estimativas
➤ Estimativa de recursos da atividade	➤ Definir Equipe e Organização do Projeto
➤ Estimativa de duração da atividade	➤ Planejar Fases e Iterações ➤ Desenvolver Plano de Iteração
➤ Estimativa de custos	➤ Planejar Fases e Iterações

Note que os seguintes processos de gestão do PMBOK: Definir a atividade, Sequenciar as atividades, Desenvolver o Cronograma, Desenvolver a orçamentação, Controlar o cronograma e Controlar os custos não foram mapeados para o Processo de Estimativa de Software proposto por este trabalho. Isto se deve ao fato de que estes processos não são foco do processo técnico de executar o trabalho de estimativas. Portanto, não serão detalhados neste trabalho.

4.3 Processo de Estimativa de Software

Com base no mapeamento da seção 4.2, esta seção apresenta o Processo de Estimativa de Software. Para a ilustração do mesmo foi utilizada a representação gráfica do modelo de transformação, disponível em (SLACK, 2002).

O **modelo de transformação** é o método utilizado para representar a produção de bens e ou serviços. Qualquer operação produz bens e ou serviços e isto ocorre através do processo de transformação. Pode-se descrever toda e qualquer operação pelo sistema entrada-transformação-saída. A produção envolve vários recursos chamados de *input*, ou entrada, que são usados para transformar ou serem transformados em algo que são os *outputs*, ou saída, de bens e serviços.

O Processo de Estimativa de Software é representado graficamente pelo modelo de transformação, conforme ilustra a Figura 10.



Figura 10 – Representação gráfica do Processo de Estimativa de Software (Fonte: Adaptado de SLACK, 2002).

Segundo o PMBOK (2004), o escopo do projeto é o *input* para o Processo de Estimativa de Software. O escopo do projeto define o trabalho necessário, e apenas o trabalho necessário, para que o projeto seja concluído com sucesso. O modelo de casos de uso é o artefato gerado pelo escopo. Por meio do modelo de casos de uso e da métrica UCP é possível obter a estimativa de tamanho do software. O tamanho do software, por sua vez, irá derivar as estimativas de esforço, prazo, e custos necessários para o desenvolvimento do projeto (processo de transformação), oferecendo subsídios para distribuir as atividades entre os membros da equipe, desenvolver o cronograma e determinar o orçamento do projeto.

Contudo, antes de determinar o orçamento do projeto é necessário estimar os recursos (exemplos: pessoas, equipamentos, materiais) utilizados para a realização das atividades. Além disso, uma previsão das variações de estimativas que possam ocorrer no decorrer do projeto, como uma previsão para a inflação, deve ter sido realizada. Somado esses itens, é possível determinar a estimativa do custo global do projeto.

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Capítulo - 4

Em relação às restrições (exemplo: o custo não pode exceder R\$ 100.000,00) e prioridades (exemplo: a urgência de um produto) que estão sujeitos a maioria dos projetos, uma variedade de cenários de estimativas podem ser criados à medida que a relação entre tamanho da equipe e prazo não é linear. Dessa forma, o tamanho da equipe ou o cronograma podem ser ajustados de acordo com restrições e prioridades pré-estabelecidas. Com a variedade de cenários de estimativas possíveis, o gerente de projeto deve selecionar o cenário que melhor se adapta às necessidades do projeto.

No final do projeto, os dados obtidos (exemplos: o tamanho da aplicação em PCU, o esforço de desenvolvimento, os defeitos encontrados) devem ser registrados em uma base de dados mantida pela organização. Estes dados serão úteis para compará-los com outras medidas obtidas de projetos anteriores. Com isso, índices de produtividade, como homens/hora por PCU, R\$ por PCU; e de qualidade, como defeitos por PCU, erros por PCU, podem ser avaliados e tendências podem ser geradas.

Por fim, é preciso verificar se as estimativas realizadas no início do projeto estão dentro de um percentual de erro aceitável. Para isto, o gerente deve comparar as estimativas realizadas no início do projeto com os resultados obtidos no final do projeto (*output*). “Um percentual de erro inferior a 10% (dez por cento) é aceitável para essa relação” (TETILA *et al.*, 2006). Caso o percentual de erro seja maior que 10%, avaliações devem ser tomadas para que estimativas com maior precisão possam ser realizadas nos próximos projetos. Consistência do IP (índice de produtividade); padronização dos PCU; distribuição de atividades; e conversão de PCU em prazo, são os principais itens de avaliação.

As etapas do Processo de Estimativa de Software são descritas a seguir.

Etapa 1: Iniciar o Processo de Estimativa de Software a partir do modelo de casos de uso

Objetivo: Iniciar o Processo de Estimativa de Software utilizando como ferramenta a métrica UCP.

Significado no modelo de transformação: *input*.

Responsável: Gerente de Projeto.

Pré-requisito: O escopo do projeto deve ter sido planejado e definido num primeiro momento, e o sistema deve estar modelado pelo modelo de casos de uso.

Tarefa: Solicitar o modelo de casos de uso para o analista.

Produtos de entrada: Planejamento e definição do escopo.

Produto de saída: Modelo de casos de uso.

Diretrizes: O **modelo de casos de uso**, conforme descrito na seção 3.3.3, é uma técnica utilizada para capturar e descrever os requisitos funcionais de um sistema por meio de diagramas e descrições de casos de uso. O diagrama de casos de uso é descrito em termos de atores, casos de uso e sistema modelado. Maiores detalhes sobre a técnica de modelagem de casos de uso podem ser observados em (UML, 2005).

Etapa 2: Estimar tamanho

Objetivo: Obter o tamanho do software em PCU.

Significado no modelo de transformação: Processo de transformação.

Responsável: Gerente de Projeto.

Pré-requisito: Existência do modelo de casos de uso.

Tarefas: Realizar a estimativa de tamanho do software em PCU;

Documentar o resultado da estimativa realizada.

Produto de entrada: Modelo e descrição de casos de uso.

Produto de saída: Número de PCU estimados;

Planilhas geradas na estimativa.

Diretrizes: Conforme motivos discutidos na seção 3.3.3, a estimativa de tamanho tem sido melhor realizada por meio da métrica UCP para projetos orientados a objetos. Dessa forma, o Processo de Estimativa de Software, proposto por este trabalho, utiliza a métrica UCP para realizar as estimativas, tendo como base o modelo de casos de uso. O processo de contagem dos PCU está disponível na seção 3.3.3.

Calculado o tamanho do software, em PCU, é possível determinar as estimativas de esforço, prazo e custos do projeto. Além disso, medidas de produtividade, qualidade e outros atributos de software podem ser normalizados.

Etapa 3: Estimar esforço

Objetivo: Determinar a quantidade de trabalho previsto para a realização do projeto.

Significado no modelo de transformação: Processo de transformação.

Responsável: Gerente de Projeto.

Pré-requisito: O tamanho do software (em PCUs) já deve ter sido estimado e o IP (índice de produtividade) da equipe deve ser conhecido a partir do histórico de projetos anteriores.

Tarefa: Realizar a estimativa de esforço, em homens/hora;

Documentar o resultado da estimativa realizada.

Produto de entrada: Tamanho do software (em PCUs) e IP da equipe.

Produto de saída: Número de homens/hora estimados;

Planilhas geradas na estimativa.

Diretrizes: A estimativa de esforço do projeto é determinada pelo produto entre tamanho do software (em PCUs) e IP da equipe. Em relação ao IP da equipe, KARNER (1993) sugere o esforço de 20 homens/hora por PCU. No entanto, para obter estimativas confiáveis de esforço é necessário que a equipe conheça o seu próprio índice, já que a produtividade pode variar entre organizações e indivíduos. “Experiências demonstraram que o esforço pode variar entre 15 e 36 homens/hora por PCU” (ANDA *et al.*, 2001).

O IP da equipe pode ser obtido pela divisão do tempo real gasto para o desenvolvimento de um software pelo seu tamanho (em PCUs), estimado no início do projeto.

Etapa 4: Estimar prazo

Objetivo: Determinar a duração prevista para a realização do projeto.

Significado no modelo de transformação: Processo de transformação.

Responsável: Gerente de Projeto.

Pré-requisito: Uma abordagem que calcule a distribuição de esforço da equipe durante o projeto.

Tarefa: A partir da estimativa de esforço (em homens/hora) calcular a estimativa

de prazo (em horas ou meses);

Documentar o resultado da estimativa realizada.

Produto de entrada: Estimativa de esforço em homens/hora.

Produto de saída: Estimativa de prazo em horas ou meses;

Planilhas geradas na estimativa.

Diretrizes: Após determinar a quantidade de esforço necessário para o desenvolvimento do projeto é possível derivar a estimativa de prazo. Para derivar a estimativa de prazo é necessário uma abordagem que calcule a distribuição de esforço da equipe durante o projeto. Por exemplo, “em geral, não é possível duplicar o trabalho concluído apenas dobrando a equipe de desenvolvimento” (RUP, 2002). O modelo de custo construtivo (COCOMO, *COnstructive COst MOdel*), desenvolvido por BOEHM *et al.* (2000) fornece essa informação. Pelo mesmo é possível criar uma programação viável por meio do ajuste de vários orientadores de esforço COCOMO.

A estimativa de prazo do projeto é o principal subsídio para o desenvolvimento do cronograma e para a distribuição de atividades do projeto.

Etapa 5: Estimar recursos

Objetivo: Determinar os recursos (pessoas, equipamentos ou materiais) e as quantidades de cada recurso previstos que serão usados para realizar as atividades do projeto.

Significado no modelo de transformação: Processo de transformação.

Responsável: Gerente de Projeto.

Pré-requisito: Levantamentos de quais recursos e, em que quantidade, serão necessários para o desenvolvimento do projeto.

Tarefa: Calcular os recursos previstos para o projeto;

Documentar o resultado da estimativa realizada.

Produto de entrada: Lista de recursos previstos para a realização do projeto.

Produto de saída: Valor (em reais, dólar, euro, etc.) de recursos estimados;

Planilhas geradas na estimativa.

Diretrizes: No projeto de software, pode ser requisitado a compra de equipamentos, ou a contratação de pessoas especializadas. Por exemplo, para o desenvolvimento de um sistema comercial pode ser necessário adquirir equipamentos, como leitores de código de barra, cartões magnéticos, etc. Por outro lado, em sistemas com alto nível de regras de negócios, pode ser preciso contratar um especialista para apoiar o seu desenvolvimento. Além disso, outros recursos podem ser indispensáveis para a realização do projeto, como materiais, serviços, instalações, etc.

A estimativa de recursos é estreitamente coordenada pela estimativa de custos.

Etapa 6: Estimar custos

Objetivo: Determinar o custo global previsto para a realização do projeto.

Significado no modelo de transformação: Processo de transformação.

Responsável: Gerente de Projeto.

Pré-requisito: Estimativa do custo de desenvolvimento do software, estimativa do

custo de recursos alocados e previsão das variações de estimativas.

Tarefa: Calcular o custo estimado para desenvolver o software;

Calcular o custo estimado dos recursos alocados para desenvolver as atividades;

Calcular uma previsão para as variações das estimativas;

Somar os três itens anteriores;

Documentar os resultados das estimativas realizadas.

Produto de entrada: Estimativa do custo de desenvolvimento do software, estimativa do custo de recursos alocados e previsão para as variações das estimativas.

Produto de saída: Estimativa do custo global do projeto (em reais, dólar, euro, etc.);

Planilhas geradas na estimativa.

Diretrizes: A estimativa de custos envolve o desenvolvimento de uma aproximação dos custos necessários para a realização do projeto. Nessa aproximação, o gerente de projeto considera o custo de desenvolvimento do software; o custo de recursos alocados para a realização das atividades; e as possíveis variações das estimativas, inclusive os riscos. Somado esses itens, é possível determinar a estimativa do custo global do projeto.

O custo estimado para desenvolver o software é a soma dos produtos do esforço, em homens/hora, pela hora cobrada dos profissionais do projeto. Por exemplo, supondo que para o coordenador do projeto a hora equivale a R\$ 80,00 (valor

hipotético), e que sejam necessárias 1056 horas para a coordenação do projeto, temos que: $R\$ 80,00 * 1056 = R\$ 84.480,00$. De forma análoga, supondo que para os analistas e programadores a hora equivale a R\$ 60,00 (valor hipotético), e que sejam necessárias 4.171 horas para a análise e a programação do projeto, temos que: $R\$ 60,00 * 4.171 = R\$ 250.260,00$. Portanto, o custo estimado para desenvolver o software é: $R\$ 84.480,00 + R\$ 250.260,00 = R\$ 334.740,00$.

O custo estimado dos recursos alocados para desenvolver as atividades é determinado pelos recursos (pessoas, equipamentos ou materiais) e as quantidades de cada recurso que serão usados para realizar as atividades do projeto. Isso inclui, mas não se limita a mão-de-obra especializada, materiais, equipamentos, serviços e instalações.

Possíveis variações das estimativas precisam ser incluídas no custo do projeto. Categorias especiais, como uma previsão para inflação, ou um custo de contingência, devem ser previstos.

Caso a organização executora não possuir avaliadores formalmente treinados, a equipe do projeto precisará contratar mão-de-obra especializada para estimar o custo global do projeto.

Etapa 7: Aplicar restrições e Prioridades

Objetivo: Aplicar restrições e prioridades estabelecidas com o cliente.

Significado no modelo de transformação: Processo de transformação.

Responsável: Gerente de Projeto.

Pré-requisito: Restrições e prioridades estabelecidas com o cliente.

Tarefa: Gerar uma série de cenários de estimativas que atendam às restrições e prioridades.

Produto de entrada: Lista de restrições e prioridades.

Produto de saída: Cenários de estimativas.

Diretrizes: Em geral, projetos estão sujeitos a restrições (exemplos: devem ser entregues em um determinado prazo; o custo não pode exceder R\$ 100.000,00) ou prioridades (exemplo: a urgência de um produto). Dado um tamanho fixo de produto, eles são afetados por ajustes no tamanho da equipe. Ele reporta que o relacionamento entre o tamanho da equipe e o prazo não é linear. Portanto, devem-se usar modelos científicos para gerar uma série de cenários com base nos tamanhos de equipe variáveis. O software de estimativa automatizado é muito útil para este exercício.

Etapa 8: Selecionar a estimativa ideal de esforço, prazo e custos

Objetivo: Selecionar a estimativa ideal de esforço, prazo e custos para a realização do projeto.

Significado no modelo de transformação: Processo de transformação.

Responsável: Gerente de Projeto.

Pré-requisito: Cenários de estimativas que atenderam às restrições e prioridades estabelecidas com o cliente devem ter sido geradas.

Tarefa: Selecionar o cenário de estimativa que melhor se adapta às necessidades do projeto.

Produto de entrada: Lista de cenários de estimativas de esforço, prazo e custo.

Produto de saída: Estimativas de esforço, prazo e custos que melhor se adaptam às necessidades do projeto.

Diretrizes: Com a variedade de cenários possíveis para o projeto, o gerente deve selecionar o cenário que melhor se adapta às necessidades do projeto. Isso lhe dará uma visão inicial da duração geral, do tamanho da equipe e dos custos necessários.

Etapa 9: Atualizar a base de dados

Objetivo: Criar indicadores para avaliar os pontos fortes e fracos da organização.

Significado no modelo de transformação: Processo de transformação.

Responsável: Gerente de Projeto.

Pré-requisito: Término do projeto.

Tarefa: Registrar os dados obtidos na base de dados mantida pela organização.

Produto de entrada: Dados obtidos no final do projeto (tamanho em PCU, esforço em homens/hora, prazo em horas ou meses, etc.).

Produto de saída: Base de dados atualizada.

Diretrizes: Para obter resultados significativos, a organização deve possuir alguns dados históricos similares ao projeto. O tamanho da aplicação em PCU, o esforço de desenvolvimento, a experiência da equipe, taxas de produtividade, tipos de tecnologia e de ferramentas utilizadas, grau de reuso, são alguns exemplos.

Portanto, no final do projeto, os dados obtidos devem ser registrados na base de dados mantida pela organização. Isto permite ao gerente de projeto obter indicadores para avaliar os pontos fortes e fracos da organização, como índices de produtividade (homens/hora por PCU, R\$ por PCU) e de qualidade (defeitos por

PCU, erros por PCU).

Etapa 10: Avaliar estimativas

Objetivo: Verificar se estimativas realizadas no início do projeto estão dentro de um percentual de erro aceitável.

Significado no modelo de transformação: *output*

Responsável: Gerente de Projeto

Pré-requisito: As estimativas realizadas no início do projeto devem estar documentadas;

Os dados obtidos no final do projeto devem ter sido registrados.

Tarefa: Calcular o percentual de erro das estimativas iniciais com os resultados obtidos no final do projeto.

Produto de entrada: Estimativas realizadas no início do projeto;

Resultados obtidos no final do projeto.

Produto de saída: Estimativas realizadas no início do projeto inferior ou superior ao percentual de erro.

Diretrizes: As estimativas realizadas no início do projeto devem ser comparadas com os resultados obtidos no final do projeto para avaliar o percentual de erro das estimativas. Um percentual de erro inferior a 10% (dez por cento) é aceitável para essa relação. Caso o erro seja maior que 10%, avaliações devem ser tomadas para que estimativas com maior precisão possam ser realizadas nos próximos projetos.

Quatro fatores principais devem ser considerados:

➤ **Consistência do IP (índice de produtividade):** a estimativa do IP de 20

homens/hora por PCU, proposta por KARNER (1993), pode variar entre organizações e indivíduos. Dessa forma, a equipe deve conhecer o seu IP para obter estimativas confiáveis de esforço. Eventualmente, pode haver mudanças no IP da equipe ao longo do tempo em razão de fatores como: adesão de novas ferramentas, mudança de pessoal, especificidade técnica, etc. Portanto, deve ser calculado o IP no final de um projeto para verificar a sua consistência. Caso o percentual de erro entre o IP do projeto e o IP da equipe seja maior que 10%, a média dos IPs dos três últimos projetos deve ser utilizada.

- **Padronização dos PCU:** a variação nos tipos dos casos de uso é resultado da falta de padrões entre dois tipos de casos de usos: incluídos e estendidos. O comportamento comum é executado pelos casos de usos incluídos e os opcionais, pelos casos de uso estendidos. O relacionamento tipo “incluído” é usado para evitar a cópia do mesmo texto em fluxos alternativos de diversos casos de uso. O caso de uso “estendido” indica que uma instância do caso de uso pode ou não incluir o comportamento especificado no outro caso de uso. No entanto, não há regras precisas para a contagem dos casos de uso incluídos e estendidos. KARNER (1993) não recomenda a contagem dos casos de uso estendidos e incluídos. Por outro lado, ANDA *et al.* (2001) acham que estes devem ser incluídos na contagem para evitar estimativa abaixo da realidade. Dessa forma, a obtenção de estimativas confiáveis de esforço exige a padronização dos tipos de casos de uso pela organização.
- **Distribuição de atividades:** o diagrama de atividades, no modelo de casos de uso, é o *input* para a distribuição do fluxo de trabalho pelas atividades. No

entanto, a má distribuição de recursos entre atividades e sub-atividades do projeto pode refletir numa previsão de cronograma e custos equivocados. Nesse sentido, o gerente de projeto deve realizar a revisão da distribuição de atividades e ou a revisão do diagrama de atividades para corrigir a ocorrência de erros futuros.

- **Conversão de esforço em prazo:** dado o esforço estimado, em homens/hora por PCU, é necessária uma abordagem que calcule a relação entre a distribuição de esforço da equipe e a estimativa de prazo do projeto. O COCOMO é, provavelmente, o modelo mais adequado para essa solução. No entanto, por ser matematicamente complexo, o modelo pode admitir certos erros de cálculos, caso o profissional que os fez não possua experiência com os vários orientadores de esforço COCOMO. Uma boa alternativa para essa questão é a utilização de softwares de estimativas automatizados.

Todavia, caso os quatro fatores principais anteriormente citados estejam satisfeitos; e o percentual de erro entre as estimativas iniciais e os resultados obtidos no final do projeto seja maior que 10%. Um método de avaliação e melhoria de estimativa deve ser proposto para o Processo de Estimativa de Software, a fim de obter estimativas com maior precisão.

4.5 Considerações Finais do Capítulo

Este capítulo apresentou um Processo de Estimativa de Software para padronizar o trabalho de executar estimativas. O processo proposto foi definido com base nos guias PMBOK, RUP e na métrica UCP. A partir do mapeamento da relação

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Capítulo - 4

entre os guias e a métrica, foram definidas as etapas do processo. Essas etapas foram representadas graficamente pelo modelo de transformação: *input*-transformação-*output*. Dessa forma, foram identificadas as ações gerenciais a serem tomadas pelo gerente de projeto em cada etapa do Processo de Estimativa de Software.

Foram mapeados os processos de gestão de estimativas propostos pelo PMBOK para as atividades de desenvolvimento de estimativas propostas pelo RUP. Verificou-se, com isso, que os mesmos estão em consonância em seus aspectos essenciais, não havendo nenhuma incompatibilidade fundamental entre eles. Termos diferentes são usados, no entanto, não se contradizem, descrevendo conceitos semanticamente similares ou idênticos.

A união dos guias e da métrica UCP significou uma sinergia na elaboração do Processo de Estimativa de Software: o PMBOK servindo como um guia para descrever as práticas de gestão de estimativas; o RUP servindo como um *framework* para atender as características do desenvolvimento de estimativas; e a métrica UCP atuando como ferramenta para o trabalho de executar estimativas.

5 – Pesquisa-ação

5.1 Introdução

Este capítulo apresenta a pesquisa-ação realizada com três projetos de software de duas conceituadas instituições brasileiras, sendo dois projetos de uma instituição pública e um projeto de uma instituição privada. Os objetivos deste capítulo são: (1) mostrar os conceitos (2) experimentar as ações do processo e (3) relatar a experiência. Para realizar esta pesquisa, foram definidas as etapas da pesquisa-ação, descritas na seção 5.2.

5.2 Etapas da Pesquisa-ação

Para realizar a pesquisa-ação foram definidos alguns conjuntos de ações que, embora não ordenados no tempo, podem ser considerados como etapas da pesquisa-ação (GIL, 2002). São elas:

5.2.1 Seleção da amostra

Tão logo tenha sido delimitado o universo da pesquisa, surge o problema de determinar os elementos que serão pesquisados.

Quando o universo de investigação é geograficamente concentrado e pouco numeroso, convém que sejam pesquisados todos os elementos. Isto é importante para garantir a conscientização e a mobilização da população em torno da proposta de ação envolvida pela pesquisa. Quando, porém o universo é numeroso e esparso, como é o caso dos projetos de software, é recomendável à seleção de uma amostra.

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Capítulo - 5

Isso não significa, no entanto, que a amostra deva ser selecionada de acordo com procedimentos rigidamente estatísticos, pois estes poderiam neutralizar o efeito de conscientização que é pretendido nesse tipo de investigação. De modo geral, o mais recomendável nas pesquisas desse tipo é a utilização de amostras não probabilísticas, selecionadas pelo critério de intencionalidade (GIL, 2002).

Uma amostra intencional é aquela em que os indivíduos são selecionados com base em certas características tidas como relevantes pelos pesquisadores e participantes, mostrando-se mais adequada para a obtenção de dados de natureza qualitativa; o que é o caso da pesquisa-ação (GIL, 2002).

Nesse sentido, assumindo o critério de intencionalidade, foram selecionados três projetos de software desenvolvidos por duas instituições brasileiras, uma pública e outra privada, em que gerentes de projetos utilizaram as métricas *Use Case Points* (UCP) e *Function Points* (FP) em suas estimativas. Esse critério de intencionalidade foi utilizado para efeito de comparação entre métricas de estimativas tradicionais (UCP e FP) e Processo de Estimativa de Software, sugerido pelo trabalho. Porém, tanto a instituição pública quanto a privada não autorizaram a divulgação de seus nomes, nem de seus projetos, por participarem de concorrências e licitações públicas.

5.2.2 Coleta de dados

Diversas técnicas são adotadas para a coleta de dados na pesquisa-ação. A mais usual é a entrevista aplicada coletiva ou individualmente. Também se utilizam outras técnicas, como o questionário, o formulário, a observação participante, a história de vida, a análise de conteúdo e o sociodrama (GIL, 2002).

No contexto particular da pesquisa-ação, questionários e formulários não são

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Capítulo - 5

suficientes em si mesmos. Eles trazem informações sobre o universo considerado, nos quais serão analisados e discutidos em reuniões e seminários com a participação de pessoas representativas. O processamento estatístico das respostas, com computadores ou não, nunca é suficiente. O processamento adequado sempre requer uma função argumentativa dando relevo e conteúdo social às interpretações (THIOLLENT, 2004).

Dessa forma, a coleta de dados realizada neste trabalho alia o uso de questionários com a experiência pessoal dos participantes do estudo. Os participantes do estudo são os gerentes de projetos das instituições e o aluno Everton Castelhão Tetila do curso de Pós-Graduação em Engenharia de Produção da UNIP, cujos perfis são de profissionais que trabalham na área de TI. Os questionários utilizados na coleta de dados estão disponíveis nos Apêndices.

5.2.3 Análise e interpretação dos dados

A análise e interpretação dos dados na pesquisa-ação constituem tema bastante controverso. Há pesquisas em que os procedimentos adotados são muito semelhantes aos da pesquisa clássica, o que implica considerar os passos: categorização, codificação, tabulação, análise estatística e generalização. Há, porém, pesquisas em que se privilegia a discussão em torno dos dados obtidos, de onde decorre a interpretação de seus resultados. Dessa discussão participam pesquisadores, participantes e especialistas convidados (GIL, 2002).

Para a análise e interpretação dos dados, este trabalho associa alguns passos adotados na pesquisa clássica à discussão em torno dos dados obtidos pelos questionários. Os seguintes passos da pesquisa clássica: tabulação, análise estatística

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Capítulo - 5

e generalização, estão disponíveis nas seções 5.2.5.1 e 5.2.5.2. Já a discussão informal, realizada entre os participantes do estudo, está disponível na seção 5.2.5.3.

5.2.4 Plano de ação

O campo de atuação desta pesquisa beneficia as seguintes populações: as instituições que desenvolvem software; os profissionais de software; e os clientes.

Acredita-se que a relação entre clientes, instituições e profissionais que desenvolvem software irá melhorar sensivelmente a partir da proposta deste trabalho. Contudo, algumas medidas foram identificadas para contribuir com a melhoria do Processo de Estimativa de Software:

- Mapear os processos de gestão propostos pelo PMBOK: Controlar o cronograma e Controlar os custos, para as atividades de gestão de estimativas propostas pelo RUP.
- Utilizar um método de avaliação e melhoria de estimativas para descobrir não-conformidades nos processos utilizados pela organização.

Em breve, este trabalho estará disponível no *site* da Universidade Paulista - UNIP no seguinte endereço eletrônico: http://www3.unip.br/ensino/pos_graduacao/strictosensu/ss_eng_producao_t2007.aspx. Além disso, o trabalho foi enviado via e-mail para as principais instituições de software do país, com base em (MCT, 1997). A listas das instituições de software que receberam o trabalho está disponível no Anexo 2. Espera-se, com isso, tornar a pesquisa pública no mercado e apoiar os gerentes a tomarem melhores decisões gerenciais durante o ciclo de desenvolvimento de projetos. Possíveis sugestões oriundas das instituições serão relevantes para a

continuidade e melhoria da pesquisa.

5.2.5 Divulgação dos resultados

A informação obtida na pesquisa-ação pode ser divulgada externamente aos setores interessados, por intermédio de congressos, conferências, simpósios, meios de comunicação de massa ou elaboração de relatórios com as mesmas formalidades dos outros tipos de pesquisa (GIL 2002).

Este trabalho faz uso de Relatório para a divulgação dos resultados. O relatório corresponde aos seguintes itens: apresentação dos dados; análise e interpretação; e discussão dos resultados.

5.2.5.1 Apresentação dos dados

Três formas de aplicação de métricas de estimativas foram propostas para estimar o custo de cada projeto de software. Primeiro, cada projeto foi estimado utilizando a métrica *Function Points*. Logo após, cada projeto foi estimado utilizando a métrica *Use Case Points*. E, finalmente, cada projeto foi estimado utilizando o Processo de Estimativa de Software proposto por este trabalho. As estimativas iniciais de cada uma das métricas utilizadas foram comparadas com os resultados obtidos nos finais dos projetos. Assim, o percentual de erro entre a estimativa inicial e resultado obtido foi estabelecido. Os erros calculados para cada uma das métricas utilizadas foram comparados entre si. Isto permitiu compreender melhor o comportamento de cada projeto e avaliar o desempenho das métricas utilizadas.

Os questionários utilizados para estimar os custos do *Projeto 1*, desenvolvido pela Instituição privada, estão disponíveis nos apêndices A, B e C. E os questionários

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Capítulo - 5

utilizados para estimar os custos dos outros dois projetos: *Projeto 2* e *Projeto 3*, realizados pela instituição pública, estão disponíveis nos apêndices D, E, F e G, H, I, respectivamente.

5.2.5.2 *Análise e interpretação*

As tabulações dos dados dos três projetos pesquisados estão disponíveis nos apêndices J, K e L. Tais apêndices serão discutidos a seguir.

O apêndice J compara as três formas de aplicação de métricas de estimativas para o *Projeto 1*, da Instituição privada. A métrica *Function Points* obteve no projeto as seguintes estimativas: **esforço:** 5.227,2 homens/hora; **prazo:** 6 meses e **custo:** R\$ 334.752,00. Já a métrica *Use Case Points* obteve no projeto as seguintes estimativas: **esforço:** 5.542,6 homens/hora; **prazo:** 6,2 meses e **custo:** R\$ 354.950,00. Finalmente, o Processo de Estimativa de Software obteve no projeto as seguintes estimativas: **esforço:** 5.542,6 homens/hora; **prazo:** 6,2 meses e **custo:** R\$ 365.402,00.

Os resultados obtidos no final do projeto foram: **esforço:** 5.644 homens/hora; **prazo:** 6,25 meses e **custo:** R\$ 371.580,00. Dessa forma, é possível estabelecer o percentual de erro para cada uma das métricas utilizadas. A Figura 11 ilustra os erros de custos obtidos pelas métricas *Function Points*, *Use Case Points* e Processo de Estimativa de Software, no *Projeto 1*.

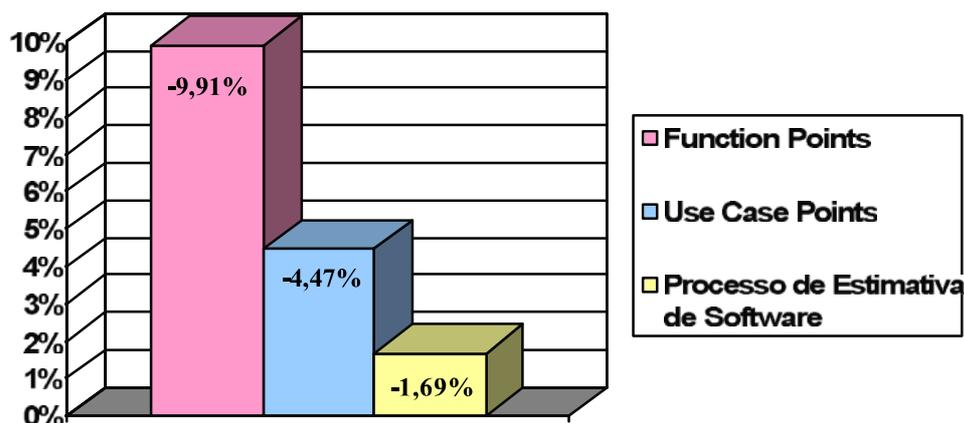


Figura 11 – Erros de custos obtidos pelas métricas *Function Points*, *Use Case Points* e Processo de Estimativa de Software, no Projeto 1.

Os erros de custo obtidos pelas métricas tradicionais (FP e UCP) foram de -9,91% e -4,47%, respectivamente. Portanto, são aceitáveis para a relação entre estimativas iniciais e resultados obtidos “um percentual de erro inferior a 10% (dez por cento) é aceitável para essa relação” (TETILA *et al.*, 2006). Todavia, o erro calculado pode ser expressivo quando se trata de um projeto de grande porte. Isto porque as estimativas iniciais realizadas pelas métricas FP e UCP foram inferiores ao custo final do projeto em R\$ 36.828,00 e R\$ 16.630,00, respectivamente.

O erro obtido pelo Processo de Estimativa de Software foi de -1,69%. Isto, em termos de custo, representa R\$ 6.178,00. Porém, o percentual de erro poderia ser ainda menor, já que ocorreram faltas de membros da equipe que não estavam previstas no planejamento do projeto. Fato que desencadeou um esforço de desenvolvimento ligeiramente acima do esperado. Consulte o Apêndice J.

É razoável supor que o Processo de Estimativa de Software obteve menor erro que as métricas tradicionais (FP e UCP) porque este considera a variação das estimativas (exemplo: inflação) ao longo do desenvolvimento do projeto, ao contrário das métricas tradicionais, que estimam somente o custo do

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Capítulo - 5

desenvolvimento do software. A inflação no período do desenvolvimento do *Projeto* foi de 2,94% (5,7% ao ano). Veja a Tabela 8.

Ano	Inflação ao ano (%)
2004	7,6
2005	5,7
2006	3,1
2007	4,1

Tabela 8 – Perspectivas para a inflação (Fonte: BCB, 2006).

O apêndice K compara de forma análoga, as três formas de aplicação de métricas de estimativas para o *Projeto 2*, da Instituição Pública. A métrica *Function Points* obteve no projeto as seguintes estimativas: **esforço:** 12.058 homens/hora; **prazo:** 8,65 meses e **custo:** R\$ 595.544,00. Já a métrica *Use Case Points* obteve as seguintes estimativas: **esforço:** 11.553,6 homens/hora; **prazo:** 8,5 meses e **custo:** R\$ 570.632,00. Finalmente, o Processo de Estimativa de Software obteve as seguintes estimativas: **esforço:** 11.553,6 homens/hora; **prazo:** 8,5 meses e **custo:** R\$ 588.078,00.

Os resultados obtidos no final do projeto foram: **esforço:** 11.840 homens/hora; **prazo:** 8,6 meses e **custo:** R\$ 603.730,00. Dessa forma, é possível estabelecer o percentual de erro para cada uma das métricas utilizadas. A Figura 12 ilustra os erros de custos obtidos pelas métricas *Function Points*, *Use Case Points* e Processo de Estimativa de Software, no *Projeto 2*.

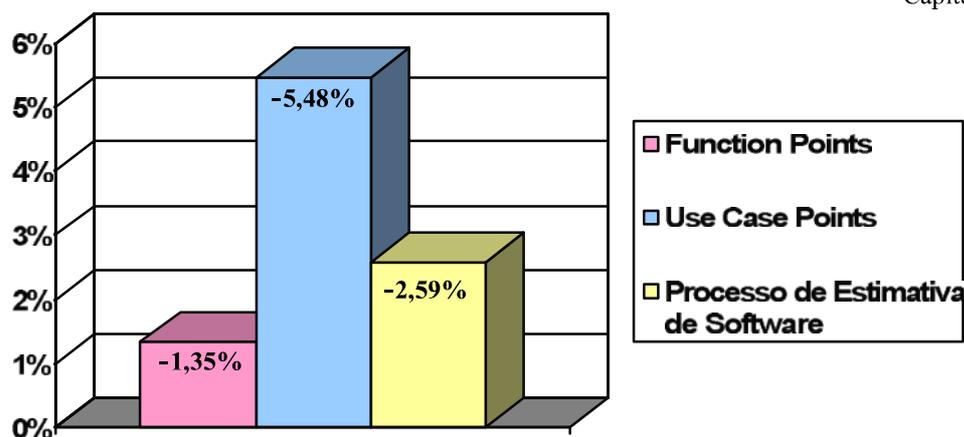


Figura 12 – Erros de custos obtidos pelas métricas *Function Points*, *Use Case Points* e Processo de Estimativa de Software, no Projeto 2.

Os erros de custo obtido pelas métricas tradicionais (FP e UCP) foram de -1,35% e -2,59%, respectivamente. O erro de custo obtido pelo Processo de Estimativa de Software foi de -2,59%. Portanto, os percentuais de erros das três métricas utilizadas foram inferiores ao limite máximo de 10% aconselhável em (TETILA *et al.*, 2006).

No *Projeto 2*, o esforço previsto pelas métricas UCP e Processo de Estimativa de Software, estiveram ligeiramente abaixo da realidade, o que acarretou estimativas menos precisas. Isto ocorreu porque o Índice de Produtividade (IP) da equipe foi um pouco superior ao IP de 20 homens/hora por PCU proposto por Karner (KARNER, 1993).

Contudo, o erro de custo obtido pelo Processo de Estimativa de Software foi menor que o erro de custo obtido pela métrica UCP. Dois fatores foram determinantes para isso: (1) o Processo de Estimativa de Software considera a variação das estimativas ao longo do projeto: a inflação no período foi de 3,5%; (2) o Processo de Estimativa de Software prevê o custo dos recursos alocados: ocorreu treinamento em J2EE (*Java 2 Enterprise Edition*) durante a realização do projeto.

Consulte o Apêndice K.

Finalmente, o apêndice L compara as três formas de aplicação de métricas de estimativas para o *Projeto 3*, da Instituição Pública. A métrica *Function Points* obteve no projeto as seguintes estimativas: **esforço**: 9.539 homens/hora; **prazo**: 7,91 meses e **custo**: R\$ 510.050,00. Já a métrica *Use Case Points* obteve no projeto as seguintes estimativas: **esforço**: 10.760 homens/hora; **prazo**: 8,3 meses e **custo**: R\$ 575.337,00. Finalmente, o Processo de Estimativa de Software obteve no projeto as seguintes estimativas: **esforço**: 10.760 homens/hora; **prazo**: 8,3 meses e **custo**: R\$ 609.266,00.

Os resultados obtidos no final do projeto foram: **esforço**: 10.384 homens/hora; **prazo**: 8,16 meses e **custo**: R\$ 594.640,00. Dessa forma, é possível estabelecer o percentual de erro para cada uma das métricas utilizadas. A Figura 13 ilustra os erros de custos obtidos pelas métricas *Function Points*, *Use Case Points* e Processo de Estimativa de Software, no *Projeto 3*.

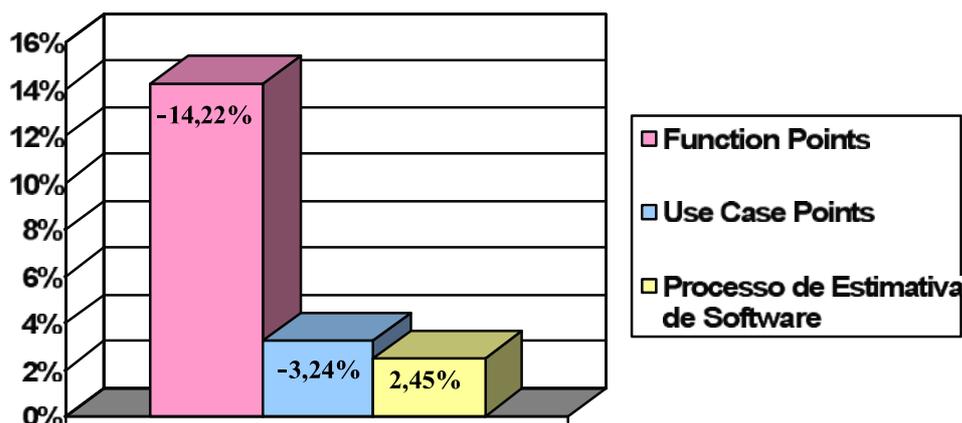


Figura 13 – Erros de custos obtidos pelas métricas *Function Points*, *Use Case Points* e Processo de Estimativa de Software, no *Projeto 3*.

Os erros de custos obtidos pelas métricas UCP e Processo de Estimativa de Software foram de -3,24% e 2,45%, respectivamente. Logo, os percentuais de erros

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Capítulo - 5

são inferiores ao limite máximo de 10% aconselhável em (TETILA *et al.*, 2006).

Por outro lado, o erro de custo obtido pela métrica FP foi de -14,22%, ou seja, erro superior ao aconselhado (valor absoluto). Neste caso, quatro itens deveriam ser avaliados para que estimativas com maior precisão possam ser realizadas nos próximos projetos. Os itens de avaliação são: consistência do IP; padronização dos PCU; distribuição de atividades; e conversão de PCU em prazo. A Etapa 10 do Processo de Estimativa de Software: Avaliar Estimativas (seção 4.3) descreve cada item de avaliação.

Em relação ao erro de custo obtido pela métrica FP, três fatores foram determinantes para que esse erro fosse maior que o recomendado: (1) o esforço estimado ficou bem abaixo da realidade: o IP da equipe foi superior ao IP utilizado na métrica; (2) a métrica FP não considera a variação das estimativas ao longo do projeto: durante o desenvolvimento do projeto houve reajustes salariais de três membros da equipe, além disso, a estimativa de custo deveria ser reajustada em relação à inflação do período, que foi de 3,1%; (3) a métrica FP não prevê o custo dos recursos alocados: para a realização do projeto, foi necessária a aquisição da ferramenta *Rational Functional Tester*. Consulte o Apêndice L.

5.2.5.3 *Discussão dos resultados*

Três formas de aplicação de métricas foram propostas para estimar o custo de cada projeto de software. Isto permitiu comparar o desempenho do Processo de Estimativa de Software com o desempenho das métricas tradicionais (FP e UCP). Esta seção discute os resultados analisados na seção anterior (5.2.5.2).

No *Projeto 1*, da Instituição privada, presume-se que o Processo de

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Capítulo - 5

Estimativa de Software obteve menor percentual de erro que as métricas tradicionais (FP e UCP) porque este considerou a inflação do período ao longo do desenvolvimento do projeto, ao contrário das métricas FP e UCP.

O índice de inflação mede, entre outras coisas, a variação geral dos preços e do custo de vida. No período de desenvolvimento do *Projeto 1* a inflação foi de 2,94% (5,7% ao ano). Esse índice foi importante para calcular a variação dos preços dos bens consumidos (exemplos: aluguel, luz, água, telefone, funcionários) ao longo do projeto.

Já no *Projeto 2* da Instituição Pública, o percentual de erro da métrica FP foi menor que o percentual de erro do Processo de Estimativa de Software. Isto porque o esforço estimado pela métrica FP foi mais preciso que o esforço estimado pelo Processo de Estimativa de Software. Todavia, o percentual de erro do Processo de Estimativa de Software foi menor que o percentual de erro da métrica UCP, à medida que o primeiro considerou tanto a inflação do período (3,5% ao ano) quanto o treinamento (*J2EE*) que ocorreram ao longo do projeto.

Por fim, no *Projeto 3* da Instituição Pública é lícito supor que o Processo de Estimativa de Software obteve menor erro que as métricas tradicionais (FP e UCP) porque este considerou, além da inflação do período (3,1%), o reajuste salarial de membros da equipe.

Na ocasião, ocorreu o reajuste salarial de três membros da equipe devido ao plano de carreira da instituição. Um dos membros da equipe obteve o reajuste salarial de 25% por ter completado um ano de trabalho. Os outros dois membros da equipe obtiveram os reajustes de 12,5% por terem completado três anos trabalhados. Os

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Capítulo - 5

reajustes, no entanto, não ocorreram no mesmo mês para cada membro da equipe. Situação em que o salário de cada membro foi calculado separadamente. Em média, os salários dos membros obtiveram um reajuste de 20,83%, ao longo de 4,7 meses do projeto, o equivalente a R\$ 15.780,00 de aumento salarial.

Em relação aos projetos pesquisados, é possível concluir que o erro médio obtido pelo Processo de Estimativa de Software é de 2,24%. Logo, a sua precisão média é de 97,76%. Pode-se afirmar com 95% de confiança que o erro percentual médio do Processo de Estimativa de Software não deva sair do intervalo [1,04%; 3,45%]. Veja a Tabela 9. O Processo de Estimativa de Software com UCP é 4,3% mais preciso que as métricas de estimativas tradicionais (FP e UCP) sem processo formal de estimativas.

Métrica	N	Média	Desvio Padrão	Erro Padrão	Intervalo de confiança para a média de 95%		Mínimo	Máximo
					Limite superior	Limite inferior		
FP	3	8,4933	6,55091	3,78217	-7,7800	24,7667	1,35	14,22
UCP	3	4,3967	1,12180	0,64767	1,6100	7,1834	3,24	5,48
PES	3	2,2433	0,48429	0,27960	1,0403	3,4464	1,69	2,59
Total	9	5,0444	4,31997	1,43999	1,7238	8,3651	1,35	14,22

Tabela 9 – Valor absoluto do percentual de erro obtido pelas métricas *Function Points*, *Use Case Points* e Processo de Estimativa de Software nos projetos 1, 2 e 3.

Admite-se, porém, que o considerável índice de precisão das estimativas obtido pelo Processo de Estimativa de Software pode ter sido influenciado pelas informações obtidas pelos gerentes de projetos. O Processo de Estimativa de Software foi o único a ser realizado após a conclusão dos projetos. Isto permitiu calcular com exatidão todos os recursos consumidos (exemplos: treinamento, aquisição de ferramenta, reajuste salarial) ao longo do projeto. A situação ideal, no

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Capítulo - 5

entanto, seria realizar as estimativas nas fases iniciais dos projetos. Nessa situação, certamente tais recursos não seriam previstos com exatidão, ou ainda, poderiam ter sido negligenciados na fase de planejamento do projeto, o que diminuiria a precisão das estimativas realizadas pelo Processo de Estimativa de Software.

5.3 Considerações Finais do Capítulo

Este capítulo apresentou a pesquisa-ação realizada com três projetos de software de duas instituições brasileiras, uma pública e outra privada. Isto foi fundamental para comparar o desempenho do Processo de Estimativa de Software com o desempenho das métricas tradicionais (FP e UCP). Os resultados da pesquisa-ação demonstram que, em média, o Processo de Estimativa de Software com UCP é 4,3% mais preciso que as métricas de estimativas tradicionais (FP e UCP) sem processo formal de estimativas.

Durante a realização da pesquisa, a maior dificuldade encontrada foi localizar instituições de software que tivessem dados históricos de projetos anteriores. A maioria das instituições não possuía qualquer tipo de métrica para realizar as medições. Outras, porém, não registravam as medições quando as realizavam.

Uma razão para isso, é que os benefícios de se utilizar métricas de estimativas não são bem definidos, isto por causa da falta de padrões quando da aplicação das estimativas. Não existe, até onde se sabe, um guia que seja capaz de avaliar o percentual de erro das medições calculadas e que oriente a melhoria das estimativas ao longo do tempo. Com isso, muitos profissionais não conseguem analisar as tendências das medições e identificar os pontos fortes e fracos de uma organização.

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Capítulo - 5

Uma outra razão, é que em muitas organizações, os processos de desenvolvimento de software ainda estão organizados de forma precária, não estando preparados para fazer uso de medições.

Em relação aos resultados obtidos nos projetos, três elementos foram fundamentais para o aumento da precisão nas estimativas: (1) a previsão das variações de estimativas, que foi fundamental para calcular a variação geral dos preços ao longo do projeto; (2) a estimativa dos recursos: que conseguiu prever os preços dos recursos alocados para a realização das atividades; (3) o IP, que, quando consistente, garantiu que a estimativa de esforço fosse mais precisa, ocasionando menor percentual de erro nas medições.

6 – Conclusões e Perspectivas Futuras

Este trabalho apresentou um Processo de Estimativa de Software para padronizar o trabalho de executar estimativas. O Processo de Estimativa de Software foi definido com base nos guias PMBOK, RUP e na métrica UCP. A partir do cruzamento dos guias com a métrica UCP, verificou-se que ambos são capazes de se relacionar, podendo ser usados de forma combinada.

Uma pesquisa-ação foi proposta para mostrar os conceitos, experimentar as ações do Processo de Estimativa de Software e relatar a experiência. Os resultados demonstraram que o Processo de Estimativa de Software é, em média, 4,3% mais preciso que as métricas de estimativas tradicionais (FP e UCP).

Espera-se, com isso, que o Processo de Estimativa de Software apresentado venha a contribuir com o desenvolvimento de projetos de software dentro de um cronograma e orçamento previsíveis e com um nível de qualidade adequada.

Como trabalhos futuros, este trabalho visa adaptar ações da gestão de estimativas ao Processo de Estimativa de Software apresentado. Para isto, será necessário mapear os processos de gestão propostos pelo PMBOK: Controlar o cronograma e Controlar os custos, para as atividades de gestão de estimativas propostas pelo RUP. Além disso, um método de avaliação e melhoria de estimativas para descobrir não-conformidades nos processos utilizados pela organização seria útil ao processo.

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Referências Bibliográficas

Referências Bibliográficas

AGUIAR, M. Pontos de função ou pontos de caso de uso? Como estimar projetos orientados a objetos. **Developers' Magazine**, [S.l.], v. 7, n. 77, jan. 2003.

ANDRADE, M. M. **Introdução à Metodologia do Trabalho Científico**. 6. ed., São Paulo: Atlas, 2003. 174 p.

ANDA, B. Comparing effort estimates based on use case points with expert estimates. **Empirical Assessment in Software Engineering (EASE 2002)**, Keele, Apr. 8-10, 2002.

_____. et al. Estimating software development effort based on use cases: experiences from industry. In: INTERNATIONAL CONFERENCE ON THE UNIFIED MODELING LANGUAGE (UML2001), 4, 2001. Toronto. **Proceedings**. London: Springer-Verlag, 2001.

BASIL, V. R.; CALDIERA, G.; ROMBACH, H. D. The goal question metric approach. [S.l.]: John Wiley & Sons, 1994. p. 528-532. (Encyclopedia of Software Engineering, 2).

BCB. BANCO CENTRAL DO BRASIL. Setor externo, Inflação e Atividade: Perspectivas para 2006. **Banco Central do Brasil**. 06 abr. 2006. 2006. Disponível em: <<http://www.bcb.gov.br/?APRES2006012>>. Acesso em: 30 mai. 2007.

BOEHM, B. W.; ABTS C., BROWN, A. W.; CHULANI, S.; CLARK, B. K.; HOROWITZ, E.; MADACHY, R.; REIFER, D.; STEECE, B. **Software Cost Estimation whith COCOMO II**. New Jersey: Prentice-Hall, 2000.

BRAGA, A. **Análise de pontos por função**. Rio de Janeiro: Infobook, 1996.

CALDIERA, G. et al. Definition and experimental evaluation of function points for object-oriented systems. In: INTERNATIONAL SYMPOSIUM ON SOFTWARE METRICS, 5, 1998. Bethesda. **Proceedings**. Washington: IEEE Computer Society, 1998.

CHARBONNEAU, S. Software Project Management - A Mapping between RUP and the PMBOK. **Xelation Software Corporation**, [S.l.], May. 2004. Disponível em: <<http://www.developerworks/rational/library/4721.htm>>. Acesso em: 17 ago. 2006.

CMMI. CAPABILITY MATURITY MODEL INTEGRATION. **CMMI for development: CMMI-DEV**. Pittsburgh: Software Engineering Institute, v. 1.2, 2006. 573 p.

COSTA, I. **Contribuição para o aumento da qualidade e produtividade de uma fábrica de software através da padronização do processo de recebimento de serviços de construção de software**. 2003. 203 f. Tese (Engenharia de Produção) - Universidade de São Paulo, São Paulo, 2003.

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Referências Bibliográficas

DAMODARAN, M.; WASHINGTON, A. Estimation using use case points. **University of Houston-Victoria**, Victoria, 2002.

DEKKERS, C. A. Desmistificando pontos de função: entendendo a terminologia. **IT Metrics Strategies**, [S.l.], out. 1998.

DIETZ, T. Methods for analyzing data from delphi panels: some evidence from a forecasting study. **Technological Forecasting and Social Change**, New York, v. 31, p. 79-85, 1987.

FENTON, N. E.; PFLEEGER, S. L. **Software Metrics: A rigorous and practical approach**. 2nd, Boston: PWS Publishing Company, Feb. 1998. 656 p.

FERNANDES, A. A. **Gerência de software através de métricas**. São Paulo: Atlas, 1995. 421 p.

FREIRE, H. Calculando estimativas: o método de pontos de caso de uso. **Developer's Magazine**, [S.l.], n. 78, fev. 2003.

GIL, A. C. **Como elaborar projetos de pesquisa?** 4.ed., São Paulo: Atlas, 2002. 171 p.

HALTEAD, M. H. Elements of software science. New York: **Elsevier Computer Science Library**, 1977.

HAZAN, C. Análise de pontos de função: uma abordagem gerencial. In: CONGRESSO NACIONAL DA SBC, 2000. Curitiba. **Jornada de Atualização em Informática**. Curitiba: PUCPR, 2000.

IFPUG. INTERNATIONAL FUNCTION POINT USERS GROUP. **Function point counting practices manual**. Michigan: International Function Point Users Group, Release 4.2, Jan. 2004.

JONES, C. **Applied software measurement: assuring productivity and quality**. 2nd, New York: McGraw-Hill, 1996.

KARNER, G. Use Case Points: resource estimation for Objectory projects. **Objective Systems SF AB**, [S.l.], Set. 1993.

LONGSTREET, D. Fundamentals of function point analysis. **Longstreet Consulting Online**, Blue Springs, 2002. Disponível em: <<http://www.ifpug.com/fpafund.htm>>. Acesso em: 1 ago. 2005.

_____. Function points training and analysis manual. **Longstreet Consulting Online**, Blue Springs, Oct. 2004. Disponível em: <<http://www.softwaremetrics.com/freemanual.htm>>. Acesso em: 23 set. 2005.

MCCABE, T. J. A. Complexity measure. **IEEE Transactions on Software Engineering**, [S.l.], v. 2, n. 4, 1976.

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Referências Bibliográficas

MCPHEE, C. SENG 621: Software process management: software size estimation. **University of Calgary**, Calgary, 1999. Disponível em: <http://sern.ucalgary.ca/courses/seng/621/W98/kuloor/software_size_estimation.html>. Acesso em: 19 set. 2005.

MCT. MINISTÉRIO DE CIÊNCIA E TECNOLOGIA. Estudo mostra chances do Brasil no exterior. **Ministério de Ciência e Tecnologia**, Brasília, 30 mai. 2006. 2006. Disponível em: <http://agenciact.mct.gov.br/index.php?action=/content/view-&cod_objeto=34617>. Acesso em: 07 jun. 2006.

_____. Pesquisa: qualidade e produtividade no setor de software brasileiro. **Encontro da Qualidade e Produtividade em Software (EQPS)**, São Paulo, 2 ago. 2005. 2005. Disponível em: <http://www.mct.gov.br/upd_blob/2674.pdf>. Acesso em: 23/11/2005.

_____. Pesquisa: qualidade e produtividade no setor de software brasileiro. **Encontro da Qualidade e Produtividade em Software (EQPS)**, 1997. Disponível em: <<http://www.mct.gov.br/index.php/content/view/19839.html>>. Acesso em: 01/06/2007.

PMBOK. PROJECT MANAGEMENT BODY OF KNOWLEDGE. **A guide to the project management body of knowledge (PMBOK Guide)**. 3th, Newtown Square: Project Management Institute, Nov. 2004. 257 p.

PRESSMAN, R. S. **Engenharia de software**. 5. ed., São Paulo: McGraw-Hill, 2002. 843 p.

PUTNAM, L. H. A general empirical solution to the macro software sizing and estimating problem. **IEEE Transactions on Software Engineering**, [S.l.], v. 4, n. 4, p. 345-361, Jul. 1978.

ROSS, M. Size does matter: continuous size estimating and tracking. **Quantitative Software Management**, Glendale, Feb. 1999.

RUP. RATIONAL UNIFIED PROCESS. **Rational unified process**. [S.l.]: Rational Software Corporation, v. 2002.05.00, 2002.

SLACK, N.; CHAMBERS, S.; HARRISON, A. **Administração da produção**. 2. ed., São Paulo: Atlas, 2002. 747 p.

SOMMERVILLE, I. **Engenharia de software**. 6. ed., São Paulo: Prentice-Hall, 2003. 606 p.

SOWEBOK. **Guide to the software engineering body of knowledge**. Los Alamitos: IEEE Computer Society, v. 2004, 2004. 202 p.

TETILA, E. C.; COSTA, I.; SPÍNOLA, M. M. Estimativa de software combinando métricas: um estudo de caso. In: SIMPÓSIO DE ADMINISTRAÇÃO DA PRODUÇÃO, LOGÍSTICA E OPERAÇÕES INTERNACIONAIS (SIMPOI), 9, 2006. São Paulo. **Anais**. São Paulo: Fundação Getúlio Vargas. 29-31 ago. 2006.

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Referências Bibliográficas

THIOLLENT M. **Metodologia da Pesquisa-ação**. 13. ed., São Paulo: Cortez, 2004. 108 p.

TUNBRIDGE, P. Lord Kelvin: his influence on electrical measurements and units. **Institution of Engineering and Technology**, London, Apr. 1992.

UML. UNIFIED MODELING LANGUAGE. **Unified modeling language: superstructure**. [S.l.]: Object Management Group, v. 2.0, Aug. 2005. 710 p.

VAZQUEZ, C. E.; SIMÕES, G. S.; ALBERT, R. M. **Análise por ponto de função: medição, estimativas e gerenciamento de projetos de software**. 3. ed., São Paulo: Érica, 2005. 230 p.

Glossário

Vocábulo	Significado	Referência
<i>Alfa</i>	Tipo de teste em que o testador individual determina o que fazer. Geralmente é realizado pela organização do usuário final.	RUP. RATIONAL UNIFIED PROCESS. Rational unified process. [S.I.]: Rational Software Corporation, v. 2002.05.00, 2002.
Artefato	Uma informação que é produzida, modificada ou usada por um processo; define uma área de responsabilidade e está sujeita a controle de versão.	RUP. RATIONAL UNIFIED PROCESS. Rational unified process. [S.I.]: Rational Software Corporation, v. 2002.05.00, 2002.
Atividade	Uma unidade de trabalho que um papel pode ser solicitado a executar.	RUP. RATIONAL UNIFIED PROCESS. Rational unified process. [S.I.]: Rational Software Corporation, v. 2002.05.00, 2002.
<i>Baseline</i>	Um <i>release</i> revisado e aprovado de artefatos que constitui uma base ajustada para desenvolvimento ou evolução posterior e que só pode ser alterado através de um procedimento formal, como Gerenciamento de Mudança e Controle de Configuração.	RUP. RATIONAL UNIFIED PROCESS. Rational unified process. [S.I.]: Rational Software Corporation, v. 2002.05.00, 2002.
<i>Benchmarking</i>	Processo sistemático usado para estabelecer metas para melhorias no processo, nas funções, nos produtos, comparando uma empresa com outras.	Logisticabs: Enciclopédia <i>on-line</i> : http://www.logisticabs.com.br
<i>Beta</i>	Tipo de teste realizado “antes do <i>release</i> ” no qual uma amostra de consumidores-alvo testa o produto.	RUP. RATIONAL UNIFIED PROCESS. Rational unified process. [S.I.]: Rational Software Corporation, v. 2002.05.00, 2002.
<i>Bug</i>	Qualquer defeito encontrado em um programa de computador.	Wikipédia: Enciclopédia <i>on-line</i> : http://pt.wikipedia.org/
Casos do Uso	Uma descrição de comportamento do sistema em termos de seqüências de ações.	RUP. RATIONAL UNIFIED PROCESS. Rational unified process. [S.I.]: Rational Software Corporation, v. 2002.05.00, 2002.
Cenário	Uma instância de caso de uso descrita; subconjunto de um caso de uso. Usado para ilustrar uma interação ou a execução da instância de um caso de uso.	RUP. RATIONAL UNIFIED PROCESS. Rational unified process. [S.I.]: Rational Software Corporation, v. 2002.05.00, 2002.
Classe	Uma descrição de um conjunto de objetos que compartilham os mesmos atributos, operações, métodos, relacionamentos e semântica.	RUP. RATIONAL UNIFIED PROCESS. Rational unified process. [S.I.]: Rational Software Corporation, v. 2002.05.00, 2002.
Componente	Parte física e substituível de um sistema que empacota implementação, além de fornecer e se adaptar à realização de um conjunto de interfaces.	RUP. RATIONAL UNIFIED PROCESS. Rational unified process. [S.I.]: Rational Software Corporation, v. 2002.05.00, 2002.
Confiabilidade	Requisitos que envolvem tolerância a falhas, previsibilidade, recuperação e precisão de informações.	PRESSMAN, R. S. Engenharia de software . 5. ed., São Paulo: McGraw-Hill, 2002. 843 p.

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Glossário

Desempenho	Requisitos que especificam velocidade de processamento e recuperação de informações, tempo de resposta, taxa de utilização de recursos computacionais, etc.	PRESSMAN, R. S. Engenharia de software . 5. ed., São Paulo: McGraw-Hill, 2002. 843 p.
Disciplina	Coleção de atividades relacionadas a uma "área de interesse" principal.	RUP. RATIONAL UNIFIED PROCESS. Rational unified process . [S.l.]: Rational Software Corporation, v. 2002.05.00, 2002.
Documento	Coleção de informações que são representadas em papel ou em uma mídia que representa um papel.	RUP. RATIONAL UNIFIED PROCESS. Rational unified process . [S.l.]: Rational Software Corporation, v. 2002.05.00, 2002.
EAP	Ferramenta de decomposição hierárquica de tarefas que precisam ser feitas para completar um projeto em partes manejáveis.	Wikipédia : Enciclopédia <i>on-line</i> : http://pt.wikipedia.org/
Eficiência	Grau de aproveitamento dos recursos utilizados ao se produzir um produto ou realizar um serviço.	Wikipédia : Enciclopédia <i>on-line</i> : http://pt.wikipedia.org/
Empacotamento	A ocultação de uma representação interna de objeto de software. O objeto fornece uma interface que consulta e manipula os dados sem expor sua estrutura básica.	RUP. RATIONAL UNIFIED PROCESS. Rational unified process . [S.l.]: Rational Software Corporation, v. 2002.05.00, 2002.
Escalabilidade	Escalabilidade é a propriedade de um sistema qualquer que lhe confere a capacidade de aumentar seu desempenho sob carga quando recursos são acrescentados a esse sistema.	Wikipédia : Enciclopédia <i>on-line</i> : http://pt.wikipedia.org/
Estimativa	Método de prever os valores de consumo aproximado de uma operação.	PRESSMAN, R. S. Engenharia de software . 5. ed., São Paulo: McGraw-Hill, 2002. 843 p
Extensibilidade	Refere-se à capacidade de ampliar a funcionalidade presente sem conseqüências imprevistas sobre o conjunto da estrutura.	FAYAD, M.; CLINE, P. Aspects of Software Adaptability. Communications of the ACM . v.39, n.10, Oct. 1996.
Fase	O tempo entre dois marcos primários do projeto, durante o qual um conjunto bem definido de objetivos é atendido, artefatos são concluídos e decisões são tomadas sobre passar ou não para a próxima fase.	RUP. RATIONAL UNIFIED PROCESS. Rational unified process . [S.l.]: Rational Software Corporation, v. 2002.05.00, 2002.
<i>Feedback</i>	Retro-alimentação às ações que têm seu efeito informado claramente para permitir ajustes às futuras ações.	Wikipédia : Enciclopédia <i>on-line</i> : http://pt.wikipedia.org/
Fluxos de trabalho	A seqüência de atividades realizadas em um negócio que produz um resultado de valor observável para um ator individual do negócio.	RUP. RATIONAL UNIFIED PROCESS. Rational unified process . [S.l.]: Rational Software Corporation, v. 2002.05.00, 2002.
<i>Framework</i>	Uma micro-arquitetura que fornece um template extensível para aplicativos dentro de um determinado domínio.	RUP. RATIONAL UNIFIED PROCESS. Rational unified process . [S.l.]: Rational Software Corporation, v. 2002.05.00, 2002.

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Glossário

Funcionalidade	Requisitos que expressam o comportamento do sistema.	PRESSMAN, R. S. Engenharia de software . 5. ed., São Paulo: McGraw-Hill, 2002. 843 p
Iteração	Uma especificação de como os estímulos são enviados entre instâncias para realizar uma determinada tarefa. A interação é definida no contexto de uma colaboração.	RUP. RATIONAL UNIFIED PROCESS. Rational unified process . [S.l.]: Rational Software Corporation, v. 2002.05.00, 2002.
Instância	Entidade à qual um conjunto de operações pode ser aplicado e que possui um estado que armazena os efeitos das operações.	RUP. RATIONAL UNIFIED PROCESS. Rational unified process . [S.l.]: Rational Software Corporation, v. 2002.05.00, 2002.
Interface	Uma coleção de operações usadas para especificar um serviço de uma classe ou de um componente.	RUP. RATIONAL UNIFIED PROCESS. Rational unified process . [S.l.]: Rational Software Corporation, v. 2002.05.00, 2002.
Iteração	Uma seqüência distinta de atividades com um plano criado através de <i>baseline</i> e critérios de avaliação que resultam em um <i>release</i> (interno ou externo).	RUP. RATIONAL UNIFIED PROCESS. Rational unified process . [S.l.]: Rational Software Corporation, v. 2002.05.00, 2002.
<i>Kaizen</i>	Processo de melhorias contínuas, com bom senso e baixos investimentos.	Logisticabs : Enciclopédia <i>on-line</i> : http://www.logisticabs.com.br
Linha-base	Um <i>release</i> revisado e aprovado de artefatos que constitui uma base ajustada para desenvolvimento ou evolução posterior e que só pode ser alterado através de um procedimento formal, como Gerenciamento de Mudança e Controle de Configuração.	RUP. RATIONAL UNIFIED PROCESS. Rational unified process . [S.l.]: Rational Software Corporation, v. 2002.05.00, 2002.
Manutenibilidade	Facilidade com que um programa pode ser corrigido.	PRESSMAN, R. S. Engenharia de software . 5. ed., São Paulo: McGraw-Hill, 2002. 843 p
Marco	O ponto onde termina formalmente uma iteração; corresponde a um ponto de <i>release</i> .	RUP. RATIONAL UNIFIED PROCESS. Rational unified process . [S.l.]: Rational Software Corporation, v. 2002.05.00, 2002.
Medição	O ato de determinação de uma medida	IEEE. INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. IEEE standard glossary of software engineering terminology . New York: IEEE Standards Board, Std 610.12-1990, Set. 1990. 84 p.
Medida	Uma indicação quantitativa da extensão, quantidade, dimensão, capacidade ou tamanho de algum atributo de um produto, ou de um processo.	IEEE. INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. IEEE standard glossary of software engineering terminology . New York: IEEE Standards Board, Std 610.12-1990, Set. 1990. 84 p.

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Glossário

Métrica	Uma medida quantitativa do grau em que um sistema, componente ou processo possui determinado atributo.	IEEE. INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. IEEE standard glossary of software engineering terminology . New York: IEEE Standards Board, Std 610.12-1990, Set. 1990. 84 p.
Papel	Uma definição do comportamento e das responsabilidades de um indivíduo (ou conjunto de indivíduos) dentro do contexto de uma organização de engenharia de software.	RUP. RATIONAL UNIFIED PROCESS. Rational unified process . [S.l.]: Rational Software Corporation, v. 2002.05.00, 2002.
Performance	Requisitos que especificam velocidade de processamento e recuperação de informações, tempo de resposta, taxa de utilização de recursos computacionais, etc.	PRESSMAN, R. S. Engenharia de software . 5. ed., São Paulo: McGraw-Hill, 2002. 843 p
Plano de Desenvolvimento de Software	Conjunto de atividades e de tarefas divididas por seqüências de tempo, com recursos atribuídos, contendo dependências de tarefas, para o projeto.	RUP. RATIONAL UNIFIED PROCESS. Rational unified process . [S.l.]: Rational Software Corporation, v. 2002.05.00, 2002.
Portabilidade	Capacidade que um programa (software) tem de ser compilado ou executado em diferentes arquiteturas de sistemas computacionais (diferentes arquiteturas de hardware ou de sistema operacional).	Wikipédia : Enciclopédia <i>on-line</i> : http://pt.wikipedia.org/
Rastrear	Uma dependência que indica um relacionamento histórico ou processual entre dois elementos que representam o mesmo conceito sem regras específicas para derivar um do outro.	RUP. RATIONAL UNIFIED PROCESS. Rational unified process . [S.l.]: Rational Software Corporation, v. 2002.05.00, 2002.
Refinamento	Representa uma especificação mais completa de algo que já tenha sido especificado em um certo nível de detalhes.	RUP. RATIONAL UNIFIED PROCESS. Rational unified process . [S.l.]: Rational Software Corporation, v. 2002.05.00, 2002.
Reuso	Uso posterior ou repetido de um artefato. O uso de um artefato preexistente.	RUP. RATIONAL UNIFIED PROCESS. Rational unified process . [S.l.]: Rational Software Corporation, v. 2002.05.00, 2002.
<i>Release</i>	Um subconjunto do produto final que é o objeto de avaliação em um marco principal. É uma versão estável e executável do produto, que vem acompanhada dos artefatos necessários para sua utilização (como notas de release ou instruções de instalação, por exemplo).	RUP. RATIONAL UNIFIED PROCESS. Rational unified process . [S.l.]: Rational Software Corporation, v. 2002.05.00, 2002.
<i>Stakeholders</i>	É uma referência aos envolvidos no projeto de alguma forma. Por exemplo, cliente, equipe de projeto. Pode-se utilizar, também, o termo "parte interessada".	Wikipédia : Enciclopédia <i>on-line</i> : http://pt.wikipedia.org/

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Glossário

Subsistema	Um elemento de modelo que tem a semântica de um pacote, de tal forma que ele possa conter outros elementos de modelo, e uma classe, para que ela tenha comportamento.	RUP. RATIONAL UNIFIED PROCESS. Rational unified process . [S.l.]: Rational Software Corporation, v. 2002.05.00, 2002.
Suporte	Requisitos de teste e manutenção que determinam como o sistema irá se manter atualizado durante seu ciclo de vida. Geralmente, estes não são requisitos impostos ao sistema, mas ao processo de desenvolvimento.	PRESSMAN, R. S. Engenharia de software . 5. ed., São Paulo: McGraw-Hill, 2002. 843 p.
<i>Timeboxing</i>	A abordagem ao gerenciamento da programação de uma iteração recomendada no RUP: tendo estabelecido inicialmente o escopo e a programação de uma iteração, o gerente de projeto é incentivado a gerenciar de forma ativa esse escopo, a fim de cumprir a data final planejada para a iteração, em vez de adia-la para acomodar o escopo planejado originalmente, caso o desenvolvimento leve mais tempo que o imaginado.	RUP. RATIONAL UNIFIED PROCESS. Rational unified process . [S.l.]: Rational Software Corporation, v. 2002.05.00, 2002.
Transação	É um conjunto de atividades atômicas, as quais são executadas completamente ou não. Quando tiver a mesma transação em todos os diagramas de seqüência, como procedimentos de segurança, a transação deve ser contada apenas uma vez porque a funcionalidade é implementada apenas uma vez e reutilizada em outros casos de uso.	SCHNEIDER, G.; WINTERS, J. Applying use cases: a practical guide . 2nd, New York: Addison-Wesley, 2001.
Usabilidade	Requisitos relacionados a fatores humanos, como estética, facilidade de uso e de aprendizagem.	PRESSMAN, R. S. Engenharia de software . 5. ed., São Paulo: McGraw-Hill, 2002. 843 p.
Visão	Abstração de aspectos particulares do sistema, em nível de necessidades dos envolvidos e recursos do sistema	RUP. RATIONAL UNIFIED PROCESS. Rational unified process . [S.l.]: Rational Software Corporation, v. 2002.05.00, 2002.

Anexo I

O questionário a seguir identifica o nível de influência dos fatores de complexidade técnica, de acordo com IFPUG (2003) e KARNER (1993) e ambiental propostos por KARNER (1993) para desenvolvimento de projetos de software OO.

Questionário para identificar os níveis de influências dos fatores de complexidade técnica e ambiental da métrica UCP	
Instruções:	
Assinale o grau de influência de cada fator abaixo do projeto ou sistema de acordo com a escala de 0 a 5. Após determinar o valor de cada fator, multiplique pelo seu peso e some o total dos valores para obter os fatores de complexidade técnica e ambiental, conforme seção 3.3.3.	
Fatores de Complexidade Técnica	
1. Sistemas distribuídos: Descreve o nível em que a aplicação transfere dados entre os componentes do sistema.	
0 ()	A aplicação não participa da transferência de dados ou processamento de funções entre os componentes do sistema.
1 ()	A aplicação prepara dados para processamento pelo usuário final em outro componente do sistema, como planilhas eletrônicas ou banco de dados.
2 ()	Dados são preparados para transferência, então são processados em outro componente do sistema (não para processamento pelo usuário final).
3 ()	Processamento distribuído e transferência de dados <i>on-line</i> apenas e em uma direção.
4 ()	Processamento distribuído e transferência de dados <i>on-line</i> e em ambas direções.
5 ()	O processamento de funções é executado dinamicamente no componente mais apropriado do sistema.
2. Desempenho da aplicação: Identifica os objetivos de performance da aplicação, em termos de tempo de resposta ou taxa de transações, estabelecidos ou aprovados pelo usuário.	
0 ()	O usuário não estabeleceu nenhum requisito especial sobre performance.
1 ()	Requisitos de performance e projeto foram estabelecidos e revisados, mas nenhuma ação especial foi tomada.
2 ()	Tempo de resposta ou taxa de transações são críticos durante as horas de pico. Não é necessário nenhum projeto especial para a utilização de CPU. O limite para o processamento é o dia seguinte.
3 ()	Tempo de resposta ou taxa de transações são críticos durante todas as horas de trabalho. Não foi necessário nenhum projeto especial para a utilização de CPU. O limite de processamento é crítico.
4 ()	Adicionalmente, requisitos especificados pelo usuário são exigentes o bastante para que tarefas de análise de performance sejam necessárias na fase de projeto.
5 ()	Adicionalmente, ferramentas de análise de performance devem ser utilizadas nas fases de projeto, desenvolvimento e/ou implementação para que os requisitos de performance do usuário sejam atendidos.
3. Eficiência do usuário final (<i>on-line</i>): Descreve em que nível considerações sobre fatores humanos e facilidade de uso pelo usuário final influenciam o desenvolvimento da aplicação. O projeto inclui:	
➤	Auxílio para navegação, como teclas de função, saltos, menus gerados dinamicamente;
➤	Menus;
➤	Ajuda <i>on-line</i> e documentação;

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Anexos

<ul style="list-style-type: none"> ➤ Movimentação automática do cursor; ➤ Paginação; ➤ Impressão remota por meio de transações <i>on-line</i>; ➤ Teclas de função pré-definidas; ➤ Tarefas em lote submetidas a transações <i>on-line</i>; ➤ Seleção feita por posicionamento de cursor em tela de dados; ➤ Uso intenso de vídeo reverso, brilho, cores e outros indicadores; ➤ Documentação impressa das transações; ➤ Interface de mouse; ➤ Janelas <i>pop-up</i>; ➤ Utilização de número mínimo de telas para executar uma função do negócio; ➤ Suporte a dois idiomas (conte como quatro itens); ➤ Suporte a mais de dois idiomas (conte como seis itens); <p>0 () Nenhum dos itens anteriores.</p> <p>1 () De um a três dos itens anteriores.</p> <p>2 () De quatro a cinco dos itens anteriores.</p> <p>3 () Seis ou mais dos itens anteriores, mas não existem requisitos específicos do usuário associados à eficiência.</p> <p>4 () Seis ou mais dos itens anteriores e requisitos explícitos sobre a eficiência para o usuário final são fortes o bastante para necessitarem de tarefas de projeto que incluam fatores humanos, como minimizar o número de toques no teclado, maximizar padrões de campo e uso de modelos.</p> <p>5 () Seis ou mais dos itens anteriores e requisitos explícitos sobre a eficiência para o usuário final são fortes o bastante para necessitarem do uso de ferramentas e processos especiais para demonstrar que os objetivos foram alcançados.</p>
<p>4. Processamento interno complexo: Descreve em que nível o processamento lógico ou matemático influencia o desenvolvimento da aplicação. Os seguintes componentes estão presentes:</p>
<ul style="list-style-type: none"> ➤ Controle sensível e/ou processamento específico de segurança da aplicação. Exemplo: processamento especial de auditoria. ➤ Processamento lógico extensivo. Exemplo: sistema de gestão de crédito. ➤ Processamento matemático extensivo. Exemplo: sistema de otimização de corte de tecidos. ➤ Muito processamento de exceção resultando em transações incompletas que devem ser processadas novamente. Exemplo: transações incompletas em ATM em função de problemas de teleprocessamento, falta de dados ou de edição. ➤ Processamento complexo para manipular múltiplas possibilidades de entrada e saída. Exemplo: sistema de extrato de conta corrente que emite via terminal de retaguarda, auto atendimento, web, e-mail, telefone celular. <p>0 () Nenhum dos itens anteriores.</p> <p>1 () Qualquer um dos itens anteriores.</p> <p>2 () Quaisquer dois dos itens anteriores.</p> <p>3 () Quaisquer três dos itens anteriores.</p> <p>4 () Quaisquer quatro dos itens anteriores.</p> <p>5 () Todos os cinco itens anteriores.</p>
<p>5. Reusabilidade do código em outras aplicações: A aplicação e seu código foram especificamente projetados, desenvolvidos e suportados para serem reutilizados em outras aplicações.</p>
<p>0 () Não há código reutilizável.</p> <p>1 () Código reutilizável é utilizado na aplicação.</p> <p>2 () Menos de 10% da aplicação levou em consideração as necessidades de mais de um usuário.</p>

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Anexos

<p>3 () 10% ou mais da aplicação levou em consideração as necessidades de mais de um usuário.</p> <p>4. () A aplicação foi especificamente empacotada e/ou documentada para fácil reutilização. Ela é customizada pelo usuário por meio de manutenção de parâmetros.</p> <p>5. () A aplicação foi projetada e documentada para facilitar a reutilização de código e a aplicação é customizada para uso através de parâmetros que podem ser atualizados pelo usuário.</p>
<p>6. Facilidade de Instalação: Descreve em que nível a conversão de ambientes preexistentes influencia do desenvolvimento da aplicação.</p>
<p>0 () O usuário não definiu considerações especiais, assim como não é requerido nenhum <i>setup</i> para a instalação.</p> <p>1 () O usuário não definiu considerações especiais, mas é necessário <i>setup</i> para a instalação.</p> <p>2 () Requisitos de instalação e conversão foram definidos pelo usuário, e guias de conversão e instalação foram fornecidas e testadas. Não é considerado importante o impacto da conversão.</p> <p>3 () Requisitos de instalação e conversão foram definidos pelo usuário, e guias de conversão e instalação foram fornecidas e testadas. É considerado importante o impacto da conversão.</p> <p>4 () Além do item 2, ferramentas de instalação e conversão automáticas foram fornecidas e testadas.</p> <p>5 () Além do item 3, ferramentas de instalação e conversão automáticas foram fornecidas e testadas.</p>
<p>7. Usabilidade (facilidade operacional): Descreve em que nível a aplicação atende a alguns aspectos operacionais, como inicialização, segurança e recuperação. A aplicação minimiza a necessidade de atividades manuais, como montagem de fitas, manipulação de papel e intervenção manual pelo operador.</p>
<p>0 () Não foi estabelecida pelo usuário outra consideração que não os procedimentos normais de segurança.</p> <p>1 - 4 () Um, alguns ou todos os seguintes itens são válidos para a aplicação. Selecione todos aqueles que sejam válidos. Cada item tem um valor de um ponto, a exceção de onde seja citado o contrário.</p> <ul style="list-style-type: none"> ➤ Procedimentos de inicialização, salvamento e recuperação foram fornecidos, mas é necessária a intervenção do operador. ➤ Procedimentos de inicialização, salvamento e recuperação foram fornecidos, e não é necessária a intervenção do operador (conte como dois itens). ➤ A aplicação minimiza a necessidade de montagem de fitas. ➤ A aplicação minimiza a necessidade de manipulação de papel. <p>5 () Aplicação projetada para operação não-assistida. Isto é, não é necessário <i>nenhuma intervenção do operador</i> para operar o sistema, que não seja a inicialização e término da aplicação. A recuperação automática de erros é uma característica da aplicação.</p>
<p>8. Portabilidade: A aplicação foi especificamente projetada, desenvolvida e suportada para instalação em múltiplos locais para múltiplas organizações.</p>
<p>0 () Os requisitos do usuário não consideram a necessidade de mais de um usuário/local de instalação.</p> <p>1 () Necessidade de múltiplos locais foi considerada no projeto, e a aplicação foi projetada para operar apenas <i>nos mesmos</i> ambientes de hardware e de software <i>similares</i>.</p> <p>2 () Necessidade de múltiplos locais foi considerada no projeto, e a aplicação foi projetada para operar em apenas ambientes de hardware e de software <i>similares</i>.</p> <p>3 () Necessidade de múltiplos locais foi considerada no projeto, e a aplicação foi projetada para operar em ambientes <i>diferentes</i> de hardware e de software.</p> <p>4 () Adicionalmente aos itens 1 ou 2, plano de suporte e documentação são fornecidos e</p>

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Anexos

<p>testados para suportar a aplicação em múltiplos locais.</p> <p>5 () Adicionalmente ao item 3, plano de suporte e documentação são fornecidos e testados para suportar a aplicação em múltiplos locais.</p>
<p>9. Facilidade de Manutenção: Descreve em que nível a aplicação foi especificamente desenvolvida para facilitar a mudança de sua lógica de processamento ou estrutura de dados. As seguintes características podem ser válidas para a aplicação:</p> <ul style="list-style-type: none"> ➤ São fornecidos mecanismos de consulta flexível, que permitem a manipulação de pedidos simples; por exemplo, lógica de e/ou aplicada a apenas um arquivo lógico (conte como um item). ➤ São fornecidos mecanismos de consulta flexível, que permitem a manipulação de pedidos de média complexidade; por exemplo, lógica de e/ou aplicada a mais de um arquivo lógico (conte como dois itens). ➤ São fornecidos mecanismos de consulta flexível, que permitem a manipulação de pedidos complexos; por exemplo, lógica de e/ou combinadas em um ou mais arquivos lógicos (conte como três itens). ➤ Dados de controle do negócio são mantidos pelo usuário por meio de processos interativos, mas as alterações só têm efeito no próximo dia útil. ➤ Dados de controle do negócio são mantidos pelo usuário por meio de processos interativos, e as alterações têm efeito imediato (conte como dois itens). <p>0 () Nenhum dos itens anteriores. 1 () Qualquer um dos itens anteriores. 2 () Quaisquer dois dos itens anteriores. 3 () Quaisquer três dos itens anteriores. 4 () Quaisquer quatro dos itens anteriores. 5 () Todos os cinco itens anteriores.</p>
<p>10. Concorrência: Descreve em que nível o alto volume de transações influencia o projeto, desenvolvimento, instalação e suporte da aplicação.</p> <p>0 () Não é antecipado nenhum período de pico de transações. 1 () São antecipados períodos de pico de processamento (por exemplo, mensal, quinzenal, periódico, anual). 2 () Picos de transação semanais são previstos. 3 () Picos de transação diários são previstos. 4 () Altas taxas de transação definidas pelo usuário nos requisitos ou os níveis de serviços acordados são altos o bastante para requererem tarefas de análise de performance na fase de projeto. 5 () Adicionalmente, existem requisitos de ferramentas de análise de performance nas fases de projeto, desenvolvimento e/ou instalação.</p>
<p>11. Características especiais de segurança: Descreve o nível de segurança exigido pela aplicação.</p> <p>0 () Nenhuma solicitação do usuário para considerar a necessidade de controle de segurança da aplicação. 1 () Necessidade de controle de segurança foi levada em consideração no projeto do sistema. 2 () Necessidade de controle de segurança foi levada em consideração no projeto do sistema e a aplicação foi projetada para ser acessada somente por usuários autorizados. 3 () Necessidade de controle de segurança foi levada em consideração no projeto do sistema e a aplicação foi projetada para ser acessada somente por usuários autorizados. O acesso será controlado e auditado. 4 () Um plano de segurança foi elaborado e testado para suportar o controle de acesso a aplicação. 5 () Um plano de segurança foi elaborado e testado para suportar o controle de acesso a</p>

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Anexos

aplicação e a auditoria.
<p>12. Acessível por terceiros: Descreve o nível de acesso de terceiros (usuários, organizações externas, proprietário) na aplicação. O acesso à aplicação pode incluir as seguintes características:</p> <ul style="list-style-type: none"> ➤ A aplicação está disponível ao público da Internet em geral. ➤ A aplicação mantém e controla a integração de terceiros. ➤ A aplicação consome serviços de terceiros (contar um item para cada terceira parte envolvida). ➤ A aplicação publica e documenta uma API (<i>Application Programming Interface</i>). ➤ A aplicação deve acessar o software de terceiros sem tornar a aplicação pública (contar como dois itens). <p>0 () Nenhum dos itens anteriores. 1 () Qualquer um dos itens anteriores. 2 () Quaisquer dois dos itens anteriores. 3 () Quaisquer três dos itens anteriores. 4 () Quaisquer quatro dos itens anteriores. 5 () Todos os cinco itens anteriores.</p>
<p>13. Facilidades especiais de treinamento: Descreve o nível de facilidade de treinamento de usuários.</p> <p>0 () Nenhuma solicitação do usuário para considerar a necessidade de treinamento especial. 1 () Necessidade de treinamento especial foi levada em consideração no projeto do sistema. 2 () Necessidade de treinamento foi levada em consideração no projeto do sistema e a aplicação foi projetada para ser acessada com facilidade pelos usuários. 3 () Necessidade de treinamento especial foi levado em consideração no projeto do sistema e a aplicação foi projetada para ser utilizada por usuários de diversos níveis. 4 - 5 () Um plano de treinamento foi elaborado e testado para facilitar o uso da aplicação.</p>
Fatores de Complexidade Ambiental
<p>1. Familiaridade com o processo formal de desenvolvimento: Descreve a experiência da equipe com o processo (método) utilizado no desenvolvimento do projeto.</p> <p>0 () A equipe não é familiar com o processo de desenvolvimento de software. 1 () A equipe tem conhecimento teórico do processo de desenvolvimento de software. 2 - 3 () Um ou mais membros utilizou o processo uma ou poucas vezes. 3 - 4 () Pelo menos a metade dos membros da equipe tem experiência no uso do processo em diferentes projetos. 5. () Toda a equipe tem experiência no uso do processo em vários projetos diferentes.</p>
<p>2. Trabalhadores com dedicação parcial: Mede a estabilidade da equipe e a influência do trabalho parcial na produtividade.</p> <p>0 () Não tem membro com dedicação parcial. 1 - 2 () Poucos membros (20%) trabalham em período parcial. 3 - 4 () A metade dos membros da equipe trabalham em período parcial. 5 () Todos os membros da equipe trabalham em período parcial.</p>
<p>3. Capacidade do líder do projeto: Descreve a experiência do líder do projeto na análise de requisitos e modelagem.</p> <p>0 () O líder do projeto é novato. 1 - 2 () Possui experiência de poucos projetos. 3 - 4 () Pelo menos 2 anos de experiência com vários projetos.</p>

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Anexos

5 ()	Pelo menos 3 anos de experiência com vários projetos.
4. Experiência com a aplicação de desenvolvimento: Descreve a experiência com diferentes tipos de aplicação ou com o tipo de aplicação que está sendo desenvolvida.	
0 ()	Todos os membros da equipe são novatos.
1 - 2 ()	Poucos membros da equipe possuem alguma experiência (de 1 a 1 ½ ano). Os outros são novatos.
3 ()	Todos os membros tem mais de 1 ½ ano de experiência.
4 ()	A maioria da equipe tem 2 anos de experiência.
5 ()	Todos os membros são experientes.
5. Experiência em orientação a objetos: Descreve a experiência da equipe com análise e projeto OO, modelagem de casos de uso, classes e componentes.	
0 ()	A equipe não é familiar com análise e projeto OO.
1 ()	Todos os membros tem menos de 1 ano de experiência.
2 - 3 ()	Todos os membros tem de 1 a 1 ½ ano de experiência.
4 ()	A maioria da equipe tem mais de 2 anos de experiência.
5 ()	Todos os membros são experientes (mais de 2 anos).
6. Motivação: Descreve a motivação da equipe.	
0 ()	Não motivada.
1 - 2 ()	Pouca motivada.
3 - 4 ()	Motivada.
5 ()	Bastante motivada.
7. Dificuldade na linguagem de programação: Descreve a experiência com ferramentas primárias de desenvolvimento e com a linguagem de programação escolhida.	
0 ()	Todos os membros da equipe são programadores experientes.
1 ()	A maioria dos membros da equipe possuem mais de 2 anos de experiência.
2 ()	Todos os membros tem mais de 1 ½ ano de experiência.
3 ()	A maioria da equipe tem mais de 1 ano de experiência.
4 ()	Poucos membros da equipe tem alguma experiência (1 ano). Os outros são novatos.
5 ()	Todos os membros da equipe são novatos.
8. Requisitos Estáveis: Mede o grau de mudança de requisitos e inseguranças sobre o significado dos requisitos.	
0 ()	Requisitos muitos instáveis com mudanças freqüentes.
1 - 2 ()	Requisitos instáveis. Clientes demandam algumas mudanças em diversos intervalos.
3 - 4 ()	Estabilidade global. Pequenas mudanças são realizadas.
5 ()	Requisitos estáveis ao longo do desenvolvimento.

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Anexos

Anexo II

Lista das instituições de software que receberam o trabalho via e-mail. (Adaptado de Fonte: MCT, 1997).

Lista das instituições de software que receberam o trabalho via e-mail		
Nome	Site	E-mail
ABACUS	http://www.abacusinformatica.com.br	corporativo@abacusinformatica.com.br
ABC 71 SOLUÇÕES EM INFORMÁTICA	http://www.abc71.com.br	consultoria@abc71.com.br
AÇÃO INFORMÁTICA	http://www.acaoinformatica.com.br	comercial@acaoinformatica.com.br
AÇÃO SISTEMAS	http://www.universalrh.com.br	pnurh@acao.com.br
ACC INFORMÁTICA	http://www.acc.com.br	draco@acc.com.br
ADAPTA INFORMÁTICA	http://www.adapta.com.br	comercial@adapta.com.br
ADP	http://www.adp.com.br	adp@adp.com.br
ADSERVIS	http://www.adservis.com.br	adservis@adservis.com.br
AES	http://www.aes.com.br	aes@aes.com.br
AGÊNCIA MODERNA TECNOLOGIA	http://www.amt.com.br	comercial@amt.com.br
AGRISOFT	http://www.agrisoft.com.br	agrisoft@agrisoft.com.br
AIT	http://www.ait.com.br	ti@ait.com.br
ALTO QI	http://www.altoqi.com.br	dti@altoqi.com.br
ALTUS	http://www.altus.com.br	comercial@altus.com.br
AMBIENTE	http://www.polymap.com	admin@polymap.com
AMPLA CONSULTORIA EM INFORMAÇÃO	http://www.amplaconsultoria.com.br	rh@amplaconsultoria.com.br
APLIGRAF	http://www.apligraf.com.br	apligraf@apligraf.com.br
APORTE	http://www.aporte.com.br	aporte@aporte.com.br
APS FROMSOFT	http://www.fromsoft.com.br	negócios@fromsoft.com.br
ARTWARE SOLUÇÕES EM CONECTIVIDADE	http://www.artware.com.br	contato@artware.com.br
ASPEC	http://www.aspec.com.br	atendimento@aspec.com.br
ASTROLÁBIO	http://www.astrolabio.com.br	info@astrolabio.com.br
ATAN SISTEMAS	http://www.atan.com.br	atan@atan.com.br
ATI	http://www.ati.com.br	postmaster@ati.com.br
ATONUS	http://www.atonus.com.br	antonio@atonus.com.br
ATOS	http://www.atos.com.br	atos@atos.com.br
ATTEST	http://www.attest.com.br	attest@attest.com.br
AUDACES	http://www.audaces.com.br	audaces@audaces.com.br
AUDIDATA	http://www.audidata.com.br	comercial@audidata.com.br
AUGE TECNOLOGIA E SISTEMAS	http://www.auge.com.br	auge@auge.com.br
AUSLAND	http://www.ausland.com.br	comercial@ausland.com.br
AUTOENGE	http://www.autoenge.com.br	vendas@autoenge.com.br
AYK	http://www.ayk.com.br	comercial@ayk.com.br
BASE	http://www.base.inf.br	marcel@base.inf.br
BEMATECH	http://www.bematech.com.br	info@bematech.com.br
BERNARD SISTEMAS	http://www.bernard.com.br	comercial@bernard.com.br
BETA INFORMÁTICA	http://www.betainformatica.com.br	betainfo@betainformatica.com.br

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Anexos

BETHA SISTEMAS	http://www.betha.com.br	betha@betha.com.br
BKM	http://www.bkmonline.com.br	bkm@bkmonline.com.br
BL INFORMÁTICA	http://www.blnet.com	bl@blnet.com
BMA	http://www.bmanet.com.br	producoes@ilhavertical.com.br
BMS	http://www.bms.com.br	alipio.lessa@bms.com.br
BUREAU INFORMÁTICA	http://www.internetbureau.com.br	internetbureau@terra.com.br
CEBI INFORMÁTICA	http://www.cebi.com.br	cebi@cebi.com.br
CELEPAR	http://www.celepar.pr.gov.br	jones@celepar.gov.br
CELTA SISTEMAS E MÉTODOS	http://www.celta.org.br	celta@celta.org.br
CEPROMAT	http://www.cepromat.com.br	auditoria.geral@cepromat.com.br
CETIL SISTEMAS	http://www.cetil.com.br	cetil@cetil.com.br
CGI CONSULTORIA	http://www.webcgi.com.br	gerencia@webcgi.com.br
CHART INFORMÁTICA	http://www.chart.com.br	chart@chart.com.br
CHRONUS	http://www.chronus.com.br	chronus@chronus.com.br
CI&T INFORMÁTICA	http://www.cit.com.br	rh@cit.com.br
CIAQUATRO	http://www.ciaquatro.com.br	cqinfo@ciaquatro.com.br
CIASC	http://www.ciasc.gov.br	ciasc@ciasc.gov.br
CISS INFORMÁTICA	http://www.ciss.com.br	comercial@ciss.com.br
CNP	http://www.cnp.com.br	cnp@cnp.com.br
CODIUB – TI	http://www.codiub.com.br	webmaster@codiub.com.br
CONSISA INFORMÁTICA	http://www.consisa.inf.br	comercial@consisa.inf.br
CONSULT	http://www.consult.com.br	consultoria@consult.com.br
CONSULTE	http://www.tecno-consulte.com.br	comercial@e2000i.com.br
CONSULTWARE INFORMÁTICA	http://www.consultware.com.br	consultware@consultware.com.br
CONSYSTEM	http://www.consystem.com.br	comercial@consystem.com.br
CONTMASTER	http://www.contmaster.com.br	vendas@contmaster.com.br
CPM	http://www.cpm.com.br	faleconosco@cpm.com.br
CQS	http://www.cqs.com.br	tecnico@cqs.com.br
CRITERIUS	http://www.criterius.com.br	criterius@criterius.com.br
CSP	http://www.csp.com.br	vendas@csp.com.br
CTZ CONSULTORIA	http://www.ctz.com.br	ctz@ctz.com.br
CUMERLATO & SCHUSTER	http://www.cs.inf.br	luiz.piton@cs.inf.br
D'ARTE SISTEMAS	http://www.darte.com.br	darte@darte.com.br
DATA CEMPRO	http://www.datacempro.com.br	webmaster@datacempro.com.br
DATA FIX	http://www.datafix.com.br	datafix@datafix.com.br
DATAMEC	http://www.unisys.com.br	unisys@unisys.com.br
DATAREGIS	http://www.dataregis.com.br	vendas@dataregis.com.br
DATASERV	http://www.sigma.com.br	rh@sigma.com.br
DATASUL	http://www.datasul.com.br	datasul@datasul.com.br
DECON INFORMÁTICA	http://www.deconinformatica.com.br	decon@deconinformatica.com.br
DEGGY SYSTEM	http://www.deggy.com.br	deggy@deggy.com.br
DIGITRO	http://www.portaldigitro.com.br	portal@digitro.com.br
DISOFT	http://www.disoft.com.br	rh@disoft.com.br
DOCTOR SYS	http://www.doctorsys.com.br	contato@doctorsys.com.br
DRIVE CONSULTORIA	http://www.drive.com.br	drive@drive.com.br
DSC	http://www.dsc.ufcg.edu.br	dsc@dsc.ufcg.edu.br
EAC	http://www.eacsoftware.com.br	comercial@eacsoftware.com.br
EASY SYSTEMS INFORMÁTICA	http://www.easysystem.com.br	contato@easysystem.com.br
EBS SISTEMAS	http://www.ebs.com.br	ebs@ebs.com.br

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Anexos

ECS INFORMÁTICA	http://www.ecs.com.br	comercial@ecs.com.br
EFE INFORMÁTICA	http://www.efec.com.br	efe@efe.com.br
ELIPSE	http://www.elipse.com.br	elipse@elipse.com.br
ELOTECH INF. E SISTEMAS	http://www.elotech.com.br	elotech@elotech.com.br
ENGESOFT	http://www.engesoft.com	sac@engesoft.com
ENGSOFT	http://www.engsoft.com.br	engsoft@engsoft.com.br
EPSOFT	http://www.epsoft.com.br	contato@epsoft.com.br
EQUATORIAL	http://www.equatorialsistemas.com.br	ccghizoni@equatorialsistemas.com.br
EQUIPLANO	http://www.equiplano.com.br	equiplano@equiplano.com.br
ESQUEMATIKA	http://www.esquematika.com.br	comercial@esquematika.com.br
ESTALO DESENVOLVIMENTO	http://www.estalo.com.br	estalo@estalo.com.br
EXIT INFORMÁTICA	http://www.exitinformatica.com.br	faleconosco@exitinformatica.com.br
EXOTECH INFORMÁTICA	http://www.exotech.com.br	llevinia@exotech.com.br
FÁCIL INFORMÁTICA	http://www.facil.com.br	comercial@facil.com.br
FENASOFT	http://www.fenasoft.com.br	fenasoft@fenasoft.com.br
FINANCIAL INFORMÁTICA	http://www.financialinformatica.com.br	marketing@financialinformatica.com.br
FLUXUS	http://www.fluxus.com.br	desenvolvimento@fluxus.com.br
FOCCO	http://www.focco.com.br	focco@focco.com.br
FOTON	http://www.fotongroup.com.br	info@fotongroup.com.br
FRAGATA SOFTWARE	http://www.fragata.com.br	vendas@fragata.com.br
GENESE	http://www.novagenese.com.br	webmaster@novagenese.com.br
GENESIS SOFTWARE	http://www.genesis-softwareonline.com	info@genesis-softwareonline.com
GENS	http://www.gens.com.br	comercial@gens.com.br
GEOGRAPH	http://www.geograph.com.br	geograph@geograph.com.br
GEOTEC	http://www.geotec.com.br	geotec@geotec.com.br
GFS	http://www.gfs.com.br	gfs@gfs.com.br
GKO	http://www.gko.com.br	info@gko.com.br
GREEN SOFTWARE	http://www.green.com.br	rh@green.com.br
GUBERMAN INFORMÁTICA	http://www.guberman.com.br	client@guberman.com.br
HDS	http://www.dalmark.com.br	comercial@dalmark.com.br
HOTSOFT	http://www.hotsoft.com.br	vendas@hotsoft.com.br
HST SISTEMAS & TECNOLOGIA	http://www.hst.com.br	rh@hst.com.br
HUMANA ASSESSORIA	http://www.gestaohumana.com.br	rh@gestaohumana.com.br
IBM BRASIL	http://www.ibm.com.br	nucleoibm@dancarmarketing.com.br
INDUSOFT	http://www.indusoft.com.br	contato@indusoft.com.br
INTELBRÁS	http://www.intelbras.com.br	rh@intelbras.com.br
ITAIPIU	http://www.pti.org.br	pti@pti.org.br
ITEC	http://www.itec.com.br	telemkt@itec.com.br
JABUR INFORMÁTICA	http://www.jaburinfo.com.br	sales@jaburinfo.com.br
JME INFORMÁTICA	http://www.jme.com.br	jme@jme.com.br
KAIZEN DATABASE	http://www.kankei.com.br	kankei@kankei.com.br
KERNEL	http://www.kernelinformatica.com.br	kernel@kernelinformatica.com.br
KINETICS APLICAÇÕES MULTIMÍDIA	http://www.kinetics.com.br	kinetics@kinetics.com.br
KONSULTEX	http://www.konsultex.com.br	info@konsultex.com.br
KOUNEN	http://www.kounen.com	webmaster@kounen.com
LAN DESIGNERS	http://www.landesigners.com.br	rh@landesigners.com.br

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Anexos

LASERTECH	http://www.lasertech.com.br	pedidos@asertech.com.br
LEME INFORMÁTICA	http://www.leme.psi.br	webmaster@leme.psi.br
LEUCOTRON	http://www.leucotron.com.br	atende@leucotron.com.br
LEXIKON INFORMÁTICA	http://www.lexikon.com.br	suporte@lexikon.com.br
LG INFORMÁTICA	http://www.lg.com.br	lg@lg.com.br
LIDERCOMP	http://www.lidercomp.com.br	lidercomp@lidercomp.com.br
LIGHT INFOCON	http://www.lightinfocon.com.br	info@lightinfocon.com.br
LIMIAR INFORMÁTICA	http://www.limiar.com.br	limiar@limiar.com.br
LIVEWARE	http://www.liveware.com.br	liveware@liveware.com.br
LOGOCENTER	http://www.logocenter.com.br	info@logocenter.com.br
LYNX	http://www.lynxtec.com.br	vendas@lynxtec.com.br
MALISOFT	http://www.malisoft.com.br	malisoft@malisoft.com.br
MANAGER SYSTEMS	http://www.sisclinica.com.br	info@managersystems.com.br
MARCA SISTEMAS	http://www.marca.com.br	marca@marca.com.br
MICROSOFT	http://www.microsoft.com.br	rh@microsoft.com.br
MICROMEGA	http://www.micromega.com.br	micromega@micromega.com.br
MICROSIGA	http://www.microsiga.com.br	microsiga@microsiga.com.br
N&L INFORMÁTICA	http://www.nl.com.br	rh@nl.com.br
NASAJON SISTEMAS	http://www.nasajon.com.br	nasajon@nasajon.com.br
NEC DO BRASIL	http://www.nec.com.br	nec@nec.com.br
NEO MIDIA	http://www.neomidia.org	neomidia@neomidia.org
NETWORK INFORMÁTICA	http://www.ebusines.com.br	ebusines@ebusines.com.br
NEWDATA INFORMÁTICA	http://www.newinfo.com.br	newdata@newinfo.com.br
OBJETO	http://www.objetodesign.com	odesign@objetodesign.com
OCTUS	http://www.octus.com.br	info@octus.com.br
OGEDA	https://www.ogeda.com.br	rh@ogeda.com.br
OLOSTECH	http://www.olostech.com.br	olos@olostech.com.br
ONION SISTEMAS	http://www.onion.com.br	info@onion.com.br
OPENCADD	http://www.opencadd.com.br	marketing@opencadd.com.br
OPTIMAL	http://www.optimal.com.br	info@optimal.com.br ...
ORIGIN	http://www.atosorigin.com.br	rh.atosorigin@atosorigin.com.br
PLASSOFT	http://www.plassoft.com.br	info@plassoft.com.br
POLITEC	http://www.politec.com.br	webmaster@bsb.politec.com.br
PROCESS INFORMÁTICA	http://www.netprocess.com.br	comercial@netprocess.com.br
PROCOMP	http://www.procomp.com.br	marketing@procomp.com.br
PROSOFT	http://www.prosoft.com.br	rh@prosoft.com.br
QUALIPRO	http://www.qualipro.com.br	cinqdir@cinq.com.br
QUALYSUL	http://www.qualysul.com.br	qualysul@qualysul.com.br
QUASAR	http://www.quasar.org.br	contato@quasar.org.br
RCN CONSULTORIA	http://www.topmanager.com.br	desenvolvimento@rcnnet.com.br
RCS INFORMÁTICA	http://www.rcs.srv.br	rca@rca.srv.br
RED CONSULTORIA	http://www.redconsultoria.com.br	contato@redconsultoria.com.br
RETA INFORMÁTICA	http://www.reta.com.br	comercial@reta.com.br
RGM DO BRASIL	http://www.rgmdobrasil.com.br	rgmbr@rgmdobrasil.com.br
RIOSOFT	http://www.riosoft.com.br	mkt@riosoft.com.br
RM SISTEMAS	http://www.rm.com.br	comercial@rm.com.br
SENIOR SISTEMAS	http://www.senior.com.br	senior@senior.com.br
SERPRO	http://www.serpro.gov.br	serpro@serpro.gov.br

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Anexos

SIGMA DATASERV	http://www.sigma.com.br	rh@sigma.com.br
SISTEMAS CORDON	http://www.scordon.com.br	comercial@scordon.com.br
SOFT CONSULTORIA	SOFT CONSULTORIA	soft@soft.com.br
SUN SOFTWARE	http://www.sunsoftware.com.br	marketing@sunsoftware.com.br
TECNOCOOP SISTEMAS	http://www.tecnocoop.com	tecnofjf@tecnocoop.com.br
TECNOLOGICA INFORMÁTICA	http://www.tecnologica.com.br	comercial@tecnologica.com.br
TRANSOFT	http://www.transoft.com.br	sandoval@transoft.com.br
TSA	http://www.tsamg.com.br	tsa@tsamg.com.br
ULTRASYSTEM	http://www.ultracar.com.br	suporte@ultracar.com.br
UNIWARE	http://www.uniware.com.br	uniware@uniware.com.br
UNIWAY	http://www.uniway.com.br	central@uniway.com.br
UTESC	http://www.utesc.br	utesc@utesc.br
UTILSOFT	http://www.utilsoft.com.br	rhumanos@utilsoft.com.br
UX INFORMÁTICA	http://www.ntux.com.br	contato@ntux.com.br
VECTOR	http://www.vectornet.com.br	comercial@vectornet.com.br
VGA-INFORMÁTICA	http://www.vgainfo.com.br	vgainfo@vgainfo.com.br
VIASOFT	http://www.viasoft.com.br	viasoft@viasoft.com.br
VISUAL SYSTEMS	http://www.visualonline.com.br	atendimento@visualonline.com.br
VM CONSULTORIA	http://www.vm.com.br	vmisp@vm.com.br
WA INFORMÁTICA	http://www.wainf.com.br	suporte@wainf.com.br
WIT PLANEJAMENTO	http://www.wit.com.br	wit@wit.com.br
WK SISTEMAS	http://www.wk.com.br	desenvolvimento@wk.com.br
YOKOGAWA	http://www.yokogawa.com.br	marketing@br.yokogawa.com
ZÊNITE	http://www.zenitetecnologia.com.br	contato@zenitetecnologia.com.br

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Apêndices

Apêndice A

O questionário a seguir utiliza a métrica *Function Points* para estimar o custo do *Projeto 1*, da Instituição privada. Maiores detalhes sobre o projeto podem ser observados a seguir.

Estimativa de Custo de Projeto de Software Análise por Pontos de Função (FP – <i>Function Points</i>)		
Nome do Projeto do Sistema: Projeto 1 (nome não divulgado)		
Nome do Cliente: Instituição privada (nome não divulgado)		
Data de Confeção da Planilha: 28/12/2004		
Autor da Planilha: não divulgado		
Revisor da Planilha: não divulgado		
Versão da Planilha: 1.0		
1. Funções de Entrada ou Entrada Externa (EE)		
Fluxo de Entrada ou Impulsos	Quantidade	Total
	45	216
2. Funções de Saída ou Saída Externa (SE)		
Fluxo de Saída ou Resposta	Quantidade	Total
	20	90
3. Funções de Arquivo ou Arquivos Lógicos Internos (ALI)		
Entidade ou Arquivo Lógico	Quantidade	Total
	37	264,55
4. Funções de Interface ou Arquivo de interface Externa (AIE)		
Arquivo de Interface Externa	Quantidade	Total
	7	35,70
5. Funções de Consulta Externa (CE)		
Fluxo de Saída ou Resposta	Quantidade	Total
	30	123
Total de PF não ajustados:		729,3
6. Fatores de Influência ou Níveis de Influência		
Características Gerais do Sistema		
Comunicação de dados		5
Processamento distribuído		4
Desempenho		5
Utilização de equipamentos		3
Volume de transações		4
Entrada de dados on-line		4

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Apêndices

Eficiência do usuário final	3
Atualização on-line	4
Processamento complexo	4
Reusabilidade	2
Facilidade operacional	2
Facilidade de instalação	4
Múltiplos locais	1
Facilidade de mudanças	4
Total de PF ajustados:	831,3
7. Índice de Produtividade da Equipe	6,29
8. Esforço (em homens-hora)	5.227,2
9. Horas trabalhadas no mês	132
10. Esforço (em homens-mês)	39,6
11. Prazo Ótimo (em meses)	6
12. Média dos valores cobrados por hora	R\$ 64,04
13. Custo de desenvolvimento do software (em Reais)	R\$ 334.752,00

Apêndice B

O questionário a seguir utiliza a métrica *Use Case Points* para estimar o custo do *Projeto 1*, da Instituição privada. Maiores detalhes sobre o projeto podem ser observados a seguir.

Estimativa de Custo de Projeto de Software	
Análise por Pontos de Casos de Uso (UCP - <i>Use Case Points</i>)	
Nome do Projeto do Sistema: Projeto 1 (nome não divulgado)	
Nome do Cliente: Instituição privada (nome não divulgado)	
Data de Confeção da Planilha: 15/4/2005	
Autor da Planilha: não divulgado	
Revisor da Planilha: não divulgado	
Versão da Planilha: 1.0	
1. Peso dos atores	
Quantidade	Total
58	107
2. Peso dos casos de Uso	
Quantidade	Peso dos Casos de Uso
21	178
Total de PCU não ajustados:	
285	
3. Fatores de Influência ou Níveis de Influência	
Fatores de Complexidade Técnica	
Sistemas Distribuídos	8
Desempenho da aplicação	5
Eficiência do usuário final (on-line)	3
Processamento interno complexo	4
Reusabilidade do código em outras aplicações	2
Facilidade de instalação	2
Usabilidade (facilidade operacional)	1
Portabilidade	2
Facilidade de manutenção	4
Concorrência	4
Características especiais de segurança	2
Acesso direto para terceiros	4
Facilidades especiais de treinamento	2
Fatores de Complexidade Ambiental	
Familiaridade com o processo formal de desenvolvimento	6
Trabalhadores com dedicação parcial	-4
Capacidade do líder de projeto	2

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Apêndices

Experiência com a aplicação de desenvolvimento	1.5
Experiência em orientação a objetos	4
Motivação	3
Dificuldade da linguagem de programação	-3
Requisitos estáveis	6
Total de PCU ajustados:	277,13
4. Índice de Produtividade da Equipe	20
5. Esforço (em homens-hora)	5.542,6
6. Horas trabalhadas no mês	132
7. Esforço (em homens-mês)	41,99
8. Prazo (em meses)	6,2
9. Média dos valores cobrados por hora	R\$ 64,04
10. Custo de desenvolvimento do software (em Reais)	R\$ 354.950,00

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Apêndices

Apêndice C

O questionário a seguir utiliza o Processo de Estimativa de Software para estimar o custo do Projeto 1, da Instituição privada. Maiores detalhes sobre o projeto podem ser observados a seguir.

Estimativa de Custo de Projeto de Software	
Análise por Pontos de Casos de Uso (UCP - <i>Use Case Points</i>) utilizando o Processo de Estimativa de Software	
Nome do Projeto do Sistema: Projeto 1	
Nome do Cliente: Instituição privada (nome não divulgado)	
Data de Confeção da Planilha: 24/05/2007	
Autor da Planilha: Everton Castelão Tetila	
Revisor da Planilha: Ivanir Costa	
Versão da Planilha: 1.0	
1. Peso dos atores	
Quantidade	Total
58	107
2. Peso dos casos de Uso	
Quantidade	Peso dos Casos de Uso
21	178
Total de PCU não ajustados:	
285	
3. Fatores de Influência ou Níveis de Influência	
Fatores de Complexidade Técnica	
Sistemas Distribuídos	8
Desempenho da aplicação	5
Eficiência do usuário final (on-line)	3
Processamento interno complexo	4
Reusabilidade do código em outras aplicações	2
Facilidade de instalação	2
Usabilidade (facilidade operacional)	1
Portabilidade	2
Facilidade de manutenção	4
Concorrência	4
Características especiais de segurança	2
Acesso direto para terceiros	4
Facilidades especiais de treinamento	2
Fatores de Complexidade Ambiental	
Familiaridade com o processo formal de desenvolvimento	6
Trabalhadores com dedicação parcial	-4

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Apêndices

Capacidade do líder de projeto	2		
Experiência com a aplicação de desenvolvimento	1.5		
Experiência em orientação a objetos	4		
Motivação	3		
Dificuldade da linguagem de programação	-3		
Requisitos estáveis	6		
Total de PCU ajustados:	277,13		
Índice de Produtividade da Equipe	20		
Esforço (em homens-hora)	5.542,6		
Horas trabalhadas no mês	132		
Esforço (em homens-mês)	41,99		
Prazo (em meses)	6,2		
Média dos valores cobrados por hora	R\$ 64,04		
Custo de desenvolvimento do software (em Reais)	R\$ 354.950,00		
Estimativa de Recursos			
Tipo	Recurso	Quantidade	Custo
TOTAL			
Previsão de Variações de Estimativas			
Tipo	Variação do período (%)	Variação do período (reais)	
1. Inflação média anual	5,7	R\$ 10.452,00	
TOTAL		R\$ 10.452,00	
Estimativa de Custo Global do Projeto			
1. Custo de desenvolvimento do software			R\$ 354.950,00
2. Estimativa de recursos			R\$ 0,00
3. Previsão de variações de estimativas			R\$ 10.452,00
TOTAL			R\$ 365.402,00

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Apêndices

Apêndice D

O questionário a seguir utiliza a métrica *Function Points* para estimar o custo do *Projeto 2*, da Instituição Pública. Maiores detalhes sobre o projeto podem ser observados a seguir.

Estimativa de Custo de Projeto de Software Análise por Pontos de Função (FP – <i>Function Points</i>)		
Nome do Projeto do Sistema: Projeto 2 (nome não divulgado)		
Nome do Cliente: Instituição pública (nome não divulgado)		
Data de Confeção da Planilha: 05/07/2006		
Autor da Planilha: não divulgado		
Revisor da Planilha: não divulgado		
Versão da Planilha: 1.0		
14. Funções de Entrada ou Entrada Externa (EE)		
Fluxo de Entrada ou Impulsos	Quantidade	Total
	108	518
15. Funções de Saída ou Saída Externa (SE)		
Fluxo de Saída ou Resposta	Quantidade	Total
	75	337
16. Funções de Arquivo ou Arquivos Lógicos Internos (ALI)		
Entidade ou Arquivo Lógico	Quantidade	Total
	27	193
17. Funções de Interface ou Arquivo de interface Externa (AIE)		
Arquivo de Interface Externa	Quantidade	Total
	10	50
18. Funções de Consulta Externa (CE)		
Fluxo de Saída ou Resposta	Quantidade	Total
	78	323
Total de PF não ajustados:		1421
19. Fatores de Influência ou Níveis de Influência		
Características Gerais do Sistema		
Comunicação de dados		4
Processamento distribuído		4
Desempenho		4
Utilização de equipamentos		3
Volume de transações		4
Entrada de dados on-line		5

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Apêndices

Eficiência do usuário final	3
Atualização on-line	4
Processamento complexo	4
Reusabilidade	2
Facilidade operacional	2
Facilidade de instalação	3
Múltiplos locais	1
Facilidade de mudanças	3
Total de PF ajustados:	1577,31
20. Índice de Produtividade da Equipe	7,64
21. Esforço (em homens-hora)	12.058
22. Horas trabalhadas no mês	120
23. Esforço (em homens-mês)	100,48
24. Prazo Ótimo (em meses)	8,65
25. Média dos valores cobrados por hora	R\$ 49,39
26. Custo de desenvolvimento do software (em Reais)	R\$ 595.652,00

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Apêndices

Apêndice E

O questionário a seguir utiliza a métrica *Use Case Points* para estimar o custo do *Projeto 2*, da Instituição Pública. Maiores detalhes sobre o projeto podem ser observados a seguir.

Estimativa de Custo de Projeto de Software	
Análise por Pontos de Casos de Uso (UCP - <i>Use Case Points</i>)	
Nome do Projeto do Sistema: Projeto 2 (nome não divulgado)	
Nome do Cliente: Instituição pública (nome não divulgado)	
Data de Confeção da Planilha: 09/07/2006	
Autor da Planilha: não divulgado	
Revisor da Planilha: não divulgado	
Versão da Planilha: 1.0	
11. Peso dos atores	
Quantidade	Total
31	73
12. Peso dos casos de Uso	
Quantidade	Peso dos Casos de Uso
74	425
Total de PCU não ajustados:	
498	
13. Fatores de Influência ou Níveis de Influência	
Fatores de Complexidade Técnica	
Sistemas Distribuídos	8
Desempenho da aplicação	4
Eficiência do usuário final (on-line)	3
Processamento interno complexo	4
Reusabilidade do código em outras aplicações	2
Facilidade de instalação	1,5
Usabilidade (facilidade operacional)	1
Portabilidade	2
Facilidade de manutenção	3
Concorrência	4
Características especiais de segurança	4
Acesso direto para terceiros	5
Facilidades especiais de treinamento	3
Fatores de Complexidade Ambiental	
Familiaridade com o processo formal de desenvolvimento	4,5
Trabalhadores com dedicação parcial	-4
Capacidade do líder de projeto	1,5

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Apêndices

Experiência com a aplicação de desenvolvimento	1.5
Experiência em orientação a objetos	2
Motivação	3
Dificuldade da linguagem de programação	-3
Requisitos estáveis	4
Total de PCU ajustados:	577,68
14. Índice de Produtividade da Equipe	20
15. Esforço (em homens-hora)	11.553,60
16. Horas trabalhadas no mês	120
17. Esforço (em homens-mês)	96,28
18. Prazo (em meses)	8,5
19. Média dos valores cobrados por hora	R\$ 49,39
20. Custo de desenvolvimento do software (em Reais)	R\$ 570.632,00

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Apêndices

Apêndice F

O questionário a seguir utiliza o Processo de Estimativa de Software para estimar o custo do Projeto 2, da Instituição Pública. Maiores detalhes sobre o projeto podem ser observados a seguir.

Estimativa de Custo de Projeto de Software	
Análise por Pontos de Casos de Uso (UCP - <i>Use Case Points</i>) utilizando o Processo de Estimativa de Software	
Nome do Projeto do Sistema: Projeto 2 (nome não divulgado)	
Nome do Cliente: Instituição pública (nome não divulgado)	
Data de Confeção da Planilha: 25/05/2007	
Autor da Planilha: Everton Castelão Tetila	
Revisor da Planilha: Ivanir Costa	
Versão da Planilha: 1.0	
4. Peso dos atores	
Quantidade	Total
31	73
5. Peso dos casos de Uso	
Quantidade	Peso dos Casos de Uso
74	425
Total de PCU não ajustados:	
498	
6. Fatores de Influência ou Níveis de Influência	
Fatores de Complexidade Técnica	
Sistemas Distribuídos	8
Desempenho da aplicação	4
Eficiência do usuário final (on-line)	3
Processamento interno complexo	4
Reusabilidade do código em outras aplicações	2
Facilidade de instalação	1,5
Usabilidade (facilidade operacional)	1
Portabilidade	2
Facilidade de manutenção	3
Concorrência	4
Características especiais de segurança	4
Acesso direto para terceiros	5
Facilidades especiais de treinamento	3
Fatores de Complexidade Ambiental	
Familiaridade com o processo formal de desenvolvimento	4,5
Trabalhadores com dedicação parcial	-4

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Apêndices

Capacidade do líder de projeto	1,5		
Experiência com a aplicação de desenvolvimento	1.5		
Experiência em orientação a objetos	3		
Motivação	3		
Dificuldade da linguagem de programação	-3		
Requisitos estáveis	3		
Total de PCU ajustados:	577,68		
Índice de Produtividade da Equipe	20		
Esforço (em homens-hora)	11.553,60		
Horas trabalhadas no mês	120		
Esforço (em homens-mês)	96,28		
Prazo (em meses)	8,5		
Média dos valores cobrados por hora	R\$ 49,39		
Custo de desenvolvimento do software (em Reais)	R\$ 570.632,00		
Estimativa de Recursos			
Tipo	Recurso	Quantidade	Custo
1. Treinamento	J2EE	1	R\$ 3.300,00
TOTAL			R\$ 3.300,00
Previsão de Variações de Estimativas			
Tipo	Variação do período (%)	Variação do período (reais)	
1. Inflação média anual	~ 3,5	R\$ 14.146,00	
TOTAL		R\$ 14.146,00	
Estimativa de Custo Global do Projeto			
4. Custo de desenvolvimento do software			R\$ 570.632,00
5. Estimativa de recursos			R\$ 3.300,00
6. Previsão de variações de estimativas			R\$ 14.146,00
TOTAL			R\$ 588.078,00

Apêndice G

O questionário a seguir utiliza a métrica *Function Points* para estimar o custo *Projeto 3*, da Instituição Pública. Maiores detalhes sobre o projeto podem ser observados a seguir.

Estimativa de Custo de Projeto de Software Análise por Pontos de Função (FP – <i>Function Points</i>)		
Nome do Projeto do Sistema: Projeto 3 (nome não divulgado)		
Nome do Cliente: Instituição pública (nome não divulgado)		
Data de Confeção da Planilha: 11/02/2006		
Autor da Planilha: não divulgado		
Revisor da Planilha: não divulgado		
Versão da Planilha: 1.0		
27. Funções de Entrada ou Entrada Externa (EE)		
Fluxo de Entrada ou Impulsos	Quantidade	Total
	96	460
28. Funções de Saída ou Saída Externa (SE)		
Fluxo de Saída ou Resposta	Quantidade	Total
	42	188
29. Funções de Arquivo ou Arquivos Lógicos Internos (ALI)		
Entidade ou Arquivo Lógico	Quantidade	Total
	64	456
30. Funções de Interface ou Arquivo de interface Externa (AIE)		
Arquivo de Interface Externa	Quantidade	Total
	8	41
31. Funções de Consulta Externa (CE)		
Fluxo de Saída ou Resposta	Quantidade	Total
	67	273
Total de PF não ajustados:		1418
32. Fatores de Influência ou Níveis de Influência		
Características Gerais do Sistema		
Comunicação de dados		4
Processamento distribuído		3
Desempenho		3
Utilização de equipamentos		3
Volume de transações		4
Entrada de dados on-line		3

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Apêndices

Eficiência do usuário final	3
Atualização on-line	4
Processamento complexo	3
Reusabilidade	2
Facilidade operacional	2
Facilidade de instalação	3
Múltiplos locais	1
Facilidade de mudanças	2
Total de PF ajustados:	1488,90
33. Índice de Produtividade da Equipe	6,4
34. Esforço (em homens-hora)	9.539
35. Horas trabalhadas no mês	120
36. Esforço (em homens-mês)	79,49
37. Prazo Ótimo (em meses)	7,91
38. Média dos valores cobrados por hora	R\$ 53,47
39. Custo de desenvolvimento do software (em Reais)	R\$ 510.050,00

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Apêndices

Apêndice H

O questionário a seguir utiliza o Processo de Estimativa de Software para estimar o custo do Projeto 3, da Instituição Pública. Maiores detalhes sobre o projeto podem ser observados a seguir.

Estimativa de Custo de Projeto de Software	
Análise por Pontos de Casos de Uso (UCP - <i>Use Case Points</i>)	
Nome do Projeto do Sistema: Projeto 3 (nome não divulgado)	
Nome do Cliente: Instituição pública (nome não divulgado)	
Data de Confeção da Planilha: 15/02/2006	
Autor da Planilha: não divulgado	
Revisor da Planilha: não divulgado	
Versão da Planilha: 1.0	
21. Peso dos atores	
Quantidade	Total
39	92
22. Peso dos casos de Uso	
Quantidade	Peso dos Casos de Uso
75	430
Total de PCU não ajustados:	
522	
23. Fatores de Influência ou Níveis de Influência	
Fatores de Complexidade Técnica	
Sistemas Distribuídos	8
Desempenho da aplicação	5
Eficiência do usuário final (on-line)	5
Processamento interno complexo	4
Reusabilidade do código em outras aplicações	3
Facilidade de instalação	2
Usabilidade (facilidade operacional)	2
Portabilidade	8
Facilidade de manutenção	4
Concorrência	4
Características especiais de segurança	4
Acesso direto para terceiros	4
Facilidades especiais de treinamento	3
Fatores de Complexidade Ambiental	
Familiaridade com o processo formal de desenvolvimento	4,5
Trabalhadores com dedicação parcial	-2
Capacidade do líder de projeto	1,5

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Apêndices

Experiência com a aplicação de desenvolvimento	2
Experiência em orientação a objetos	4
Motivação	3
Dificuldade da linguagem de programação	-2
Requisitos estáveis	6
Total de PCU ajustados:	538
24. Índice de Produtividade da Equipe	20
25. Esforço (em homens-hora)	10.760
26. Horas trabalhadas no mês	120
27. Esforço (em homens-mês)	89,66
28. Prazo (em meses)	8,3
29. Média dos valores cobrados por hora	R\$ 53,47
30. Custo de desenvolvimento do software (em Reais)	R\$ 575.337,00

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Apêndices

Apêndice I

O questionário a seguir utiliza o Processo de Estimativa de Software para estimar o custo do Projeto 3, da Instituição Pública. Maiores detalhes sobre o projeto podem ser observados a seguir.

Estimativa de Custo de Projeto de Software	
Análise por Pontos de Casos de Uso (UCP - <i>Use Case Points</i>) utilizando o Processo de Estimativa de Software	
Nome do Projeto do Sistema: Projeto 3 (nome não divulgado)	
Nome do Cliente: Instituição pública (nome não divulgado)	
Data de Confeção da Planilha: 29/05/2007	
Autor da Planilha: Everton Castelão Tetila	
Revisor da Planilha: Ivanir Costa	
Versão da Planilha: 1.0	
7. Peso dos atores	
Quantidade	Total
39	92
8. Peso dos casos de Uso	
Quantidade	Peso dos Casos de Uso
75	430
Total de PCU não ajustados:	
522	
9. Fatores de Influência ou Níveis de Influência	
Fatores de Complexidade Técnica	
Sistemas Distribuídos	8
Desempenho da aplicação	5
Eficiência do usuário final (on-line)	5
Processamento interno complexo	4
Reusabilidade do código em outras aplicações	3
Facilidade de instalação	2
Usabilidade (facilidade operacional)	2
Portabilidade	8
Facilidade de manutenção	4
Concorrência	4
Características especiais de segurança	4
Acesso direto para terceiros	4
Facilidades especiais de treinamento	3
Fatores de Complexidade Ambiental	
Familiaridade com o processo formal de desenvolvimento	4,5
Trabalhadores com dedicação parcial	-2

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Apêndices

Capacidade do líder de projeto	1,5		
Experiência com a aplicação de desenvolvimento	2		
Experiência em orientação a objetos	4		
Motivação	3		
Dificuldade da linguagem de programação	-2		
Requisitos estáveis	6		
Total de PCU ajustados:	538		
Índice de Produtividade da Equipe	20		
Esforço (em homens-hora)	10.760		
Horas trabalhadas no mês	120		
Esforço (em homens-mês)	89,66		
Prazo (em meses)	8,3		
Média dos valores cobrados por hora	R\$ 53,47		
Custo de desenvolvimento do software (em Reais)	R\$ 575.337,00		
Estimativa de Recursos			
Tipo	Recurso	Quantidade	Custo
1. Ferramenta de testes para aplicativos Java, VS.NET e Web.	Rational Functional Tester	1	R\$ 5.813,00
TOTAL\			
Previsão de Variações de Estimativas			
Tipo	Variação do período (%)	Variação do período (reais)	
1. Inflação média anual	3,1	R\$ 12.336,00	
2. Reajuste salarial de membros da equipe	20,83	R\$ 15.780,00	
TOTAL			R\$ 28.116,00
Estimativa de Custo Global do Projeto			
7. Custo de desenvolvimento do software			R\$ 575.337,00
8. Estimativa de recursos			R\$ 5.813,00
9. Previsão de variações de estimativas			R\$ 28.116,00
TOTAL			R\$ 609.266,00

Apêndice J

Quadro comparativo das métricas de estimativas de projetos de software												
Projeto	Instituição	Métrica Utilizada	Estimativas Iniciais			Resultados Obtidos			Percentual de Erro de Custo (%)			
			Tamanho	Esforço	Prazo	Custo	Esforço	Prazo	Custo	Estimativas Iniciais / Resultados Obtidos		
1	Instituição privada	FP	831,3	5.227,2	6	334.752,00	5.644	6,25	371.580,00	-9,91		
			Estimativas Iniciais			Resultados Obtidos			Percentual de Erro de Custo (%)			
		UCP	277,13	5.542,6	6,2	354.950,00	5.644	6,25	371.580,00	-4,47		
		Métrica Utilizada	Estimativas Iniciais			Resultados Obtidos			Percentual de Erro de Custo (%)			
		Processo	277,13	5.542,6	6,2	365.402,00	5.644	6,25	371.580,00	-1,69		
			Estimativas Iniciais			Resultados Obtidos			Estimativas Iniciais / Resultados Obtidos			

Processo de Estimativa de Software com a Métrica *Use Case Points*, PMBOK e RUP.

Apêndices

Apêndice K

Quadro comparativo das métricas de estimativas de projetos de software												
Projeto	Instituição	Métrica Utilizada	Estimativas Iniciais			Resultados Obtidos			Percentual de Erro de Custo (%)			
			Tamanho	Esforço	Prazo	Custo	Esforço	Prazo	Custo	Estimativas Iniciais / Resultados Obtidos	Estimativas Iniciais / Resultados Obtidos	
Projeto 2	Instituição pública	FP	1577,31	12.058	8,65	595.544,00	11.840	8,6	603.730,00	-1,35		
			Estimativas Iniciais			Resultados Obtidos			Percentual de Erro de Custo (%)		Estimativas Iniciais / Resultados Obtidos	
			Tamanho	Esforço	Prazo	Custo	Esforço	Prazo	Custo	-5,48		
		UCP	577,68	11.553,60	8,5	570.632,00	11.840	8,6	603.730,00			
		Métrica Utilizada	Estimativas Iniciais			Resultados Obtidos			Percentual de Erro de Custo (%)		Estimativas Iniciais / Resultados Obtidos	
		Processo	Tamanho	Esforço	Prazo	Custo	Esforço	Prazo	Custo	-2,59		
			577,68	11.553,60	8,5	588.078,00	11.840	8,6	603.730,00			

