

UNIVERSIDADE PRESBITERIANA MACKENZIE

LEANDRO AP. PAPAÍ MARIN

CUSTOS EM PROJETOS DE TI: COMO ESTIMAR, DETERMINAR E CONTROLAR
CUSTOS PARA PROJETOS DE SOFTWARES.

São Paulo

2012

LEANDRO AP. PAPAÍ MARIN

CUSTOS EM PROJETOS DE TI: COMO ESTIMAR, DETERMINAR E CONTROLAR
CUSTOS PARA PROJETOS DE SOFTWARES.

Trabalho de Conclusão de Curso apresentado ao Programa de Pós-graduação Lato Sensu da Universidade Presbiteriana Mackenzie, como requisito parcial para a obtenção do Título de Especialista em Gestão de Projetos.

ORIENTADOR: PROF^a. DR^a. ÉLIDA JACOMINI NUNES

São Paulo

2012

RESUMO

Conforme publicação feita pela Harvard Business Review, (HAZRATI, 2011), os projetos de TI têm passado do orçamento estimado em cerca de 200% e tiveram atrasos de 70% dos projetos. Os resultados mostram que os projetos de TI são muitos mais arriscados do que se imagina e custam muito caro para as organizações e que podem até acabar com a saúde financeira das empresas. Este trabalho foi realizado com o intuito de mostrar as principais técnicas de estimativas e métrica de um projeto de TI. As diferentes formas de tratar cada peculiaridade de um projeto de TI podem fazer adotar diferentes formas de estimativas e métricas. Com uma estimativa mais assertiva o gerente de projeto terá um orçamento mais enxuto e a organizações terão uma visão mais claro de quanto o projeto terá de gasto e conseguindo implementar o produto ou serviço final. Também são abordadas as principais técnicas e ferramentas utilizadas para estimativa de projeto de TI como COCOMO II e Pontos por função e além de abordar o controle de custo durante do projeto usando o Valor Agregado.

Palavra-Chaves: Custo, Estimativas, Gestão de Projetos, Engenharia de Software.

ABSTRACT

As publication made by Harvard Business Review, (Hazrati, 2011), IT projects have passed the budget estimated at about 200% and 70% had delays of projects. The results show that many IT projects are riskier than you think and are very expensive for organizations and may even end the financial health of companies. This work was performed in order to show the main techniques and metrics estimates an IT project. The different ways of treating each peculiarity of an IT project can adopt different forms of estimates and metrics. With an estimated more assertive project manager will have a leaner budget and organizations will have a clearer vision of how the project will implement spending and getting the final product or service. Also addressed are the main techniques and tools used to estimate IT project as COCOMO II and points beyond by function and to address cost control during the project using Earned Value.

Key word: Cost, Estimates, Project Management, Software Engineering.

LISTA DE FIGURAS

| | | |
|------------|---|----|
| Figura 01 | EAP..... | 22 |
| Figura 02 | Computando a métrica pontos-por-função..... | 27 |
| Figura 03: | LOC por ponto-por-função..... | 28 |
| Figura 04: | Recursos de um projeto..... | 44 |

LISTA DE TABELAS

| | | |
|-----------|--|----|
| Tabela 01 | Precisão Bottom-UP..... | 23 |
| Tabela 02 | Tipo de estimativas..... | 23 |
| Tabela 03 | Métricas orientadas ao tamanho..... | 26 |
| Tabela 04 | Classificação de atores..... | 30 |
| Tabela 05 | Classificação dos Casos de Uso..... | 31 |
| Tabela 06 | Fatores de complexidade técnica..... | 32 |
| Tabela 07 | Fatores de complexidade ambiental..... | 33 |
| Tabela 08 | Tabela de complexidade..... | 39 |

LISTA DE ABREVIATURAS

| | |
|----------|---|
| AC | <i>Actual Cost</i> |
| ACWP | <i>Actual cost of work performed</i> |
| BCWS | <i>Budget cost of work scheduled</i> |
| BCWP | <i>Budget cost of work performed</i> |
| CII | COCOMO II |
| COCOMO | Constructive Cost Model |
| COTS | Commercial Of-The-Shelf |
| COTA | Custo Orçado do Trabalho Agendado |
| COTR | Custo Orçado do Trabalho Realizado |
| CPI | <i>Cost Performance Index</i> |
| CR | Custo Real |
| CRTR | Custo Real do Trabalho Realizado |
| CV | <i>Cost Variance</i> |
| EAP | Estrutura Analítica do projeto |
| EM | <i>Effort Multipliers</i> |
| ENT | Estimativa no término |
| EPT | Estimativa para terminar |
| EV | <i>Earned Value</i> |
| FCA | Fator de complexidade ambiental. |
| FCT | Fator de complexidade técnica. |
| FENABRAN | Federação Nacional dos Bancos |
| GVA | Gerenciamento de Valor Agregado |
| IDC | Índice de desempenho de custo |
| IDP | Índice de desempenho de prazo |
| IDPT | Índice de desempenho para término |
| IOC | <i>Initial Operational Capability</i> |
| KLOC | <i>Thousand Lines of Code</i> |
| LCO | <i>Lyfe Cycle Objectives</i> |
| LOC | <i>Lines of Code</i> |
| ONT | Orçamento no término |
| OO | Orientado a Objeto. |
| PMBOK | <i>Project Management Body of Knowledge</i> |

| | |
|--------|---|
| PCU | Pontos Caso de Uso |
| PCUA | Pontos de caso de Uso Ajustado. |
| PCUNA | Pontos Caso de Uso Não Ajustados |
| PF | Pontos por função. |
| PM | Persons-Month |
| PMI | Project Manager Institute |
| PV | Planned Value |
| QFD | <i>Quality Function Deployment</i> |
| SPI | <i>Schedule Performance Index</i> |
| SV | <i>Scheduled Variance</i> |
| SEE | Software Engineering Environment |
| SF | Scale Factors |
| TI | Tecnologia da informação |
| TV | <i>Time Variance</i> |
| TPNAA | Total de Pesos não Ajustados dos Atores |
| TPNAUC | Total de Pesos não ajustados dos casos de usos. |
| VA | Valor Agregado |
| VP | Valor Planejado |
| WBS | Work Breakdown Structure |

Sumário

| | |
|--|----|
| 1 INTRODUÇÃO..... | 11 |
| 1.1 OBJETIVOS..... | 11 |
| 1.1.1 OBJETIVO GERAL..... | 11 |
| 1.1.2 OBJETIVOS ESPECÍFICOS | 11 |
| 1.2 JUSTIFICATIVA | 12 |
| 1.3 METODOLOGIA..... | 13 |
| 2 ESTIMANDO CUSTO EM PROJETO DE SOFTWARE | 14 |
| 2.1 PLANEJANDO UM PROJETO..... | 15 |
| 2.2 ESTIMATIVAS DE SOFTWARE..... | 16 |
| 2.3 VANTAGENS DE ESTIMAR O CUSTO | 17 |
| 2.4 PROBLEMAS EM ESTIMAR PROJETOS DE SOFTWARE | 17 |
| 2.5 INFORMAÇÕES NECESSÁRIAS PARA ESTIMAR | 18 |
| 3 ESTIMATIVAS DE UM PROJETO..... | 20 |
| 3.1 ESTIMATIVAS ANÁLOGA..... | 20 |
| 3.2 ESTIMATIVA PARAMÉTRICA | 21 |
| 3.2 ESTIMATIVA “BOTTOM-UP” | 21 |
| 3.4 OUTRAS ESTIMATIVAS SEGUNDO O PMBOK | 23 |
| 3.4 TIPOS DE MÉTRICAS DE SOFTWARES | 24 |
| 3.4.1 MÉTRICAS ORIENTADAS AO TAMANHO | 25 |
| 3.4.2 MÉTRICAS ORIENTADAS A FUNÇÃO | 26 |
| 3.4.3 MÉTRICAS ORIENTADAS A OBJETO..... | 28 |
| 3.4.4 MÉTRICAS ORIENTADAS A CASO DE USO | 30 |
| 3.4.5 MÉTRICAS DE PROJETO DE ENGENHARIA DA WEB | 33 |
| 4 METODOLOGIAS DE ESTIMATIVAS PARA TI..... | 35 |
| 4.1 COCOMO II..... | 35 |

| | |
|--|----|
| 4.2 PONTOS POR FUNÇÃO | 37 |
| 5 DETERMINANDO ORÇAMENTO..... | 41 |
| 5.1 DETERMINANDO ORÇAMENTO EM PROJETOS DE TI..... | 42 |
| 5.2 RECURSOS EM PROJETOS DE TI..... | 44 |
| 5.2.1 RECURSOS HUMANOS | 44 |
| 5.2.2 RECURSOS SOFTWARE REUSÁVEIS | 45 |
| 5.2.2 RECURSOS DE AMBIENTE..... | 45 |
| 6 CONTROLANDO OS CUSTOS EM UM PROJETO DE TI..... | 47 |
| 6.1 FERRAMENTAS PARA CONTROLAR CUSTOS SEGUNDO PMBOK..... | 47 |
| 6.1.2 PREVISÃO..... | 47 |
| 6.1.3 INDICE DE DESEMPENHO PARA TÉRMINO | 48 |
| 6.1.4 ANÁLISE DE DESEMPENHO..... | 49 |
| 6.1.5 OUTRAS TÉCNICAS..... | 49 |
| 6.2 VALOR AGREGADO | 49 |
| 6.3 VALOR AGREGADO PARA CONTROLAR PROJETO..... | 50 |
| 7 CONCLUSÃO..... | 52 |
| REFERÊNCIAS | 54 |

1 INTRODUÇÃO

Atualmente os projetos de software possuem um grande índice de falha, e isso acontece por vários motivos. Uns desses motivos são as estimativas que são feitas errada, com um custo menor que o previsto.

Um dos grandes problemas em ambiente de TI (Tecnologia da Informação) é que são utilizadas várias tecnologias, cada um com sua peculiaridade, tornando assim um conhecimento maior para poder estimar o que será gasto no projeto, caso seja envolvido todas essas tecnologias.

Neste estudo mostraremos quais as formas de estimar custos para projetos de softwares para várias tecnologias, mostrar as técnicas utilizadas no mercado e quais são as melhores práticas segundo o instituto de gerenciamento projetos, PMI (*Project Manager Institute*), usando como referencia o PMBOK (*Project Management Body of Knowledge*).

Iremos estudar também a formar de controlar e monitorar estes gastos para que o projeto não comprometa o orçamento definido no inicio e quais ferramentas são utilizadas.

1.1 OBJETIVOS

1.1.1 OBJETIVO GERAL

Vamos mostrar as várias formas de estimativas de custo de um projeto de software, mostrando as principais técnicas e ferramentas que são utilizados. Queremos mostrar que os projetos de software podem possui uma estimativa mais assertiva se utilizar as técnicas e ferramentas corretas para cada situação e utilizar as melhores práticas que são indicadas no PMBOK.

1.1.2 OBJETIVOS ESPECÍFICOS

Como podemos estimar custos de um projeto de softwares para ambiente com múltiplas tecnologias e ambientes, que rodam complexos sistemas de software? Mostraremos neste estudo as principais técnicas e ferramentas utilizadas para estimar o custo de um projeto de software.

Vamos mostrar também como podemos controlar esses custos e verificar se o custo orçado no projeto não está passando do seu limite. Como lidar com itens que não estava no escopo e com retrabalho que podem acontecer e assim aumentando o custo do projeto.

E avaliaremos qual a contribuição que um trabalho de estimativa de custo em projeto de software pode contribuir para que as empresas não tenham prejuízo e que as mesmas não tenham surpresas com os gastos durante o projeto.

1.2 JUSTIFICATIVA

Uma pesquisa feita em 2008 pela MBI Mayer&Bunge (COMPUTEWORLD, 2008) com 185 empresas concluiu que as empresas brasileiras gastam 3% dos seu faturamentos em investimento em TI. Mas para instituições financeiras e seguradoras brasileiras os investimentos são mais intensos. Segundo a FENABRAN, os bancos têm gastado com TI em torno de 19,4 bilhões por ano para manter a imagem de pioneirismo e inovação.

Muitos destes gastos são para projetos em TI para alterar ou construir processos novos nos sistemas e que atendam as demandas do mercado e que também se inovem em um mercado tão competitivo como o setor financeiro.

Um relatório conhecido como “*Chaos Report*” (SOUZA, 2010) publicado em 2009 pela *The Standish Group* identificou os principais motivos de sucesso e fracasso de projetos em TI nos Estados Unidos. O país em questão gasta em cerca de 250 bilhões de dólares em projetos de TI, um total de 175.000 projetos.

Este relatório classificou e identificou que 24% dos projetos fracassam, 44% dos projetos são entregues com sucesso parcial e 32% dos projetos obtêm sucesso.

Outra pesquisa realizada pela Meta Group em 2000, (postado por CARVALHO, 2008) indica que mais da metade dos projetos de TI estão 180% acima do orçado resultado em perdas \$59 bilhões para as corporações.

Como os investimentos em TI estão crescendo, e cada vez mais são criados novos projetos, como podemos estimar os custos de um projeto e controlar estes custos para que a empresa invista na sua área de TI e não possua um prejuízo. Nesta monografia pretendemos mostrar quais são as formas e técnicas utilizadas para estimar custos e controlar custos.

Em um ambiente de TI de um banco, por exemplo, possui vários sistemas de softwares complexo e com inúmeras tecnologias diferentes para poder atender diferentes linhas de negócio, por exemplo, sistemas de móbile para o cliente acessar sua conta pelo telefone, ou sistemas com tecnologia de internet usada como ferramentas do banco para análise de crédito de um cliente, ou a própria tecnologia da alta plataforma (mainframe) para rodar processos pesados em poucos segundos. Como podemos estimar o custo de um projeto de software onde

todas essas tecnologias estão interligadas e cada uma possui um custo e uma forma de estimar um esforço.

Mostraremos também quais são os riscos de estimar um custo errado de um projeto de software e quais as principais consequências que isso pode tornar para o projeto e qual a melhor forma de contornar este erro.

Além de estimar o custo de um projeto é necessário controlar o custo que o projeto está tendo para que o projeto não fique fora de controle e estoure o seu orçamento. Quais são as principais técnicas e boas práticas para acompanhar o custo que um projeto está tendo e como lidar com um possível erro de escopo ou retrabalhos que os projetos sempre possuem. Nesta monografia pretendemos esclarecer essas dúvidas que acontecem em empresas que possuem um ambiente complexo com múltiplas tecnologias.

1.3 METODOLOGIA

O tipo de pesquisa realizado será o teórico tendo como base desta monografia estudar a área de conhecimento de custo, mostrando as principais técnicas e ferramentas para estimar o custo de um projeto e monitorar o quanto está sendo gasto durante um projeto, baseado na metodologia PMI e estudos feitos na engenharia de software, tendo como principal autor Roger S. Pressman. Serão abordadas também técnicas de decomposição de estimativas de software como COCOMO II e pontos por função, e técnicas de controle de projeto para monitorar o custo como Valor Agregado.

2 ESTIMANDO CUSTO EM PROJETO DE SOFTWARE

Para uma conduzir um projeto de software bem sucedido, é necessário fazer o planejamento do projeto para evitar erros no fim do projeto e tentar as corrigir possíveis erros no inicio do projeto e não no final, onde isso aumenta o custo do trabalho. Conforme, PRESSMAN (1995, pag 519), os projetos gastam em cerca de 80% de seu tempo em corrigindo problemas que não foram identificados antes. Por isso é necessário estimar o trabalho a ser feito, os recursos necessários, estabelecer um cronograma que define as tarefas e o responsável e especifique as dependências.

O gerente estima em um projeto para elaborar uma previsão de quais e quantos recursos serão necessários para finalizar um projeto, pois a duração de uma tarefa depende de números de desenvolvedores, da complexidade das atividades, da produtividade e das incertezas ao desenvolvimento do projeto, PRESSMAN (1995).

Segundo PRESSMAN (1995), umas das principais atividades dos processos de gerenciamento de projetos de software é o planejamento. Quando o planejamento de um projeto é realizado o custo e tempo e prazo do projeto são definidos. E para isso é necessário planejar de forma organizada utilizando ferramentas e técnicas, bases históricas de projetos anteriores, e membros da equipe com conhecimento técnico que possam auxiliar na estimativa.

Há umas séries de técnicas de estimativas foram disponibilizados para o desenvolvimento de software cada uma com suas características, mas com algo em comum como:

- O escopo deve estar definido antes do inicio da estimativa;
- O histórico dos projetos anteriores é usado como base para estimativas de projetos que estão sendo iniciados;
- Os projetos são divididos em pequenas partes que não estimadas.

Em projetos com um escopo menor os as estimativas e o orçamento são realizados em mesmo período e por uma mesma pessoa. Mas no guia PMBOK, este processos são realizados de formas distintas, pois possuem ferramentas e técnicas diferentes.

Para criação de um plano de gerenciamento de custo são necessário criar os critérios para o planejamento, estruturação, estimativas, orçamentos e controle de custo do projeto. Os processos e ferramentas que serão realizados durante o projetos são definidos durante o ciclo

de vida do projeto e documentados no plano de gerenciamento de custo. O plano de gerenciamento de custo pode conter:

- Nível de exatidão: A partir das estimativas do custo, o dados terão um arredondamento em uma precisão, como \$100 ou \$1.000, baseado no escopo das atividades e tamanho do projeto;
- Unidade de medição para cada um dos recursos, como horas, dias ou meses;
- Códigos de conta de controle que possuem consistências de estimativas, orçamentos e controle de custo e que se conectam diretamente ao sistema de contabilidade da organização executora;
- Limites de variação para o monitoramento e desempenho de custo que podem ser especificados para indicar quando será necessário realizar alguma ação, caso o custo do projeto passe está variação;
- Definir regras para a medição de desempenho, como definir a EAP e os pontos de medições das contas de controles serão feitos, técnicas de medições de valor agregado a serem usadas, especificar as equações de cálculo do gerenciamento do valor agregado para determinar as previsões projetadas;
- A frequência e formatos dos relatórios que serão enviados durante o ciclo do projeto;
- A descrição de cada um dos processos do gerenciamento dos custos será documentada.

2.1 PLANEJANDO UM PROJETO

Segundo Pressman, o planejamento de projeto permite que o gerente faça estimativas razoáveis dos recursos, custos e cronogramas. As estimativas devem tentar definir os melhores de piores cenários, delimitado dentro do projeto. Conforme o projeto vai se desenvolvendo é necessário atualizar as estimativas para que o resultado chegue ao mais realista possível. Abaixo segue conjunto de tarefas para o planejamento do projeto no livro de PRESSMAN (1995):

1. Estabeleça o conjunto de tarefas.
2. Determine a viabilidade.
3. Análise de risco.
4. Defina recursos necessários.

- a. Determine recursos humanos necessários.
 - b. Defina recursos reusáveis de software.
 - c. Identifique recursos ambientais.
5. Estime custo e esforço.
 6. Decomponha o problema.
 7. Desenvolva duas ou mais estimativas usando tamanho, ponto por função, tarefas de processo ou casos de uso.
 8. Harmonize as estimativas.
 9. Desenvolva um cronograma do projeto.
 - a. Estabeleça um conjunto significativo de tarefas.
 - b. Defina uma rede de tarefas.
 - c. Use uma ferramenta de cronograma para desenvolver um diagrama de tempo.
 - d. Defina mecanismo de rastreamento de cronograma.

2.2 ESTIMATIVAS DE SOFTWARE

Segundo PRESSMAN (1995), no início os custos de software compreendia em uma pequena parcela do custo total de um sistema, e com isso um erro de estimativa não havia grande impacto no orçamento da empresa. Atualmente o cenário mudou, os projetos de softwares estão cada vez mais complexos e caros e uma estimativa errada pode fazer uma grande diferença entre prejuízo e lucro.

A medição de um esforço em um projeto de software jamais será uma ciência exata, pois existe uma grande variedade de variáveis que podem afetar o custo final do projeto como os recursos humanos, técnicas, fatores ambientais e políticas.

Para uma melhor estimativa de custo e esforço confiáveis, alguns pontos podem ser considerados.

- Adiarmos as estimativas até um ponto tardio de desenvolvimento do projeto, para que haja o máximo de informações para auxiliar nas estimativas;
- Usar técnicas de decomposição relativamente simples;
- Desenvolvimento de um modelo empírico para o custo e esforço do projeto;
- Usar uma ou mais ferramentas para estimativas automatizadas.

Cada uma das opções de estimativas de custo de software são mais eficazes se houverem dados históricos usados para fundamentar as estimativas. Se não houve uma base histórica a fase de levantamento repousará sobre uma fase muito instável.

2.3 VANTAGENS DE ESTIMAR O CUSTO

PRESSMAN (1995) afirma em seu livro, que a medição é algo muito comum na área de engenharia, como o consumo de energia, o peso, as dimensões físicas, a temperatura, a voltagem, a proporção de sinal entre outros. Mas no mundo da engenharia de software isso não acontece com tanta frequência, pois existe uma dificuldade em que medir e avaliar as medidas que são obtidas.

Conforme PARO (2005), apesar de estimar os custos requer um tempo, isso traz vantagens durante todo o desenvolvimento do projeto, como:

- Prever o esforço, recurso e custo projeto;
- Gerenciar contratos de softwares;
- Indicar qualidade do produto;
- Gerar baseline do custo;
- Monitorar o custo do projeto;
- Avaliar a produtividade das pessoas que produzem o produto;
- Avaliar os benefícios derivados de novos métodos e ferramentas de software;
- Auxilia na justificativa nos pedidos de ferramentas, recursos e treinamento para equipe.

2.4 PROBLEMAS EM ESTIMAR PROJETOS DE SOFTWARE

Existem alguns fatores que podem complicar a geração das estimativas em um projeto de softwares. No Congresso Internacional de Gestão de Tecnologia e Sistemas de Informação, realizado na USP, ROVAI (2004) cita alguns ponto levantados como:

- Complexidade e indefinições de escopo;
- Criticidades de prazos;
- Orçamentos com estimativas irrealistas ou não geradas por uma técnica confiável;
- Perda de prioridade;
- Falta de controle de escopo ou uma falta de gestão de alterações do projeto;

- Conflito entre prioridade e disputa de recursos de projetos concorrentes;
- Problemas com recurso não qualificados ou treinados;
- A não utilização de metodologias estruturadas para os processos de estimativas de do uso dos recursos.

Estes pontos podem afetar de forma significativa na estimativa final de um projeto. A utilização de metodologia, como o PMBOK, auxiliam para uma estimativa mais assertiva do custo e tempo de um projeto.

2.5 INFORMAÇÕES NECESSÁRIAS PARA ESTIMAR

Segundo o PMBOK (2008), existem algumas informações e artefatos que auxiliam nas estimativas dos projetos:

- Declaração do escopo do projeto: Um dos principais documentos necessário para a geração da estimativa do projeto. Este documento contém a descrição do produto o critério de aceite, as entregas chaves, os limites, as premissas e restrições do projeto. Uma vez definido o escopo deve ser analisado e alguns casos refinados para identificar a viabilidade dos projetos;
- Estrutura analítica do projeto: Relação de todos os componente e entregas;
- Dicionário da EAP: Relação detalhadas dos componentes e entregas de cada componente da EAP para necessário para produzir a entrega;
- Cronograma do projeto: O tipo e a quantidade dos recursos e a quantidade de tempo que esses recursos são necessários para completar a tarefa. Estas informações são primordiais para determinar o custo do projeto;
- Plano de recursos humanos: Atributo de recrutamento, índice de pessoal e reconhecimento / prêmios relacionados a entregas do projeto;
- Registros dos riscos: Devem ser registrados os riscos que ameaçam ou oportunidades que impacto o projeto de uma forma positiva ou negativa. O impacto negativo normalmente impacta em custo e podendo haver impacto no cronograma;
- Fatores ambientais da empresa: Os fatores ambientais da empresa influenciam no processo de estimativas de custo, que incluem: Condições de mercado e Informações comerciais publicadas;

- Ativo de processo organizacional: Existem alguns fatores ambientais da empresa que influenciam no processo de estimativa, como as condições de mercado, que são os serviços, produtos ou resultados que estão disponíveis, as condições de ofertas e demandas influenciam nos custos. As informações comerciais de taxa de custo de recurso e de matéria de equipamento, lista publicadas de preços e vendedores são também fontes de informações;

3 ESTIMATIVAS DE UM PROJETO

O início de um projeto de software inicia-se com um conjunto de atividades que é denominado de planejamento de projeto e umas das primeiras atividades é a realização das estimativas. Podemos citar Frederick Brooks, que é citado por PRESSAMN(1995, p.88):

“...nossas técnicas para realizar estimativas são mal desenvolvidas. Mais seriamente, elas refletem uma preocupação não expressa que é bastante inverídica, isto é, de que tudo saíra bem...uma vez que não temos certeza de nossas estimativas, frequentemente falta aos gerentes de software a gentil teimosia para fazerem as pessoas esperarem por um produto.”

As estimativas não devem ser feitas de forma aleatória, existem métricas e processos que ajudam a medir o tamanho de um projeto de software.

O guia PMI (2008) indicam algumas ferramentas e técnicas que nos auxiliam neste processo de medir o tamanho de um projeto.

3.1 ESTIMATIVAS ANÁLOGA

Este tipo de estimativa é usado quando existem poucas informações disponíveis para a estimativa do projeto. Este tipo de técnica usa como informações para estimar como escopo, custo, orçamento e duração ou medidas de escalas como tamanho e peso de um projeto anterior que seja semelhante da atual. Os valores brutos do projeto anterior são ajustados para um valor atual sabendo de algumas diferenças conhecidas.

Este tipo de estimativa é mais barata e rápida de que outros tipos de estimativas, mas o seu grau de assertividade é menos precisa. Estes tipos de técnicas só podem ser utilizados para projetos de um mesmo segmento ou projetos semelhantes que já foram realizados anteriormente, e tendo a necessidade de uma ajuda de pessoas especializadas que já participaram de outros projetos.

As Vantagens deste tipo de estimativa são:

- Normalmente é usada na fase inicial do projeto;
- Mais rápido para ser informado ao cliente;
- Útil para medir o interesse do cliente no projeto;
- Utilizada em projeto quando não temos informações detalhadas.
- As desvantagens deste tipo de estimativas são:
- Baseado em experiências anteriores;

- Deve ser realizado por uma especializada;
- Possuir grande risco ao estimar o custo do projeto;
- Estimativa menos precisa.

A estimativa por analogia também é conhecida como estimativa TOP-DOWN.

3.2 ESTIMATIVA PARAMÉTRICA

Segundo o PMBOK (2008, p. 128) a estimativa paramétrica é:

“A estimativa paramétrica é utiliza uma relação estatística entre dados históricos e outras variáveis (por exemplo, metros quadrados em construção) para calcular uma estimativa para parâmetros de atividade, como custo, orçamento e duração. Esta técnica pode produzir altos níveis de precisão dependendo da sofisticação e dos dados históricos usados no modelo. Estimativas paramétricas de custos podem ser aplicados a um projeto ou um segmento do mesmo, em conjunto com outros métodos de estimativas.”

Para determinar o custo de uma tarefa pode ser utilizada a quantidade do material multiplicado pelo valor do material, por exemplo, para construir uma parede com 50 tijolos que custa o R\$ 1,00 real cada tijolo o custo da tarefa será de R\$ 50,00 reais.

Em seu livro MELO (2012), cita que existem dois pontos a serem considerados neste tipo de estimativa, que é análise de regressão e curva de aprendizagem.

Análise de regressão é uma técnica estatística para prever quais poderão ser os valores futuros, com base em valores antigos. Está análise cria previsões quantitativas com bases em variáveis dentro de um valor, para prever variáveis em outro. Esta forma de estimativa conta unicamente com matemática estatística pura para revelar relacionamento entre variáveis e valores futuros.

Curva de aprendizagem considera que o custo de unidade cai conforme mais unidades são completadas na tarefa, isso porque os recursos aprendem enquanto completam o seu trabalho exigido. Quanto mais tarefa um recurso realiza, mais fácil se torna esta tarefa.

3.2 ESTIMATIVA “BOTTOM-UP”

Segundo o PMBOK a estimativa “BOTTOM-UP” é:

“A estimativa bottom-up é um métodos para estimar um componente do trabalho. O custo de pacotes de trabalho individuais ou atividades é estimado com o maior nível de detalhe especificado. O custo detalhado é então resumido nos níveis

mais altos e utilizado em subsequentes relatórios e rastreamento. O custo e a precisão da estimativa de custos bottom-up geralmente são influenciados pelo tamanho ou complexidade da atividade individual ou pacote de trabalho”.

Este tipo de estimativa é que possui o maior tipo de exatidão, mas para isso é necessário que o projeto esteja bem definido e todos os pacotes de entrega esteja definido na EAP. A partir de cada pacote criado é estimado o custo de cada um e somando de baixo para cima o valor de cada um, chegando ao custo total do projeto.

Segue um exemplo de EAP na Figura 01 para exemplificar a estimativa a partir de cada pacote de entrega:

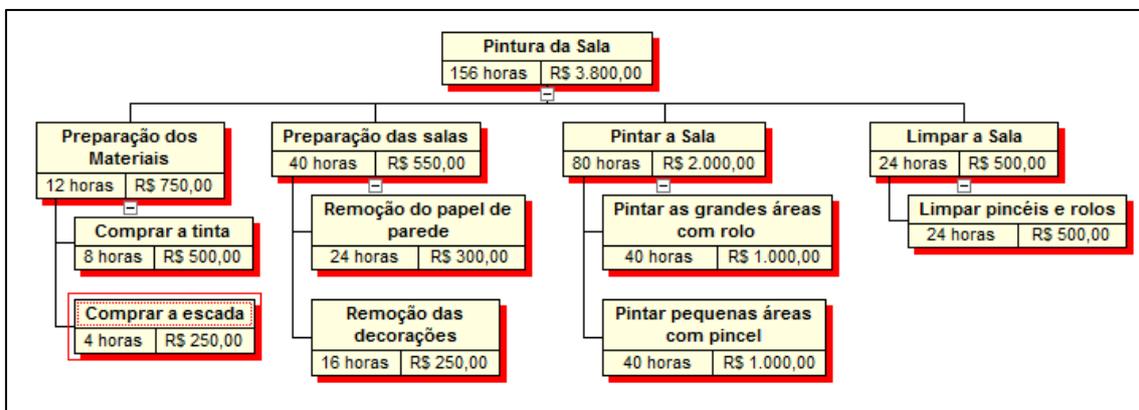


Figura 01 – EAP

Fonte: Baseado no artigo de JUNIOR (2010).

As vantagens deste tipo de estimativa são, conforme MACHADO (2008):

- Maior precisão;
- Desenvolve o comprometimento da equipe;
- Baseado na análise detalhada do projeto;
- Fornece a base para monitoramento e controle do projeto.
- As desvantagens deste tipo de estimativa são:
- Maior esforço de tempo, custo e recursos para desenvolver a estimativa;
- Tendência do time "inflar" as estimativas;
- Requer que o projeto esteja bem definido e entendido.

Na Tabela 01 segue o nível de exatidão de estimativa bottom-up a partir do nível

WBS:

| Precisão BOTTOM UP | | |
|--------------------|-------------------------|------------------------|
| Nível do WBS | Projetos de Baixo Risco | Projetos de Alto Risco |
| 1 | +/- 35% | +/- 75%-100% |
| 2 | 20% | 50-60% |
| 3 | 10% | 20-30% |
| 4 | 05% | 10-15% |
| 5 | 02% | 05-10% |

Tabela 01 – Precisão Bottom-UP.
Fonte: MARICONI (2011, s.p.).

As três estimativas apresentadas, Modelo Paramétrico, estimativa análoga e estimativa bottom-up são as mais utilizadas em projetos e conforme maior nível de exatidão de cada uma maior o tempo para estimar o projeto.

Na Tabela 02 são exibidas as características de cada estimativa:

| Tipo de estimativa | Relação com a EAP | Tempo de preparo | Precisão |
|--------------------|-------------------|------------------|--------------|
| Paramétrica | Top Down | Dia | de 25% a 75% |
| Analóga | Top Down | Semana | de 10% a 25% |
| Bottom Up | Bottom Up | Mês | de 5% a 10% |

Tabela 02 – Tipo de estimativas.
Fonte: MARICONI (2011, s.p.).

3.4 OUTRAS ESTIMATIVAS SEGUNDO O PMBOK

Existem outras estimativas que são indicadas no guia PMBOK (2008), 4ª edição que são:

- **Opinião especializada:** Os custos são influenciados por muitas variáveis como taxa de mão de obra, custos de materiais, inflação, fatores de riscos entre outras. A opinião de uma pessoa que já passou por um projeto parecido fornece informações que auxiliam na estimativa do projeto. Pode também ser usada para determinar quais métodos devem ser combinados e como reconciliar as diferenças entre eles;
- **Estimativa de três pontos:** Esta é uma técnica probabilística para estimativa de duração das atividades baseada em ponderação de três pontos. Este técnica também é conhecida como PERT (Técnica de revisão e avaliação de programa). Para chegar em uma estimativa mais aproximada são usadas três estimativas:
- **Mais provável (cm):** é o esforço total mais realista para a realização das tarefas e quaisquer outros gastos previstos;

- **Otimista** (co): é o esforço baseado no melhor cenário para a atividade;
- **Pessimista** (cp): é o esforço baseado no pior cenário para a atividade.

A estimativa PERT utiliza as três estimativas acima e aplica a uma média ponderada que é representada na formula abaixo:

$$ce = (co + 4cm + cp) / 6$$

Com custos baseado em uma equação, a estimativa pode fornecer um resultado com uma maior precisão.

- **Análise reservas:** Nas estimativas pode ser incluídos reservas de contingencias para considerar os custos das incertezas. A reserva pode ser uma porcentagem do custo total ou um valor fixo desenvolvido por métodos de análise quantitativa. Conforme o andamento do projeto e mais informações são obtidas a reserva de contingência pode ser usada, reduzida ou eliminada;
- **Software para estimativas em gerenciamento de projeto:** Aplicativos para estimativas de projetos, planilhas computadorizadas, simulações e ferramentas estatística estão se tornando amplamente aceitos na ajuda das estimativas;
- **Análise da proposta do fornecedor:** segundo o PMBOK (2008), a estimativa é baseada na cotação do fornecedor. Quando projetos são concedidos a um vendedor sob processos competitivos, um trabalho adicional de estimativa de custo pode ser requisitado da equipe do projeto para examinar os preços das entregas individuais e derivar a custo total final do projeto.

3.4 TIPOS DE MÉTRICAS DE SOFTWARES

Em seu livro Pressman (2002, p. 265) define como métrica de software da seguinte forma:

“Métricas de processo e de projeto de software são medidas quantitativas que permitem que os engenheiros de software tenham ideia da eficácia do processo de software e dos projetos que são conduzidos usando o processo como arcabouço. Dados básicos de qualidade e de produtividade são coletados. Esses dados são analisados, comparados com médias anteriores e avaliados e avaliados para determinar se ocorreram melhorias de qualidade e produtividade. Métricas também são usadas para detectar área de problemas, de modo que soluções possam ser desenvolvidas, e que o processo de software possa ser melhorado.”

As métricas em um projeto podem ser feitas definindo um conjunto limitado de medições que são fáceis de coletar, onde essas são normalizadas usando métricas orientadas a tamanho ou a função. Os resultados podem ser comparados análises anteriores ou com projetos semelhantes. As métricas são analisadas e avaliadas por gerentes em quantos as medidas são coletadas pelos engenheiros de software. As partir das análises feitas será possível um entendimento melhor do projeto.

Vamos analisar algumas métricas de software como métricas orientadas a tamanho, Métricas orientada a função, Métricas orientada a objetos, Métricas orientada a Caso de usos, Métrica de qualidade de software.

3.4.1 MÉTRICAS ORIENTADAS AO TAMANHO

Métricas orientadas ao tamanho podem ser formadas a partir de alguns registros simples de informações e processos de projetos anteriores para poder medir de uma forma direta o tamanho de projetos futuros. A partir destas informações poderá ser criada uma tabela de dados orientados ao tamanho, como segue o exemplo da tabela 3.

A tabela relaciona cada projeto de software que foi desenvolvido e concluído nos últimos anos. Ao consultar a Tabela 03 o projeto aaa-01 foram geradas 12.1 KLOC (mil linhas de código), com 24 pessoas-mês de esforço a um custo de 168 mil dólares. As informações apresentadas na tabela reapresentam todas as atividades de engenharia de software e não apenas a codificação, conforme cita Pressman (1995, p. 62).

A partir dos dados armazenados na tabela, um conjunto de métricas de qualidade e de produtividade orientado ao tamanho pode ser obtido:

- Produtividade = KLOC pessoa-mês;
- Qualidade = defeitos/KLOC;
- Custo = \$/LOC;
- Documentação = páginas de documentação/KLOC.

| Projeto | Esforço | \$ | KLOC | págs. Docum. | Erros | Pessoas |
|---------|---------|-----|------|-----------------|-------|---------|
| aaa-01 | 24 | 168 | 12.1 | 365 | 29 | 3 |
| ccc-04 | 62 | 440 | 27.2 | 1224 | 86 | 5 |
| fff-03 | 43 | 314 | 20.2 | 1050 | 64 | 6 |

Tabela 03 – Métricas orientadas ao tamanho

Fonte: PRESSMAN (1995, p. 63).

As métricas orientadas ao tamanho não é considerada a melhor forma de medir um tamanho de um projeto de software. A maior desvantagem está em usar as linhas de códigos com uma medida chave para a medição. Existem outras desvantagens como, penalizam programas bem projetados, não acomodam linguagens não procedimentais e seu planejamento requer um nível detalhado que é difícil de obter em uma fase de análise.

3.4.2 MÉTRICAS ORIENTADAS A FUNÇÃO

Métricas orientadas a função são medidas indiretas de softwares e por meio de qual ele é desenvolvido, ao invés de contar a linhas de códigos a medição é concentrada nas funcionalidades ou utilidades do software. A primeira proposta da utilização de pontos-por-função (PFs), foi feito por Albrecht, conforme Pressman (1995, p. 64), que usa uma relação empírica baseado em medidas de informações e complexidade do software.

Para a contagem dos pontos por funções, cinco características são necessárias para a computação dos pontos, que são:

- Número de entrada do usuário: Cada entrada de informação distinta que o usuário inclua no sistema é contada;
- Número de saídas do usuário: Cada saída que proporcione informações para o usuário, como relatórios, tela, mensagens entre outros são contados;
- Número de consulta: Cada entrada on-line que gera uma resposta de forma online para o usuário é considerada uma forma de consulta;
- Número de arquivos: Cada acesso há um arquivo lógico ou um agrupamento de informações é contado;
- Número de interfaces externas: Todas as interfaces que sejam usadas para transmitir informações para outros sistemas são contadas.

As partir destas informações pode ser utilizado uma tabela que segue na figura 02, para computar o total de pontos:

| Parâmetros de Medida | Contagem | Fator de Ponderação | | | | | |
|--------------------------------|----------|---------------------|-------|----------|----|---|--|
| | | Simples | Médio | Complexo | | | |
| Números de entrada do usuário | | X | 3 | 4 | 6 | = | |
| Números de saída do usuário | | X | 4 | 5 | 7 | = | |
| Números de consulta do usuário | | X | 3 | 4 | 6 | = | |
| Número de arquivos | | X | 7 | 10 | 15 | = | |
| Número de interface externas | | X | 5 | 7 | 10 | = | |
| Contagem Total | | | | | | | |

Figura 02: Computando a métrica pontos-por-função.

Fonte: PRESSMAN (1995, p. 65).

E comum ser utilizado um fator de ponderação para cada ponto de função calculado, como simples, médio e complexo, mas ainda sim a determinação é tanto quanto objetiva.

Existem também uma relação entre linhas de códigos e pontos por função para implementar o software e a qualidade do projeto, na figura 03 segue o números de linha de código necessários para construir um ponto por função, em várias linguagens de programação:

| Linguagem de Programação | LOC por Ponto por Função | | | |
|--------------------------|--------------------------|---------|-------|------|
| | Média | Mediana | Baixa | Alta |
| Access | 35 | 38 | 15 | 47 |
| Ada | 154 | — | 104 | 205 |
| APS | 86 | 83 | 20 | 184 |
| ASP 69 | 62 | — | 32 | 127 |
| Assembler | 337 | 315 | 91 | 694 |
| C | 162 | 109 | 33 | 704 |
| C++ | 66 | 53 | 29 | 178 |
| Clipper | 38 | 39 | 27 | 70 |
| COBOL | 77 | 77 | 14 | 400 |
| Cool:Gen/IEF | 38 | 31 | 10 | 180 |
| Culprit | 51 | — | — | — |
| DBase IV | 52 | — | — | — |
| Easytrieve+ | 33 | 34 | 25 | 41 |
| Excel47 | 46 | — | 31 | 63 |
| Focus | 43 | 42 | 32 | 56 |
| FORTRAN | — | — | — | — |
| FoxPro | 32 | 35 | 25 | 35 |
| Ideal | 66 | 52 | 34 | 203 |
| IEF/Cool:Gen | 38 | 31 | 10 | 180 |
| Informix | 42 | 31 | 24 | 57 |
| Java | 63 | 53 | 77 | — |
| JavaScript | 58 | 63 | 42 | 75 |
| JCL | 91 | 123 | 26 | 150 |
| JSP | 59 | — | — | — |
| Lotus Notes | 21 | 22 | 15 | 25 |
| Mantis | 71 | 27 | 22 | 250 |
| Mapper | 118 | 81 | 16 | 245 |
| Natural | 60 | 52 | 22 | 141 |
| Oracle | 30 | 35 | 4 | 217 |
| PeopleSoft | 33 | 32 | 30 | 40 |
| Perl | 60 | — | — | — |
| PL/1 | 78 | 67 | 22 | 263 |
| Powerbuilder | 32 | 31 | 11 | 105 |
| REXX | 67 | — | — | — |
| RPG II/III | 61 | 49 | 24 | 155 |
| SAS | 40 | 41 | 33 | 49 |
| Smalltalk | 26 | 19 | 10 | 55 |
| SQL | 40 | 37 | 7 | 110 |
| VBScript36 | 34 | 27 | 50 | — |
| Visual Basic | 47 | 42 | 16 | 158 |

Figura 03 – LOC por ponto-por-função.
 Fonte: PRESSMAN (2005, p. 505).

3.4.3 MÉTRICAS ORIENTADAS A OBJETO

Projetos de software que utiliza a programação orientada a objeto podem até utilizar métricas convencionais, no entanto não fornece granularidade suficiente para ajustes de cronogramas e esforço necessários durante o projeto, conforme Pressman (2005, p. 506).

Em seu artigo Borges, (2010, p. 18), explica que para uma melhor estimativa para projetos que utiliza a orientação a objeto é necessário ter uma abordagem diferente, tanto para o desenvolvimento quanto para as métricas utilizadas. Existem várias propostas para métricas a orientação a objeto que levam em considerações algumas características como Pressman (2005, p. 507) cita e explica:

- Número de scripts de cenário (*number of scenario scripts*): Sequencia de passos detalhados que descreve a interação entre o usuário e o aplicação.;
- Número de classes chaves (*number of keys classes*): são os componentes independentes.;
- Número de classes de apoio (*number of support classes*): São classes que auxiliam para o desenvolvimento do sistema, mas não está relacionado diretamente na solução que o aplicativo fornece;
- Número médio de classes de apoio por classe chaves (*average number of support classes per key class*): Estas classes são definidas durante o projeto.
- Número de subsistema (*number of subsystems*): è uma agregação de classes que usam função que é visível ao usuário final do sistema.

As métricas OO (orientado a objeto) podem ser divididas em duas categorias, medidas relacionadas com processos e medidas relacionadas com produto. As métricas relacionadas ao processo são usadas para mensurar o processo e o status do processo. O início da coleta destas informações deve ser tão metodicamente quanto possível. Segue alguns exemplos de métricas relacionadas com processo, segundo BORGES (2010, p. 18):

- Tempo total de desenvolvimento;
- Tempo de desenvolvimento em cada processo e subprocesso;
- Tempo utilizado modificando modelos de processos anteriores;
- Tempo gasto em todos os tipos de subprocessos como: especificação dos casos de uso, desenho do bloco, teste do bloco e do caso de uso para cada objeto;
- Número de diferentes tipos de falhas encontrados durante revisões;
- Número de mudanças propostas nos modelos anteriores;
- Custo da garantia de qualidade;
- Custo para introduzir novas ferramentas e processo de desenvolvimento.

Estas medições podem ser utilizadas para medições de projetos futuros.

As métricas relacionadas com produtos são utilizadas para controlar a qualidade do produto final e são aplicadas ao sistema ainda em construção. As métricas relacionadas ao produto estão divididas em três categorias:

- Métrica de análise;
- Métrica de projeto;
- Métrica de construção.

As métricas de projetos servem para mensurar aspectos importantes do sistema e além de ser fácil a coleta das informações.

3.4.4 MÉTRICAS ORIENTADAS A CASO DE USO

O caso de uso descreve as funções e característica visíveis aos usuários que são os requisitos básicos do sistema, e independe da linguagem de programação usado, além de possuir a mesma proporção de casos de testes e tamanho de LOC da aplicação. Como o caso de uso é definido logo no início do projeto ele pode ser usado como uma ferramenta de estimativa para o projeto (PRESSMAN, 2005, p. 507).

Os Pontos de Casos de Uso (PCU) foram criados em 1993 por Gustav Karner da empresa Objectory AB. A contagem desta métrica segue os seguintes passos, conforme MEDEIROS (2004) e citado por HEIMBERG; GRAHL (2005, s.p.).

1. Relacionar os atores, classificá-los de acordo com seu nível de complexidade (simples, médio ou complexo) atribuindo respectivamente os pesos 1, 2 ou 3, conforme a Tabela 04. Calcule o TPNA (Total de Pesos não Ajustados dos Atores) somando os produtos da quantidade de atores pelo seu peso.

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

Tabela 04: Classificação de atores
Fonte: MEDEIROS (2004).

2. Contar os casos de uso e atribuir o grau de complexidade sendo a complexidade baseada no número de classes e transações. Calcule o TPNAUC (Total de Pesos não ajustados dos casos de usos) somando os produtos da quantidade de casos de usos pelo respectivo peso conforme a tabela 05.

| Tipo de Caso de Uso | Descrição | Peso |
|---------------------|--|------|
| Simple | Considerar até 3 transações com menos de 5 classes de análise | 5 |
| Médio | Considerar de 4 a 7 transações com 5 a 10 classes de análise | 10 |
| Complexo | Considerar de 7 transações com pelo menos de 10 classes de análise | 15 |

Tabela 05: Classificação dos Casos de Uso

Fonte: MEDEIROS (2004).

3. Calcular PCU's não ajustados, também chamados de PCUNA, de acordo com a seguinte fórmula:

$$PCUNA = TPNA + TPNAUC$$

4. 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 que a grau não está presente ou não é influente, 3 influência média e o valor 5 indica influência significativa através de todo o processo. Após determinar o valor dos fatores, multiplicar pelo respectivo peso ilustrado na Tabela 06, somar o total e aplicar a seguinte fórmula:

$$FCT = 0.6 + (0.01 * \text{Somatório do Fator técnico})$$

| Descrição | Peso |
|--|------|
| Sistemas Distribuídos | 2,0 |
| Desempenho da aplicação | 1,0 |
| Eficiência do usuário final (on-line) | 1,0 |
| Processamento interno complexo | 1,0 |
| Reusabilidade do código em outras aplicações | 1,0 |
| Facilidade de instalação | 0,5 |
| Usabilidade (facilidade operacional) | 0,5 |
| Portabilidade | 2,0 |
| Facilidade de manutenção | 1,0 |
| Concorrência | 1,0 |
| Características especiais de segurança | 1,0 |
| Acesso direto para terceiros | 1,0 |
| Facilidades especiais de treinamento | 1,0 |

Tabela 06 – Fatores de complexidade técnica
 Fonte: MEDEIROS (2004).

5. Determinar o fator de complexidade ambiental: os fatores de complexidade ambientais indicam a eficiência do projeto e estão relacionados ao nível de experiência dos profissionais. Esses fatores descritos na tabela 4 são determinados através da escala de 0 a 5, onde 0 indica baixa experiência, 3 indica média experiência e 5 indica alta experiência. Após determinar o valor de cada fator, multiplicar pelo peso e somar o total dos valores. Em seguida, aplicar a seguinte fórmula:

$$FCA = 1,4 + (-0,03 * \text{Somatório do Fator Ambiental})$$

6. Calcular os PCU's ajustados: esse cálculo é realizado com base na multiplicação dos PCU não ajustados, na complexidade técnica e na complexidade ambiental através da seguinte fórmula:

$$PCUA = PCUNA * \text{Fator de complexidade técnica} * \text{Fator de complexidade ambiental}$$

| Fator | Descrição | Peso |
|-------|--|------|
| F1 | Familiaridade com o processo de desenvolvimento de software | 1,5 |
| F2 | Experiência na aplicação | 0,5 |
| F3 | Experiência com OO, na linguagem e na técnica de desenvolvimento | 1,0 |
| F4 | Capacidade do líder de análise | 0,5 |
| F5 | Motivação | 1,0 |
| F6 | Requisitos estáveis | 2,0 |
| F7 | Trabalhadores com dedicação parcial | -1,0 |
| F8 | Dificuldade da linguagem de programação | -1,0 |

Tabela 07 – Fatores de complexidade ambiental.

Fonte: MEDEIROS (2004).

7. Calcular a estimativa de horas de programação. Karner, o criador da estimativa, sugere a utilização de 20 pessoas-hora por unidade de PCU. Schneider e Winters sugerem o seguinte refinamento:
- X = total de itens de F1 a F6 com pontuação abaixo de 3;
 - Y = total de itens de F7 a F8 com pontuação acima de 3;
 - Se $X + Y \leq 2$, usar 20 como unidade de homens/hora;
 - Se $X + Y = 3$ ou $X + Y = 4$, usar 28 como unidade de homens/hora;
 - Se $X + Y \geq 5$, deve-se tentar modificar o projeto de forma a baixar o número, pois o risco de insucesso é relativamente alto.

$$\text{Estimativa de horas} = \text{PCUA} * \text{pessoas hora por unidade de PCU}$$

3.4.5 MÉTRICAS DE PROJETO DE ENGENHARIA DA WEB

Todo projeto de engenharia WEB tem como finalidade entregar para o usuário final uma combinação de conteúdo e funcionalidades. As métricas tradicionais são difíceis de aplicar para os projetos de WEB, por isso é necessário que as organizações que desenvolvem estes tipos de projetos desenvolvam um banco de dados que armazena informações que possa avaliar a produtividade e qualidade do projeto. Conforme PRESSMAN (2005, p. 507) seguem algumas medidas que devem ser coletadas:

- Número de páginas estáticas da WEB (number of static Web Pages);
- Número de páginas dinâmicas da WEB (number of dynamic Web Pages);
- Número de links internos da página (number of internal links);
- Número de objetos de dados persistentes (number of persistent data objects);

- Número de sistemas externos interfaceados (number of external system interfaced);
- Número de objetos de conteúdo estático (number of static content objetos);
- Número de objetos de conteúdo dinâmico (number of dynamic content objetos);
- Número de funções executáveis (number of executable functions).

4 METODOLOGIAS DE ESTIMATIVAS PARA TI

Um das dificuldades que os projetos de TI enfrentam são medir o tamanho do projeto que está sendo desenvolvido junto com a sua capacidade de produção para que os objetivos estejam alinhados com os objetivos da empresa, CALAZANS (2004).

Citado por CALAZANS (2004, p. 1), TAURION ressalta que o sucesso do uso de TI está no do grau de alinhamento entre objetivos da área de TI com os objetivos da área de negócio.

Como a área de TI necessita identificar sua capacidade produtiva presente e futura para que possa entender, controlar e aperfeiçoar o processo de produção de software e consequentemente alinhar com os objetivos do projeto com os objetivos da estratégico da empresa iremos apresentar duas ferramentas muitos utilizados na área de TI para medição de projeto que é Análise por pontos de função e COCOMO II.

4.1 COCOMO II

A metodologia COCOMO (*Constructive Cost Model*) II é um método que busca medir esforço, prazo, tamanho de equipe e custo necessário para o desenvolvimento do software, desde que se tenha a dimensão do mesmo, através de um modelo de estimativa de tamanho de software conforme LOPEZ(2005).

O precursor do COCOMO II foi o modelo COCOMO, conhecido também como COCOMO 81, mas como foi considerado obsoleto o modelo foi trocado em 2000.

Podemos definir os objetivos primários do COCOMO II, segundo LOPEZ (2005) como:

- Um modelo de estimativa de custo para o desenvolvimento de software de acordo com o modelo de ciclo de vida de software vista em Jones (1986), e as práticas de desenvolvimento da última década;
- Criar ferramentas de suporte capazes de fornecer melhoramentos do modelo, através de manutenção de informações sobre o desenvolvimento de software em uma estrutura de base de dados.
- Fornecer um framework analítico, e um conjunto de ferramentas e técnicas para avaliação dos efeitos de melhoria na tecnologia e nos custos despendidos no ciclo de vida de desenvolvimento de software.

O COCOMO II aplicado ao RUP estima o esforço, prazo e equipe média para as fases de elaboração e construção. As fases de “Inception” e “Transition” são estimadas como percentuais da soma elaboração + construção. Os marcos que caracterizam o escopo abrangido pelo CII, conforme AGUIAR (2010) são:

- LCO (*Lyfe Cycle Objectives*): Ponto no qual é escolhida uma possível arquitetura para o projeto (não necessariamente aquela que será de fato utilizada). Ocorre ao final da Inception.
- IOC (*Initial Operational Capability*): Ponto no qual é concluído o desenvolvimento do software, estando o sistema pronto para entrega e teste final.

Depois de estabelecido o escopo é definido o esforço (PM, *Persons-Month*), em pessoas-mês, na seguinte formula:

$$PM = A * (Size^E) * \text{Produtório } (i=1 \text{ até } n, EM(i))$$

Onde E é definido pela seguinte formula:

$$E = B + 0,001 * \text{Somatório } (j=1 \text{ até } 5, SF(j))$$

E onde as seguintes variáveis são:

- A: Constante que deve ser calibrada para o ambiente, a partir de dados históricos;
- Size: Tamanho do software em LOC. Caso o tamanho seja dado em Pontos de Função, estimar a quantidade de LOC a partir de tabelas de backfiring [8], que permitem converter Pontos de Função para LOC;
- EM(i): Effort Multipliers: Até 17 fatores que irão adequar o modelo ao projeto específico;
- E: Expoente do Esforço;
- B: Constante que deve ser calibrada a partir de dados históricos.
- SF(j) (*Scale Factors*): Cinco fatores que irão ajustar o expoente do Esforço, de modo a adequar a fórmula a um projeto específico.

O Prazo que é calculo em meses é definido pela seguinte formula:

$$TDEV = C * PM^F$$

Onde:

$$F = D + (0,2 * 0,01) * \text{Somatório } (j=1 \text{ até } 5, SF(j))$$

As variáveis “C” e “D” devem ser calibradas a partir de dados históricos e a equipe média é obtida através da divisão de Esforço pelo Prazo.

A calibração do modelo COCOMO II foi originalmente calibrado com dados de 161 projetos dentre de mais 2000 projetos candidatos. Embora o modelo possa ser executado com parâmetros nominais o correto é utilizar dados históricos para o ambiente que será desenvolvido utilizando projetos semelhantes, segundo AGUIAR (2010).

A seguir segue um roteiro simplificado de como obter uma estimativa com COCOMO II no início do projeto, utilizando o software USC CII:

1. Calibrar o CII para a base histórica de projetos obtida, salvando os coeficientes obtidos em um modelo;
2. Caso a organização-alvo não trabalhe com LOC, estimar o tamanho do aplicativo em Pontos de Função, utilizando o método da NESMA, obtendo a quantidade de LOC a partir de backfiring, utilizando a tabela contida no USC CII;
3. Determinar os 7 Effort Multipliers do Early Design Model e entrar com eles no USC CII;
4. Determinar o Esforço, o Prazo e a Equipe Média;
5. Obter a distribuição do Esforço e Prazo por fase do projeto (Inception,Elaboration, Construction e Transition, se for utilizado o RUP – notar que o USC CII também trabalha com o modelo Waterfall - Cascata).

Segundo AGUIAR (2010) o modelo COCOMO II utiliza de modelos paramétricos que faz tratamentos matemáticos para obtenção de estimativas de projeto, no entanto é necessária a preparação adequada do modelo e de cuidados estudos para que a estimativa seja eficiente para projetos de software.

4.2 PONTOS POR FUNÇÃO

A métrica ponto por função (*Function Point* – FP), foi inicialmente proposta por Albrecht , e é a técnica que mede o tamanho funcional de um software, a partir de uma perspectiva funcional e independente da tecnologia adotada conforme REINALDO (2012). Independentemente da linguagem de programação é possível medir o tamanho do esforço do projeto de software apenas pelas funções que o software possui. Pontos de função (FP) é a unidade de medida.

Conforme JENNY (2011), pontos por função pode ser usado para:

- Estimar o custo ou esforço necessário para projetar, codificar e testar software;
- Prever o número de erros que vão ser encontrados durante o teste;

- Prever o número de componentes e/ou número de linhas de código projetadas no sistema implementado.

Ainda citado por JEENY (2011), uma vez estabelecidos os limites e o escopo da contagem é necessário classificar as funções do tipo dado, que podem ser:

- Número de Entradas: uma entrada externa processa dados ou processa informações de controle que entram pela fronteira da aplicação. Esses dados, através de um processo lógico único, atualizam Arquivos Lógicos Internos. Exemplo: Tela de Entrada on-line (Contar uma entrada para cada função de manutenção como inclusão, alteração e exclusão).
- Número de Saídas: cada saída do usuário que proporcione informações orientadas à aplicação. (Relatórios, mensagens de erro, etc);
- Número de Consultas: uma consulta é definida como uma entrada on-line que resulte na geração de alguma resposta. Exemplo: Telas de alteração ou remoção de dados, que mostram o que será alterado ou removido antes de sua ação efetiva, devem ser consideradas como Consultas Externas, Telas de seleção de relatórios que permitem informar parâmetros para o relatório escolhido, devem ser consideradas como sendo a parte de entrada de uma Consulta Externa;
- Número de arquivos internos: cada arquivo-mestre lógico, isto é, um agrupamento lógico de dados, que pode ser uma parte de um grande banco de dados ou um arquivo convencional, é contado. Exemplo: tabela de clientes, tabela de usuários, etc;
- Número de interfaces externas: um Arquivo de Interface Externa é um grupo de dados logicamente relacionados ou informações de controle especificadas pelo usuário, que é utilizado pela aplicação, mas sofre manutenção a partir de outra aplicação.

Assim que as informações foram coletadas a Tabela 08 pode ser preenchida com cada contagem e seu nível de complexidade:

| Parâmetro de Medida | Contagem | Fator de Ponderação | | | | |
|-------------------------------|----------|---------------------|-------|----------|---|--|
| | | Simple | Médio | Complexo | | |
| Número de Entradas | | x 3 | 4 | 6 | = | |
| Número de Saídas | | x 4 | 5 | 7 | = | |
| Número de Consultas | | x 3 | 4 | 6 | = | |
| Número de Arquivos | | x 7 | 10 | 15 | = | |
| Número de Interfaces Externas | | x 5 | 7 | 10 | = | |
| Contagem Total | | | | | | |

Tabela 08: Tabela de complexidade

Fonte: JEENY (2011)

É também necessário calcular os fatores de ajustes (F_i) para o pontos de funções, para isso é necessário responder algumas perguntas como:

- O sistema requer backup e recuperação confiáveis?
- São exigidas comunicações de dados?
- Há funções de processamento distribuídas?
- O desempenho é crítico?
- O sistema funcionará num ambiente operacional existente, intensamente utilizado?
- O sistema requer entrada de dados on-line?
- A entrada de dados on-line exige que a transação de entrada seja elaborada em múltiplas telas ou operações?
- Os arquivos-mestres são atualizados on-line?
- A entrada, saída, arquivos ou consultas são complexos?
- O processo interno é complexo?
- O código foi projetado de forma a ser reusável?
- A conversão e a instalação estão incluídos no projeto?
- O sistema é projetado para múltiplas instalações em diferentes organizações?
- A aplicação é projetada de forma a facilitar mudanças e o uso pelo usuário?

Para pergunta respondida é usada uma escala onde:

0 – Sem influência;

1 – Incidental;

2 – Moderado;

3 – Médio;

4 – Significativo;

5 – Essencial;

E para calcular os Pontos de Função é usada a seguinte fórmula: $FP = \text{Contagem Total} \times (0,65 + 0,01 \times \sum (fi))$. E para chegar a um valor de FP , deve estimar-se a quantidade de horas por FP.

Para a produtividade é usada à fórmula: $\text{Produtividade} = FP/\text{pessoa mês}$.

Para a qualidade é usada à fórmula: $\text{Qualidade} = \text{defeitos}/FP$.

Para o custo é usada à fórmula: $\text{Custo} = \text{CUSTO}/FP$.

Para a documentação é usada a formula: $\text{Documentação} = \text{páginas de documentação}/FP$.

5 DETERMINANDO ORÇAMENTO

Segundo o guia PMBOK (2008), determinar o orçamento do projeto é:

“...é o processo de agregação dos custos estimados de atividades individuais ou pacotes de trabalhos para estabelecer uma linha de base dos custos autorizados. Essa linha de base inclui todos os orçamentos autorizados, mas exclui as reservas de gerenciamento.”

Na criação do orçamento do projeto tem que estar todos os processos necessários para a realização do projeto, o custos de todos os recursos para a implementação do projeto e este custo deve estar autorizado pelo(s) financiadores do projeto e não deve ultrapassar o orçamento definido, conforme citado por COSTA (2011, p. 105).

Segundo o PMBOK(2008), existem algumas informações para o auxílio da definição do orçamento, que segue abaixo:

- Estimativas de custos da atividade: Para cada atividades do projeto definir um custo;
- Bases das estimativas: Quaisquer premissas básicas sobre a inclusão ou exclusão de custos indiretos ou indiretos no orçamento do projeto são especificados na base das estimativas;
- Declaração do escopo: Limitações por gasto dos recursos financeiros podem ser exigidas por contrato;
- EAP e dicionário da EAP: Fornece todos os entregáveis do projetos e seus componentes;
- Cronograma do projeto: Fornece as datas e marcos de entrega do projeto, auxiliando para agregar custos nos períodos definidos;
- Calendário dos recursos: Usado para indicar os custos de um recurso em um período;
- Contratos: Custos relacionados a produtos, serviços ou resultados que foram comprados são incluídos no orçamento.
- Ativos de processos organizacionais: Os ativos da empresa influenciam para determinar o orçamento do projeto.

Em seu livro, HELDMAN(2005, p. 183) cita também alguns itens mais comuns que aparece na lista de orçamento de um projeto e que pode ser usados em projetos futuros. Segue a lista apresentada por ela abaixo:

- Salario da equipe do projeto;

- Gastos com equipamentos e materiais;
- Custo de aluguel e instalações;
- Custo de marketing, incluindo grupo de foco de custo de pesquisa de mercado;
- Custos legais;
- Custos de viagens;
- Custos de propagandas;
- Custos de pesquisas;
- Custos de estudo de viabilidade;
- Serviço de consultoria para especialistas no assunto ou participantes do projeto;
- Telefonema, fax e taxa de longa distância;
- Suprimentos para escritórios;
- Taxas de acesso a internet ou taxa de hospedagem de websites;
- Programas de computador;
- Equipamentos;
- Treinamentos.

5.1 DETERMINANDO ORÇAMENTO EM PROJETOS DE TI

PHILLIPS (2002, p. 94) também cita alguns cuidados que o gerente deve tomar cuidado no momento em que vai orçar um projeto de software. Principalmente na área de TI existe uma grande variação de informações que muda a todo o momento como novas informações de software, tamanho de disco rígido, velocidade de processadores, novas tecnologia no mercado e entre outros. O gerente de projeto tem que estimar o tempo com essas mudanças e prever a curva de aprendizado que a equipe às vezes necessita para aprender algo novo, e o custo do projeto pode aumentar com a necessidade de cursos para a equipe, por exemplo.

Outra preocupação que um gerente de projetos de TI precisa se preocupar são os componentes de hardware, que normalmente com novas tecnologias são inseridas no mercado, o valor de das tecnologias antiga vai diminuindo, no entanto, existe momento que a procura torna-se maior que a oferta e o preço aumentam novamente e isso por atrapalhar no orçamento final do projeto. Para isso o gerente deve detalhar o máximo o orçamento do projeto e criar fases para que facilite a criação do orçamento.

Quando iniciar a criação do orçamento o gerente de projeto deve levar em contas algumas questões como:

- Dividir o projeto em fases: Quando segmenta o projeto inteiro em fases é mais fácil atribuir os valores e o volume de trabalho;
- Considerar a fase de integração: Quando se prepara o ambiente de produção para o início da implementação, as questões orçamentárias podem incluir tempo ocioso, intervalos, horas de trabalho necessárias e o tempo que o gerente de projeto precisa para verificar que as tarefas estão sendo concluídas de modo a manter o projeto dentro do cronograma;
- Considerar a carga horária total necessária para concluir cada fase do projeto: Carga de horas total para que a equipe consiga realizar cada fase do projeto;
- Considerar qualquer custo especializado: Qualquer consultoria prestada ou curso necessário para o projeto deve ser levado em conta;
- Considerar custo de equipamento: qualquer equipamento que deve ser comprado para a realização do projeto deve ser orçado no custo total do projeto;
- Considerar o custo de produção: Qualquer custo adicional com despesas como criação de manuais de implementação e manuais de usuários, projetos de página WEB e desenvolvimento.

Considerando estas questões será possível calcular as despesas para cada fase e poderá criar uma evolução de despesas que será usado para cada fase. Por exemplo, para terminar uma fase de um projeto será necessário:

- Comprar componentes de hardware;
- Comprar software;
- Licenciamento;
- Despesas com consultores;
- Alocação de desenvolvedores internos.

Desta forma é possível criar uma tabela com o orçamento de cada fase do projeto e chegando a um total do projeto.

O PMBOK (2008) cita algumas ferramentas e técnicas para determinar o orçamento de um projeto, que são:

- Agregação de custo;
- Análise de reservas;
- Opinião especializada;

- Relações Históricas;
- Reconciliação do limite de recursos financeiros.

5.2 RECURSOS EM PROJETOS DE TI

Para definir o orçamento de um projeto de software é necessário definir quais são os recursos que são utilizados no projeto. Em um projeto de software existem três categorias principais que são utilizados: pessoas, componentes de softwares reusável e o ambiente de desenvolvimento, conforme PRESSMAN (2005, p.522). A figura 04 demonstra os recursos de software:

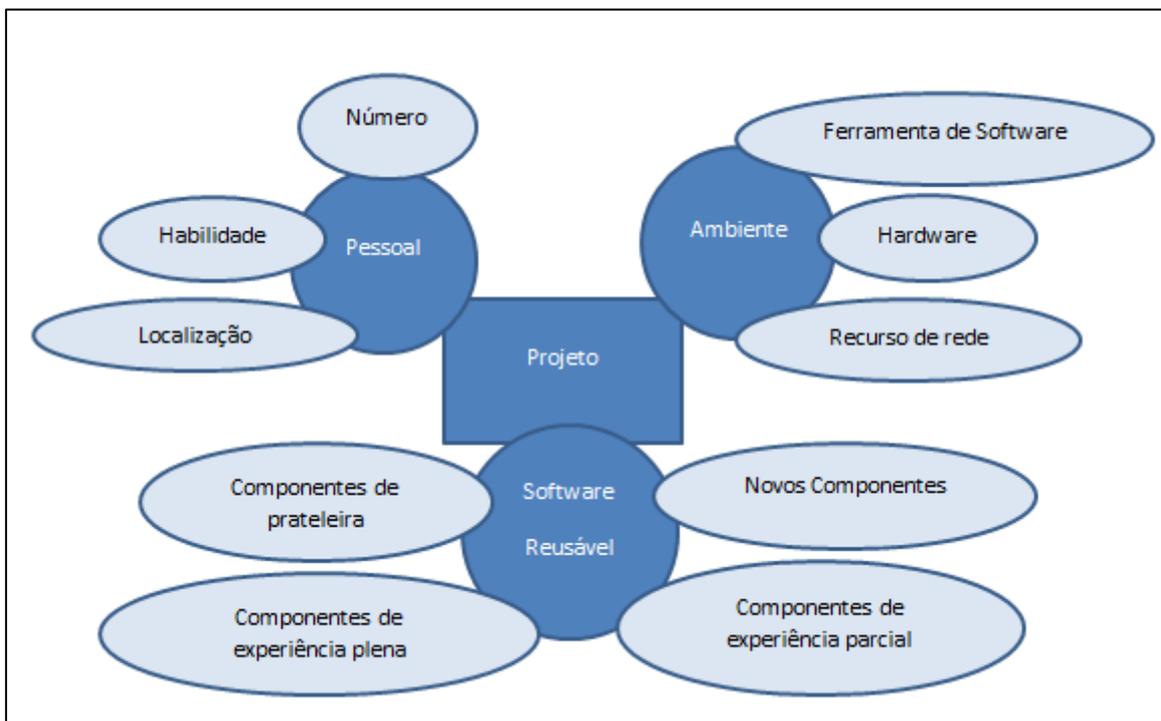


Figura 04: Recursos de Projeto.
Fonte: PRESSMAN (2005, p.522)

Cada recurso é especificado por quatro características: descrição do recurso, época em que o recurso será necessário, declaração de disponibilidade, e prazo em que o recurso estará disponível.

5.2.1 RECURSOS HUMANOS

O gerente de projeto seleciona as pessoas a partir de quais aptdões será necessário para realizar o projeto, tanto como a posição como um engenheiro ou um analista e sua especialidade como base de dados ou telecomunicações. Para projetos pequenos as vezes uma

pessoa é o suficiente para realizar o projeto e acionado um especialista quando necessário. Para projetos grandes as equipes de software podem estar localizadas em locais diferentes.

A quantidade de pessoas que será utilizada no projeto só definido após a estimativa de esforço tiver sido concluída (PRESSMAN, 2005).

5.2.2 RECURSOS SOFTWARE REUSÁVEIS

Engenharia de software em componentes enfatiza a reusabilidade, isto é, a criação e o reuso de blocos construtivos de software. Estes blocos devem ser catalogados para facilitar a referencia, padronizado e validados.

Existem quatro tipos de recursos de softwares, segundo Bennatan, citado por PRESSMAN (2005, p. 523).

- Componentes de prateleira (*commercial of-the-shelf* - COTS): São softwares já existentes que pode ser adquiridos por terceiros ou por projetos que já foram realizados e que estão prontos para serem utilizados e previamente validados.
- Componentes de experiência plena: São especificações, projetos ou códigos já existentes de projetos anteriores e que são parecidos com o projeto atual tendo uma baixa modificação nos componentes. O membro da equipe possui experiência plena para realizar as modificações necessárias e gerando um risco baixo para o projeto.
- Componentes de experiência parcial: São especificações, projetos ou códigos já existentes de projetos anteriores e que são poucos parecidos com o projeto atual tendo uma alta modificação nos componentes. O membro da equipe possui uma experiência limitada para realizar as modificações necessárias e gerando um risco alto para o projeto.
- Componentes novos: São componentes que serão construídos pela equipe especificamente para atender o projeto.

5.2.2 RECURSOS DE AMBIENTE

Segundo PRESSMAN (2005, p. 524), o ambiente de engenharia de software (*software engineering environment* – SEE) é o ambiente que apoia o projeto de software. O Hardware é a plataforma que apoias os softwares para o desenvolvimento do projeto. Como a maioria das empresas possui várias localizações é necessário que o gerente de projeto possibilite que todos

os recursos do projeto tem acesso ao SEE. Como existem várias formas de acessos e restrição é necessário cada elemento de hardware deve ser especificado pelo planejador do projeto.

6 CONTROLANDO OS CUSTOS EM UM PROJETO DE TI

Segundo PMBOK (2008), controlar o custo em um projeto é o processo de monitoramento do progresso do projeto para a atualização do seu orçamento e gerenciamento de mudança feito na linha base dos custos. A atualização do orçamento é feito quando a custo real durante o projeto que não estava estimado ou na linha base. Para um controle eficaz de custo é o gerenciamento da linha de base do desempenho do custo aprovada e a mudança das mesmas.

O controle do projeto inclui:

- Influenciar os fatores que criaram as mudanças na linha de base de custo autorizada;
- Assegurar que todas as solicitações de mudanças sejam feitas de maneira oportuna;
- Gerenciar as mudanças reais conforme ocorrem;
- Assegurar que os gastos de custos não excedam os recursos financeiros autorizados por período e total do projeto;
- Monitorar o desempenho de custos para isolar e entender as variações a partir da linha de base custo do projeto;
- Monitorar o desempenho do trabalho em relação ao custo recursos financeiros gastos;
- Prevenir que as mudanças não aprovadas sejam incluídas no relato dos custos ou do uso do recurso;
- Informar as partes interessadas apropriadas a respeito de mudanças aprovadas e custos associados e agir para manter os excessos de custos não previstos dentro dos limites;

6.1 FERRAMENTAS PARA CONTROLAR CUSTOS SEGUNDO PMBOK

Neste capítulo vamos mostrar algumas ferramentas para controlar o custo de projeto de TI, iremos apresentar algumas técnicas indicadas pelo PMBOK (2008):

6.1.2 PREVISÃO

Conforme citado no PMBOK (2008), é uma previsão para a estimativa no término (ENT) que pode ser diferente do orçamento do término (ONT) baseando no andamento do

projeto. As previsões são geradas, atualizadas e emitidas em base nas informações sobre o desempenho do trabalho conforme o andamento do projeto. É comum para este controle é uma soma manual feita de maneira *bottom-up* pelo gerente e a equipe do projeto. A ENT é baseada nos custos reais incorridos para o trabalho executado, mais para uma estimativa para terminar (EPT) o trabalho restante.

A desvantagem deste tipo de controle é que é necessário para a equipe do projeto para fornecer informações detalhada do andamento do projeto e este tempo normalmente não está orçado para a elaboração do ENT.

A equação do ENT pode ser definida como: $ENT = CR + EPT$ *bottom-up*. Que também pode ser três tipos de cálculos de ENT que são:

- Previsão da ENT para o trabalho EPT para o ritmo orçado. É o desempenho do projeto real até a data, se favorável ou desfavorável, como representado pelos custos reais e prevê que todo o trabalho EPT futuro será executado como orçado. Equação: $ENT = CR + ONT - VA$.
- Previsão da ENT para o trabalho EPT executado ao IDC presente: Assume o que tem acontecido até o momento tem a continuar no futuro. Equação: $ENT = ONT/IDC$ cumulativo.
- Previsão ENT para o trabalho EPT considerando ambos os fatores IDP e IDC: Este método o EPT será executado numa faixa de eficiência que considera os índices de desempenhos prazos e custos. Equação: $CR + [(ONT - VA) / (IDC \text{ cumulativo} \times IDP \text{ cumulativo})]$

6.1.3 INDICE DE DESEMPENHO PARA TÉRMINO

Segundo o PMBOK (2008) é a projeção calculada do desempenho de custos que deve ser atingido no trabalho restante para alcançar um objetivo de gerenciamento especificado, como ONT ou ENT. A equação para o IDPT baseado na ONT é: $(ONT - VA) / (ONT / CR)$.

Caso o IDC cumulativo fique abaixo da linha de base, todo o restante do trabalho deve ser feito no limite do ONT e se for reconhecido que o ONT não é mais possível, será necessário realizar uma na estimativa no término (ENT). A equação para o IDPT baseado na ENT é: $(ONT - VA) / (ENT - CR)$.

6.1.4 ANÁLISE DE DESEMPENHO

É comparar o desempenho de custos através do tempo, atividade do cronograma ou pacotes de trabalho acima e abaixo do orçamento e recursos financeiros estimados necessários para terminar o projeto. Se o GVA for utilizado às informações são determinadas:

- Análise variação: Compara o desempenho do projeto real com o planejado;
- Análise de tendência: Examina o desempenho do projeto através do tempo para determinar se está melhorando ou piorando;
- Desempenho do valor agregado: Compara o plano da linha base com o prazo real de desempenho.

6.1.5 OUTRAS TÉCNICAS

Existem mais duas técnicas que é citado pelo PMBOK que são:

- Análise Variação: São medições do desempenho custos (VC, IDC) usados para avaliar a magnitude de variação à linha base de custo original.
- Software de Gerenciamento de projetos: São usados para monitorar as três dimensões do GVA (VP, VA e CR), para mostrar tendências gráficas e prever uma variedade de resultados finais possíveis do projeto.

6.2 VALOR AGREGADO

O valor agregado é a relação entre os custos reais incorridos e o trabalho realizado no projeto dentro de um determinado período de tempo, e será analisado o desempenho obtido com o que foi gasto para obtê-lo. Em seu livro, VARGAS (2011, p. 16), define valor agregado como a avaliação entre o que foi obtido em relação ao que foi realmente gastos e ao que se planejava gastar, onde se propõe que o valor a ser agregado inicialmente por uma atividade é o valor orçado por ela. Assim que uma atividade é realizada, o esforço desta atividade passa a constituir o Valor Agregado do projeto.

Os três elementos básicos da análise, conforme VARGAS(2011) são:

- BCWS (*Budget cost of work scheduled*): Valor que indica a parcela do orçamento que deveria ser gasta, considerando o custo de linha da base da atividade, atribuição ou recurso. A tradução do BCWS é COTA (Custo Orçado do Trabalho Agendado), é também pode ser PV (Planned Value) ou VP (Valor Planejado).

- **BCWP** (*Budget cost of work performed* ou Valor Agregado): Valor que indica a parcela que deveria ser gasta, considerando-se o trabalho realizado até o momento e o custo de linha de base para a atividade, atribuição ou recurso. É também conhecido como COTR (Custo Orçado do Trabalho Realizado) ou EV (*Earned Value*) ou VA (Valor Agregado).
- **ACWP** (*Actual cost of work performed*): Informa os custos reais decorrentes do trabalho já realizado por um recurso ou atividade, até a data de referencia, ou data atual do projeto, proveniente dos dados financeiros. É conhecido como CRTR (Custo Real do Trabalho Realizado), AC (*Actual Cost*) ou CR (Custo Real).

A partir que são determinados esses parâmetros, a análise dos resultados é obtidos na correlação entre eles gerando algumas variações:

- **CV** (*Cost Variance*): É a diferença entre o custo previsto e o custo real até a data de referência. Se o CV for positiva, o custo do trabalho agregado estará aquém do valor realmente gasto e se for negativa, a atividade está agregando um valor inferior ao que se gastou no trabalho. A formula para está variação é: $CV = BCWP - ACWP$.
- **SV** (*Scheduled Variance*): É a diferença entre o Valor Agregado e a agenda da linha de base, se SV for positiva o projeto está adiantado senão estra atrasado. A fórmula desta variação é: $SV = BCWP - BCWS$.
- **TV** (*Time Variance*): É a diferença, em termo de tempo, entre o previsto pelo projeto e o realizado.

Durante o projeto são realizadas métricas e previsões sobre o custo e o tempo finais do projeto, e para isso são criados índices para analisar esses desempenhos, VARGAS (2011, p.21). Para o Valor Agregado são considerados os seguintes índices:

- **SPI** (*Schedule Performance Index*): É a divisão entre o Valor Agregado e o Valor planejado na linha de base. É conhecido também como IDP (Índice de Desempenho de Prazos). A formula é: $SPI = BCWP/BCWS$.
- **CPI** (*Cost Performance Index*): É a divisão entre o valor Agregado e o custo real, onde mostra o valores reais consumidos pelo projeto e os valores agregado no mesmo período. Este índice é também conhecido como IDC (Índice de Desempenho de Custos). A formula é: $CPI = BCWP/ACWP$.

As informações de índices são usadas na determinação de previsões de estatística para o custo e a duração final do projeto.

6.3 VALOR AGREGADO PARA CONTROLAR PROJETO

Foi feito uma tentativa de avaliar a produtividade de diferentes práticas de gerenciamento de projetos por THAMHAI em 1998, citada por VARGAS (2011, p.

52), e o resultado desta pesquisa mostrou que a Análise do valor agregado tem uma popularidade de 41%, sendo mais popular que as técnicas de caminho crítico, o QFD (*Quality Function Deployment*) entre outros. Esta popularidade não demonstra toda a sua aplicabilidade e valor e existem algumas justificativas para tal informação como falta de compreensão do funcionamento da técnica, ansiedade quanto ao uso adequado da ferramenta, desconforto com pouca familiaridade da técnica, custo de sua implementação entre outros, conforme informado por VARGAS (2011, p.53).

A análise do valor agregado apresenta um conjunto de recursos intrínsecos poderosos, abrangente e variado, tais como projeção de pagamento e forecasting, porém possui uma dificuldade na coleta da informação e baixa agilidade na geração das informações e sua implantação custa em torno de 1% a 1,5% do custo do projeto.

Alguns dos benefícios básicos do valor agregado são, segundo VARGAS (2011, p. 56):

- Sistema de controle gerencial unificado e confiável;
- Proporciona integração do trabalho, prazo e custo utilizando a EAP;
- permitem geração de conhecimento através de análise com projeto anteriores;
- Índices como o índice de desempenho de custos alerta sobre gastos acima do planejado ou o índice de prazo que alerta sobre atrasos;
- A combinação dos índices possibilita previsões e projeções do custo final do projeto.

Conclui-se que para aplicar a análise do valor agregado deve possuir as características de relevância e confiabilidade, VARGAS (2011, p. 56).

7 CONCLUSÃO

O motivo para este estudo é mostrar a importância de uma estimativa bem feita para um projeto de software e para que os todos *stakeholders* envolvidos não tenham surpresas durante e no fim do projeto com o seu orçamento. Pesquisas mostram que muitos projetos de softwares estouram seus orçamentos sendo cancelados ou gerando prejuízo para a empresa.

Mostramos que existem diversas técnicas e formas de estimativas que podem ser usados para estimar o tamanho de um projeto de software, cada um com suas características, vantagens de desvantagens. Todas estas técnicas de estimativas seguiram os modelos que o PMBOK (2004) indica e os estudos da engenharia de software também propõe para esta área de projetos de software. Cada projeto de softwares possui suas peculiaridades, pois os sistemas de softwares possuem uma imensa variedade de linguagem de programação, hardware e tecnologias que se alteram a todo instante e que existem várias técnicas para cada tipo, como estimar software para linguagem a orientado ao objeto, para projetos de plataforma alta, como as que utilizam mainframe, ou projetos para WEB. Todas essas técnicas de estimativas podem ser usadas de forma separadas ou serem relacionadas entre si, refinando melhor a estimativa.

Foram apresentados também duas das técnicas de estimativas que são mais utilizadas na área de projetos de software que são o COCOMO II e Pontos Por função, as duas técnicas são muito utilizado no mercado e que podem ter um estudo mais aprofundado em outras pesquisas. Esta duas técnicas possuem um foco diferente, mas a duas podem ser utilizado de forma combinada. Desta forma é possível estimar o tamanho do projeto de software e saber quantas pessoas serão necessárias para cada etapa do projeto e por quanto tempo, sendo possível controlar melhor os recursos no projeto e maximizando os gastos dentro do projeto.

A definição de orçamento também é uma fase muito importante, que será toda consolidação das informações das estimativas do projeto, junto com os gastos de recursos de hardware, software e humanos e que será aprovado pelos responsáveis do projeto de software.

Por fim, mostramos a importância e as técnicas de controle dos gastos que são indicados pela o guia PMBOK, e com certa atenção para a técnica da Análise do Valor Agregado. Esta técnicas vem sendo com umas das técnicas mais importantes para controle do projeto, tanto para o custo como para o tempo que estão sendo realizadas as tarefas do projeto. A análise do valor agregado possuem muitas informações sendo possível a continuação do seu estudo de uma forma mais detalhada.

Todas estas técnicas e controle são muito importantes dentro do projeto, que devem ser realizados no início do projeto e as informações devem ser atualizadas durante a sua execução. Estas ações serão decisivas para que o projeto de software possa ser finalizado no seu tempo e custo estimado.

REFERÊNCIAS

AGUIAR, Muricio, 2010 – *ESTIMANDO OS PROJETOS COM COCOMO II*

Disponível em: <

http://www.univasf.edu.br/~ricardo.aramos/disciplinas/ES_I_2010_2/COCOMO_II.pdf>

Acessado em 08-10-2012 às 22:00 hs

BORGES, Bruno Góis, *Extração de métricas em software orientado a objetos*. Disponível

em: <[http://www.devmedia.com.br/extracao-de-metricas-em-software-orientado-a-objetos-](http://www.devmedia.com.br/extracao-de-metricas-em-software-orientado-a-objetos-engenharia-de-software-26/17409)

[engenharia-de-software-26/17409](http://www.devmedia.com.br/extracao-de-metricas-em-software-orientado-a-objetos-engenharia-de-software-26/17409)> acessado em 12-set-12.

CALAZANS, Angélica Toffano Seidel, 2004 – *Uso das Ferramentas APF e COCOMO para Estimativa da Capacidade Produtiva da TI*. Disponível em:

<bibliotecadigital.sbc.org.br/download.php?paper=253> . Acessado em: 25-09-2012

CARVALHO, Davi, 2008 – *68% dos Projetos de TI Falham* – Disponível em:

<<http://soasimples.com/blog/?p=378>> Acessado em: 15-set-2012.

COMPUTEWORLD, *Empresas brasileiras investem 3% de seu faturamento em tecnologia,*

Disponível em: <[http://www.digitehost.com.br/tecnologias/32-empresas-brasileiras-investem-](http://www.digitehost.com.br/tecnologias/32-empresas-brasileiras-investem-3-de-seu-faturamento-em-tecnologia)

[3-de-seu-faturamento-em-tecnologia](http://www.digitehost.com.br/tecnologias/32-empresas-brasileiras-investem-3-de-seu-faturamento-em-tecnologia)>. Acesso em: 17 set. 2011.

COSTA, Rodrigo, 2011 – *Gerenciamento em projetos de TI*. Disponível em:

<<http://pt.scribd.com/doc/48654407/91/Criando-um-orcamento-de-projeto#page=105>>

Acessado em 13-set-12.

FILHO, Hamilton Chagas Perez Filho. *Método prático de controle de custo numa empresa de projetos*. Disponível em:

<<http://www.centralmat.com.br/Artigos/Mais/controlCustosEmpresaProjetos.pdf>>. Acesso

em: 30 out. 2011.

FÉ, Ana Lúcia Moura, Informationweek, *Bancos e seguradoras são as que mais investem,*

Disponível em: <[http://informationweek.itweb.com.br/1218/100-bancos-e-seguradoras-sao-](http://informationweek.itweb.com.br/1218/100-bancos-e-seguradoras-sao-os-que-mais-investem-2/)

[os-que-mais-investem-2/](http://informationweek.itweb.com.br/1218/100-bancos-e-seguradoras-sao-os-que-mais-investem-2/)>. Acesso em: 17 set. 2011.

HAZRATI, Vikas – *Projetos de TI: 400% acima do orçamento e 25% dos benefícios esperados*. Disponível em: <<http://www.infoq.com/br/news/2011/11/software-alto-risco>>. Acessado em 13-set-12.

HEIMBERG, Viviane; GRAHL, Everaldo Artur, 2005, *Estudo de Caso de Aplicação da Métrica de Pontos de Casos de Uso numa Empresa de Software*. Disponível em: <<http://www.inf.furb.br/seminco/2005/artigos/130-vf.pdf>> Acessado em 12-set-12>.

HELDMAN, Kim – *Gerencia de Projetos - Fundamentos: Um guia prático para quem quer certificação*, 2005 – 5º Edição.

JENNY, Juliana, 2011 – *Estimativa: Ponto de Função*. Disponível em: <<http://julianakolb.com/2011/05/09/estimativa-ponto-por-funcao/>>: Acessado em: 08-10-2012

JUNIOR, Clay Susini Aquino, 2010 – *Estimando custo em projetos*. Disponível em: <<http://ogerente.com.br/rede/projetos/estimativa-custo-projeto>> Acessado em 12-set-12.

LOPEZ, Pablo Ariel do Prado – *COCOMO II - Um modelo para estimativa de custos de Gerência de Projetos*, 2005 – Disponível em: <<http://www.sirc.unifra.br/artigos2005/artigo18.pdf>> Acessado em: 16-set-12.

MAIA, Jose Ricardo Correa. *Use Métricas adequadas – Garanta qualidade de projeto orientado a objeto*. Disponível em: <http://www.euax.com.br/system/attachments/4/original/2006.013-Metricas_software.pdf?1265047553>. Acesso em: 21 ago. 2011.

MACHADO, Breno, 2008, *Você sabe como estimar o valor de seu projeto*, disponível em: <<http://www.pmo4u.com.br/2008/09/voc-sabe-como-estimar-o-valor-de-seu.html>>. Acessado em 12-set-12.

MEDEIROS, Ermani. *Desenvolvendo Software com UML 2.0*. São Paulo: ed. Makron Books, 2004.

MELO, Maury, *Guia de Estudo para o Exame PMP*, 4 Ed. 2012.

MORICONI, Rubens, *Gestão de Projetos – Gestão de Custo*, 2011.

PARO, Caio – 2005, *Medidas de tamanho de desenvolvimento e de melhorias de software*
Disponível em: <<http://www.bfpug.com.br/>>. Acessado em: 16-set-12

PHILLIPS Joseph, *Gerência de Projetos de Tecnologia da Informação – 2002*.

PRESSMAN, R.S. *Engenharia de Software*, 1995, 5 Edição.

PRESSMAN, R.S. *Engenharia de Software*, 2002, 6 Edição.

PMI, *PMBOK 4º Edição – Project Management Institute , Inc.*, 2008.

REINALDO, Werley Teixeira, Cristina D’Ornellas Filipakis, 2011 - *Estimativa de Tamanho de Software Utilizando APF e a Abordagem NESMA*

Disponível em: <<http://pt.scribd.com/doc/65248300/Estimativa-de-Tamanho-de-Software-Utilizando-APF-e-a-Abordagem-NESMA>> : Acessado em: 08-10-2012

ROVAI, Ricardo Leonardo; SILVA, Marcello Muniz; CAMPANÁRIO, Milton de Abreu, 2004 - *Metodologias de Estimativa de Prazos, Custos e Orçamentos em Projetos de T.I.*

Disponível em: <<http://www.tecsi.fea.usp.br/contecsi/arquivos/docs/1CONTECSI/pdfs/104-028.pdf>>. Acessado em: 16-set-12.

SOUZA, Washington, *Chaos Report: Como esta a TI no mundo?* Disponível em:

<<http://www.blogcmmi.com.br/geral/chaos-report-como-esta-a-ti-no-mundo>> Acessado em 17-09-2011 às 12:00 hs

VARGAS, Ricardo – *Análise de valor agregado – Revolucionando o gerenciamento de prazos e custos – 5º Ed.* 2011.