



**Guia de Estimativa de Projeto
Metodologia Celepar**

Agosto 2009

Sumário de Informações do Documento			
Documento: guiaEstimativaProjeto.odt		Número de páginas: 30	
Versão	Data	Mudanças	Autor
1.0	02/01/2006	Elaboração	<i>Danielle Mayer</i>
1.1	13/04/2006	Atualização	<i>Danielle Mayer</i>
1.2	06/09/2006	Revisão	<i>Danielle Mayer</i>
1.3	16/05/07	Revisão e Atualização	Ariel Bolzan Witczak Danielle Mayer GIC – Ariel Bolzan Witczak GIC – Cleverson Budel GIC – Danielle Terezinha de Castro Mayer GTI - Luciana Maria Reis da Costa GTI - Karina Paula de Camargo Curcio
1.3	31/05/07	Revisão.	
1.3	12/08/2009	Revisão	<i>Marcos Chiarello</i>
1.3	26/11/09	Revisão – Trocando a tabela de Fatores Técnicos e Fatores Ambientais por imagem	<i>Danielle Mayer</i>

SUMÁRIO

1INTRODUÇÃO.....	4
2UCP.....	4
2.1.1UCP Não Ajustado.....	5
2.2FATORES TÉCNICOS DE COMPLEXIDADE.....	6
2.3FATORES AMBIENTAIS.....	19
2.4UCP AJUSTADO.....	23
2.5ESTIMANDO O ESFORÇO PARA A REALIZAÇÃO DO PROJETO.....	23
3ESTUDOS DE CASO (UCP).....	24
3.1ESTUDOS DE CASO APRESENTADOS NO ARTIGO DE GUSTAV KARNER.....	24
3.2ESTUDOS DE CASO APRESENTADOS NO ARTIGO DE VIVIANE HEIMBERG E EVERALDO ARTUR GRAHL.....	26
4EXTRAINDO MAIS DADOS A PARTIR DA ESTIMATIVA.....	29
5CONCLUSÃO.....	30
6REFERÊNCIAS.....	30

1 INTRODUÇÃO

A estimativa de tempo para realizar um determinado sistema depende da complexidade e tamanho do sistema a ser desenvolvido. Esta estimativa pode ser realizada a partir da aplicação de métricas, como por exemplo, UCP¹.

É possível determinar a estimativa do desenvolvimento de um sistemas através de base histórica de outros desenvolvimentos. A base histórica pode auxiliar também na determinação do índice de conversão de UCP para horas.

Este documento visa guiar a elaboração da estimativa dos projetos de desenvolvimento de software. A estimativa da complexidade é calculada baseando-se nos seguintes elementos: Atores, Casos de Uso, Fatores Técnicos e Ambientais.

O próximo tópico deste documento visa apresentar como é realizado o cálculo através do método UCP.

2 UCP

Método criado por Gustav Karner, em 1993, para medir projetos de software orientado a objetos. Este método é utilizado para mensurar projetos baseado no modelo de caso de uso, permite realizar a estimativa logo no início dos projetos durante o levantamento de requisitos. Além da avaliação do modelo de casos de uso são considerados fatores de complexidade técnica (TCF), cuja finalidade é determinar o grau de complexidade do projeto a ser construído, e fatores ambientais (EF), cuja finalidade é determinar a eficiência do projeto e o nível de experiência dos profissionais relacionados.

Os passos para realizar esta medição são descritos a seguir:

¹ Use Case Points ou Pontos por Caso de Uso, deste momento em diante denominado apenas UCP

2.1.1 UCP Não Ajustado

- Relacionar e classificar os atores envolvidos, de acordo com o nível de complexidade representado abaixo:

Complexidade	Definição	Peso
Simples	Quando o ator representa um sistema externo que é acessado através de API(Application Programming Interface)	1
Médio	Quando o ator representa um sistema externo, acessado através de um protocolo de comunicação(por exemplo: TCP/IP)	2
Complexo	Quando o ator (humano) interage com o sistema através de uma interface gráfica(GUI)	3

- Relacionar e classificar os casos de usos, de acordo com o nível de complexidade representado abaixo:

Complexidade	Definição	Peso
Simples	O caso de uso possui até 3 caminhos internos ou é realizado utilizando menos de 5 objetos (classes de análise)	5
Médio	O caso de uso possui de 4 a 7 caminhos internos ou é realizado utilizando de 5 a 10 objetos (classes de análise)	10
Complexo	O caso de uso possui mais de 7 caminhos ou é realizado utilizando mais de 10 objetos (classes de análise)	15

Caminho interno = fluxo principal + fluxos alternativos do caso uso.

Critério de Classificação: Para evitar subdimensionamento da estimativa, todo caso de uso que possuir alguma característica que o enquadre em um nível de complexidade maior deve ser

classificado neste nível.

- Cálculo do UUCP²:

$$\text{UUCP} = \text{Total de pesos dos atores} + \text{Total de pesos dos casos de uso}$$

2.2 Fatores Técnicos de Complexidade

- Atribuir um valor a cada fator numa escala de 0 a 5, onde 0 significa que o fator é irrelevante e 5 significa que é essencial. Se o fator não é importante e não é irrelevante deve-se atribuir valor 3.
- Cada fator receberá um valor no projeto:
 - 0 - Nenhuma influência;
 - 1 - Influência mínima;
 - 2 - Influência moderada;
 - 3 - Influência média;
 - 4 - Influência significativa;
 - 5 - Grande influência.

2 Unadjusted Use Case Point ou Pontos por Caso de Uso Não Ajustados, deste momento em diante denominado apenas UUCP

Código Fator	Descrição do Fator	Peso	Valor (Fator)	Resultado
F1	Sistema Distribuído	2		0
F2	Performance (Tempo de Resposta)	1		3
F3	Usabilidade	1		3
F4	Complexidade de Processamento	1		0
F5	Exigência de Reusabilidade	1		0
F6	Facilidade de Instalação	0,5		0
F7	Facilidade de operação da Aplicação	0,5		1,5
F8	Portabilidade	2		0
F9	Manutenibilidade	1		0
F10	Concorrência(acessos simultâneos)	1		3
F11	Exigências de Segurança	1		0
F12	Interdependência com software de terceiros (sistemas / componentes externos)	1		0
F13	Exigência de Treinamento(Usuários Finais)	1		3

Figura 1: Fatores Técnicos de Complexidade

- Multiplicar o Valor(Fator) atribuído pelo respectivo peso (coluna resultado)
- Totalizar o resultado da multiplicação (TFator)
- Calcular o fator de complexidade de acordo com a seguinte formula:

$$TCF^3 = 0.6 + (0.01 * Tfator)$$

Segue abaixo uma lista de características para **alguns** dos fatores de complexidade técnica, com o objetivo de auxiliar a definição de seu respectivo valor:

F1 – Sistema Distribuído: Descreve o nível em que a aplicação transfere dados entre seus componentes. Funções ou dados distribuídos dentro da fronteira são características da aplicação.

³ Technical Complexity Factor ou Fator Técnico de Complexidade, deste momento em diante denominado apenas TCF.

0 () A aplicação não participa da transferência de dados ou processamento 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 instalados no próprio computador.

2 () Os dados são preparados para transferência, transferidos e são processados em uma outra CPU da instalação (mas NÃO para processamento pelo usuário final como visto no item 1).

3 () Processamento distribuído e transferência de dados são feitos on-line apenas e em apenas uma direção.

4. () Processamento distribuído e transferência de dados são feitos on-line em ambas direções.

5 () O processamento de funções é executado dinamicamente no componente mais apropriado do sistema.

Considerações sobre sistemas distribuídos: Esta é uma característica mais presente em sistemas transacionais do que em sistemas de informação gerencial e suporte à decisão. Um sistema de desktop isolado (aplicação e banco de dados local) pontuará com 0. Um sistema de n camadas pontuará com 4. Para pontuar com 5, o sistema deveria ter componentes executando em múltiplos servidores ou processadores, sendo cada um deles selecionado dinamicamente de acordo com sua disponibilidade.

F2 – Performance (Tempo de Resposta): Descreve o nível em que considerações sobre o tempo de resposta e taxa de transações influenciam no desenvolvimento da aplicação. Os objetivos estabelecidos ou aprovados pelo usuário, em termos de tempo de resposta ou taxa de transações, influenciam (ou influenciará) no projeto, desenvolvimento, instalação e suporte da aplicação. A questão que deve ser avaliada é “Quão rápida deve ser a aplicação e o quanto isto influencia o projeto”.

0 () O usuário não estabeleceu nenhum requisito especial sobre performance.

1 () Requisitos de performance e projeto foram estabelecidos e revisados, mas nenhuma ação em

especial foi tomada.

2 () O tempo de resposta ou taxa de transações são críticos durante as horas de pico. O intervalo de tempo limite (deadline) do processamento é sempre para o próximo dia útil. Não é necessário nenhum projeto especial para utilização de CPU.

3 () O tempo de resposta ou a taxa de transações são críticos durante todo o horário de utilização. O limite de processamento é crítico. Não foi necessário nenhum procedimento especial para utilização de CPU.

4. () Os requisitos de performance estabelecidos pelo usuário são rigorosos o bastante para requerer tarefas de análise de performance na fase de análise e projeto da aplicação.

5 () 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.

Considerações sobre performance: Características típicas e com forte relação ao volume de transações. Um sistema bancário onde uma regra de negócio onde o tempo para a autenticação de um documento pelo caixa não poderia ser superior ao atual sistema, este é um exemplo de pontuação 5.

F3 - Usabilidade: Descreve em que nível considerações sobre fatores humanos e facilidade de uso pelo usuário final influenciam no desenvolvimento da aplicação. As funções interativas fornecidas pela aplicação enfatizam um projeto para o aumento da eficiência do usuário final:

Exemplos de funções:

- Auxílio para a navegação, como, por exemplo, teclas de função, saltos, menus gerados dinamicamente;
- Menus;
- Documentação/Help online;

- Movimento automático do cursor;
- Movimento de Tela (scrolling) vertical e horizontal;
- Impressão remota (via transações online);
- Teclas de Função pré-definidas;
- Execução de jobs batch a partir de transações online;
- Seleção de dados da tela via movimentação do cursor;
- Uso intenso de vídeo reverso, brilho intensificado, sublinhado, cores e outros recursos de vídeo;
- Documentação de transações online via hardcopy;
- Interface para mouse
- Janelas do tipo PopUp;
- O mínimo possível de telas para executar as funções do negócio;
- Fácil navegação entre telas (por exemplo, através de teclas de função);
- Suporte a mais de um bilíngüe (suporta dois idiomas, contar como quatro itens);
- Suporte multilíngüe (suporta mais de dois idiomas, contar como seis itens).

0 () A aplicação não apresenta nenhum dos itens acima.

1 () Apresenta de 1 a 3 dos itens acima.

2 () Apresenta de 4 a 5 dos itens acima.

3 () Apresenta 6 ou mais dos itens acima, mas não há nenhum requisito do usuário relacionado à eficiência.

4. () Apresenta 6 ou mais dos itens acima, e os requisitos estabelecidos para eficiência do usuário são rigorosos o suficiente para que a fase de projeto da aplicação inclua fatores para: minimizar a digitação, maximizar os defaults, , utilizar templates etc.

5. () Apresenta 6 ou mais dos itens acima, e os requisitos estabelecidos para eficiência do usuário são rigorosos o suficiente para que seja necessário o uso de ferramentas e processos especiais para demonstrar que os objetivos de eficiência foram alcançados.

Considerações sobre usabilidade: Algumas destas diretrizes também estão defasadas, uma vez que a interface gráfica dos sistemas operacionais atuais já provê automaticamente várias destas características (antigamente era a própria aplicação que tinha que implementá-las). Aplicações servidoras não possuem interação com usuário final e pontuarão 0. Aplicações tipicamente padrão (windows) pontuarão de 3 a 5.

F4 – Complexidade de Processamento - Descreve em que nível o processamento lógico ou matemático influencia o desenvolvimento da aplicação. A complexidade de processamento influencia no dimensionamento do sistema, e, portanto, deve ser quantificado o seu grau de influência com, base nas seguintes categorias:

- Controle sensível e/ou processamento específico de segurança da aplicação. Ex: Processamento especial de auditoria;
- Processamento lógico extensivo. Ex: sistemas de gestão de créditos;
- Processamento matemático extensivo. Ex: sistemas de mapeamento genético;
- Grande quantidade de processamento de exceções, resultante de transações incompletas que necessitam de reprocessamento. Ex: transações incompletas de terminais de atendimento bancário causadas por interrupções de comunicação, valores de dados ausentes ou validações de erros;
- Processamento complexo para manipular múltiplas possibilidades de entrada/saída. Ex: múltiplos meios e independência de equipamentos, multimídia, telefonia celular.

0 () Não apresenta nenhum dos itens acima;

1 () Apresenta qualquer um dos itens acima;

2 () Apresenta dois dos itens acima;

3 () Apresenta três dos itens acima;

4 () Apresenta quatro dos itens acima;

5 () Apresenta todos os itens acima.

Considerações sobre processamento interno complexo: Como nas outras características gerais de sistemas, deve-se avaliar não apenas uma funcionalidade específica, mas o geral da aplicação. O fato de haver uma funcionalidade com grande processamento matemático extensivo, como uma apropriação financeira de encargos ou o levantamento de um saldo devedor, deve ser considerado no contexto do sistema como um todo. Nesse caso, onde uma parte considerável do processamento envolve esse tipo de lógica, deve-se considerar esse componente como presente. Agora, caso seja uma aplicação em que esses processamentos sejam periféricos e constituam uma pequena parte do conjunto total, deve-se ponderar para menos o impacto na aplicação como um todo.

F5 – Exigência de Reusabilidade - Descreve em que nível a aplicação e o seu código foram especificamente projetados, desenvolvidos e suportados para serem utilizados em outras aplicações.

0 () Não apresenta código reutilizável;

1 () O código reutilizável é usado somente dentro da aplicação.

2 () Menos de 10% da aplicação foi feita, levando-se em conta a sua utilização por outras aplicações

3 () 10% ou mais da aplicação foi feita, levando-se em conta a sua utilização por outras aplicações.

4. () A aplicação foi especialmente empacotada e/ou documentada para uma fácil reutilização. Ela é

customizada pelo usuário no nível do código fonte;

5. () A aplicação foi especialmente empacotada e/ou documentada para uma fácil reutilização. Ela é customizada pelo usuário por meio de manutenção de parâmetros.

Considerações sobre reusabilidade: Trata-se de um requisito de quantidade e não de um requisito funcional. É difícil (senão impossível) de ser avaliada para uma aplicação em que não estão disponíveis documentos do projeto ou o código-fonte. Além disso, considerações técnicas que contrariam o objetivo principal da técnica que é medir o software do ponto de vista do usuário pela funcionalidade fornecida.

F6 – Facilidade de Instalação - Descreve em que nível a conversão de ambientes preexistentes influencia o desenvolvimento da aplicação. Um plano e/ou ferramentas de conversão e instalação foram fornecidos e testados durante a fase de teste do sistema. O cliente procura determinado grau de facilidade no processo de instalação do sistema. O comum é o próprio desenvolvedor determinar como será construído o instalador (utilizando-se de ferramentas que geram o instalador), mas se o cliente está procurando por uma instalação feita sob medida provavelmente dependerá de um módulo mais complexo. Exemplo: um cliente exige que possa escolher os módulos que irá instalar ou então exige que quando há uma nova versão ocorra uma atualização automática. Estes fatores devem ser levados em conta ao atribuir valor a este fator.

0 () Nenhuma consideração especial foi feita pelo cliente e nenhum procedimento especial é requerido instalação.

1 () O usuário não definiu considerações especiais, mas é necessário um “*setup*” para a instalação;

2 () Requisitos de instalação e conversão foram fixados pelo usuário, e guias de conversão e instalação foram fornecidas e testadas. Não é considerado importante o impacto da conversão;

3 () Requisitos de instalação determinados pelo usuário e roteiros de instalação foram preparados e testados. É considerado importante o impacto da conversão;

4. () Além do descrito no item 2, ferramentas automatizadas de instalação e conversão foram preparadas e testadas.

5. () Além do descrito no item 3, ferramentas automatizadas de instalação e conversão foram preparadas e testadas.

Considerações sobre a facilidade de instalação: Esta é uma característica importante em projetos de sistemas que irão substituir aplicações existentes, pontuando de 3 a 5 nesse caso.

F7 – Facilidade de operação da Aplicação – Descreve em que nível a aplicação atende a alguns aspectos operacionais. Procedimentos efetivos de inicialização, segurança, backup e recuperação foram desenvolvidos e testados. A aplicação minimiza a necessidade de atividades manuais, tais como montagem de fitas, manuseio de formulários e intervenção manual do operador.

0 () Nenhuma consideração especial sobre facilidade operacional, além dos procedimentos normais de backup e segurança, foi feita pelo usuário.

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

- 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;

5. () A aplicação foi projetada para não precisar de intervenção do operador no seu funcionamento normal. Apenas a inicialização e parada do sistema ficam a cargo do operador. A recuperação automática de erros é uma característica da aplicação.

Considerações sobre a facilidade de operação: Este também é um caso de defasagem nas diretrizes do IFPUG. Para aplicações em que não há necessidade da figura do operador, mas apenas

a figura do usuário, a pontuação será 5.

F8 – Portabilidade: Descreve em que nível a aplicação foi especificamente projetada, desenvolvida e suportada para ser instalada em múltiplas plataformas de hardware e software (Windows, Unix, Linux, etc).

0 () Nenhuma solicitação do usuário para considerar a necessidade de instalar a aplicação em mais de uma plataforma;

1 () Necessidade de instalação em múltiplas plataformas foi levada em consideração no projeto do sistema e a aplicação foi projetada para operar somente em ambientes idênticos de hardware e software;

2 () Necessidade de instalação em múltiplas plataformas foi levada em consideração no projeto do sistema e a aplicação foi projetada para operar somente em ambientes similares de hardware e software;

3 () Necessidade de instalação em múltiplas plataformas foi levada em consideração no projeto do sistema e a aplicação foi projetada para operar inclusive em plataformas diferentes.

4. () Um plano de documentação e manutenção foi elaborado e testado para suportar a aplicação em múltiplas plataformas e a aplicação atende aos itens 1 e 2.

5. () Um plano de documentação e manutenção foi elaborado e testado para suportar a aplicação em múltiplas plataformas e a aplicação atende ao item 3.

F9 – Manutenibilidade: 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:

- 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).

0 () Nenhum dos itens anteriores;

1 () Qualquer um dos itens anteriores;

2 () Quaisquer dos itens anteriores;

3 () Quaisquer três itens anteriores;

4 () Quaisquer quatro itens anteriores;

5 () Todos os cinco itens anteriores.

Considerações sobre a facilidade de mudanças: Existem dois tipos de componentes que estão avaliados nesta característica geral de sistema: mecanismos de consultas flexível e manutenção de dados de controle do sistema. O primeiro reflete consultas em que o próprio usuário monta relatórios a partir dos dados disponíveis no sistema. O segundo é referente à manutenção de parâmetros de forma on-line por meio de manutenções de tabelas, por exemplo.

F10 – Concorrência(acessos simultâneos): Descreve em que nível a aplicação foi especificamente desenvolvida para proporcionar acessos simultâneo.

- 0 () Não é esperado acesso simultâneo;
- 1 () São esperados acessos simultâneos esporadicamente;
- 2 () Acessos simultâneos são esperados;
- 3 () Acessos simultâneos são esperados diariamente;
- 4 () Muitos acessos simultâneos foram fixados pelo usuário para a aplicação, o que força a execução de tarefas de análise de performance na fase de projeto da aplicação;
- 5 () Requer o uso de ferramentas controle de acesso nas fases de projeto desenvolvimento e/ou implantação, além das considerações acima.

Considerações sobre concorrência: O usuário está prevendo um grande número de acessos simultâneos na aplicação, com isto irá aumentar a complexidade e por conseguinte, este valor.

F11 – Exigências de Segurança: Descreve em que nível a aplicação foi especificamente desenvolvida para proporcionar níveis de segurança.

- 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 aplicação

e a auditoria.

Considerações sobre segurança: O usuário pretende utilizar um sistema de criptografia pesado como SSL? Ou será que temos de escrever códigos personalizados e lógica para criptografia?

F12 - Interdependência com software de terceiros (sistemas/componentes externos): Indica o grau de interdependência do projeto em relação a uso de controles externos (depende de estímulos, ou estimular, componentes/sistemas externos), assim será necessário um determinado esforço no sentido de compreender esses controles e avaliar seus prós e contras em relação a sua utilização.

0 () Não há interdependências com componentes externos;

1 () Há pouco acesso a/ou de sistemas externos;

2 () Esporadicamente ocorrem acessos a/ou de componentes externos;

3 () Espera-se um médio número de acessos a/ou de sistemas externos;

4 () Frequentemente ocorrem acessos a/ou de componentes externos;

5 () Forte dependência com sistemas externos.

Considerações em relação a Interdependência com software de terceiros: Será que o projeto depende de componentes de terceiros? Precisamos dar uma atenção especial às possíveis interfaces com outros programas? Portanto, caso afirmativo, este fator deve ser avaliado em conformidade.

F13 - Exigência de Treinamento(Usuário Finais): Descreve a complexidade de utilização da aplicação, sobre a perspectiva do usuário, que necessita de treinamento específico.

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.

2.3 Fatores Ambientais

- Atribuir um valor a cada fator numa escala de 0 a 5, onde 0 significa que o fator é irrelevante e 5 significa que é essencial. Se o fator não é importante e não é irrelevante deve-se atribuir valor 3.

Código Fator	Descrição do Fator	Peso	Valor (Fator)	Resultado
Fatores que beneficiam o projeto(quanto maior, melhor)				
F1	Familiaridade com o Processo de Desenvolvimento(metodologia)	1,5		7,5
F3	Presença de Analista Experiente	0,5		2,5
F4	Experiência da Equipe em Desenvolvimento de Aplicações deste Gênero	0,5		2,5
F5	Experiência da Equipe em Orientação a Objetos	1		5
F6	Motivação	1		5
F8	Estabilidade dos Requisitos	2		10
Fatores que prejudicam o projeto(quanto maior, pior)				
F2	Colaboradores de meio período	-1		0
F7	Dificuldade da equipe relativo a linguagem de programação	-1		0

- Multiplicar o Valor(Fator) atribuído pelo respectivo peso (coluna resultado)
- Totalizar o resultado da multiplicação (EFator)
- Calcular o fator de complexidade de acordo com a seguinte formula:

$$EF^4 = 1.4 + (-0.03 * EFator)$$

- Segue abaixo uma lista de características para os fatores ambientais que visam auxiliar a definição de seu respectivo valor:

F1 - Familiaridade com o Processo de Desenvolvimento (metodologia): Indica a experiência da equipe com o processo/método utilizado para desenvolvimento do projeto.

0 () A equipe não tem experiência 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.

F2 – Colaboradores de meio período: Mede a estabilidade da equipe e a influencia do trabalho parcial na produtividade.

() 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 () Toda os membros da equipe trabalham em período parcial

F3 – Presença de Analista experiente: Indica a experiência do analista líder com análise de sistemas.

0 () O líder do projeto é novato.

4 Environmental Factor ou Fator Ambiental, deste momento em diante denominado apenas EF.

1 - 2 () Possui experiência de poucos projetos

3 - 4 () Pelo menos 2 anos de experiência com vários projetos

5. () Pelo menos 3 anos de experiência com projetos variados

F4 – Experiência da Equipe em Desenvolvimento de Aplicações deste Gênero: Indica 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

F5 – Experiência da Equipe em Orientação a Objeto: Mede 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)

F6 – Motivação: Descreve a motivação total da equipe.

0 () Não motivada

1 - 2 () Pouca motivada

3 - 4 () A equipe está motivada para fazer um bom trabalho

5. () A equipe está muito motivada e inspirada

F7 – Dificuldade da equipe relativo à linguagem de programação: Indica 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.

F8 – Estabilidade dos Requisitos: Mede o grau de mudança de requisitos e inseguranças sobre o significado dos requerimentos.

0 () Requisitos muito instáveis com mudanças freqüentes

1 - 2 () Requisitos instáveis. Clientes demandam algumas mudanças realizadas em diversos intervalos

3 - 4 () Estabilidade global. Pequenas mudanças são necessárias

5. () Requisitos estáveis ao longo do desenvolvimento

2.4 UCP Ajustado

- É calculado utilizando os dados obtidos anteriormente:

$$\text{UCP} = \text{UUCP} * \text{TCF} * \text{EF}$$

2.5 Estimando o esforço para a realização do projeto

O esforço é estimado em horas, dado por: **Horas** = UCP(Ajustados) * Produtividade

A variável “Produtividade” é determinada segundo o grau de produtividade do grupo. Quando esta diminui leva-se menos tempo na conclusão das tarefas, indicando que houve um aumento na experiência da equipe de projeto. Usa-se a seguinte tabela na atribuição do fator de produtividade:

Produtividade	Descrição
20 homens*hora/UCP	Equipe com produtividade alta
28 homens*hora/UCP	Equipe com produtividade média a baixa ou desconhecida
Maior que 28 homens*hora/UCP	Reavaliar Projeto

Tabela 1: Atribuição do fator de produtividade segundo a experiência

Geri Schneider, em seu livro Applying Use Cases (Addison-Wesley, 1998) sugere cuidados com o Fator Ambiental (EF) quando:

Contar a quantidade de fatores, entre F1 e F6, avaliados abaixo de 3
+
Contar a quantidade de fatores, F7 e F8, avaliados acima de 3

Total de Fatores

Total de Fatores = 2 ou menos	Produtividade = 20 homens*hora/UCP
Total de Fatores = 3 ou 4	Produtividade = 28 homens*hora/UCP
Total de Fatores = 5 ou mais ¹	Reavalie o projeto há alto risco de falha.

3 ESTUDOS DE CASO (UCP)

Neste tópico serão relatados alguns estudos de caso que demonstram a utilização de UCP.

3.1 Estudos de Caso apresentados no artigo de Gustav Karner

- **Projeto A:** Desenvolvimento de um sistema para gerenciar a performance em redes de telecomunicações. A equipe de projeto era de poucas pessoas e estas eram iniciantes no processo de desenvolvimento adotado, mas eram pessoas muito motivadas. A estimativa foi a seguinte:

UUCP(Unadjusted Use Case Point) = 110	Número de Atores = 5 atores de complexidade média. Número de Casos de uso = 10 casos de uso de complexidade média.
TCF(Technical Complexity Factor) = 1	Todos os fatores tiveram atribuição de valor 3.
EF(Environmental Factor) = 0,975	O fator Familiaridade com o processo de desenvolvimento orientado a objetos adotado teve atribuição de valor 1. O fator Motivação teve atribuição do valor 5. O fator Requisitos Estáveis teve atribuição do valor 4. Os demais fatores tiveram atribuição do valor 3
UCP = UUCP * TCF * EF = 107.25	Multiplicando o valor de UCP por 20 horas/homem dá um total de 2145 horas.

UUCP(Unadjusted Use Case Point) = 110	<p>Número de Atores = 5 atores de complexidade média.</p> <p>Número de Casos de uso = 10 casos de uso de complexidade média.</p>
	Os projeto levou 2150 horas para ser realizado o que representa $2150/107.25 = 20,046...$

- **Projeto B:** Desenvolvimento de um projeto de gerenciamento de redes LAN. A equipe era composta de pessoas que não tinha nenhuma experiência anterior no processo de desenvolvimento adotado.

UUCP(Unadjusted Use Case Point) = 510	<p>Número de Atores = 5 atores de complexidade média.</p> <p>Número de Casos de uso = 50 casos de uso de complexidade média.</p>
TCF(Technical Complexity Factor) = 1	Todos os fatores tiveram atribuição de valor 3.
EF(Environmental Factor) = 1,175	<p>O fator Familiaridade com o processo de desenvolvimento orientado a objetos adotado teve atribuição de valor 1.</p> <p>O fator Requisitos Estáveis teve atribuição do valor 1</p> <p>Os demais fatores tiveram atribuição do valor 3</p>
UCP = UUCP * TCF * EF = 599.25	<p>Multiplicando o valor de UCP por 20 horas/homem dá um total de 11985 horas</p> <p>Os projeto levou 12.500 horas para ser realizado o que representa $12500/599.25 = 20,859....$</p>

- **Projeto C:** Desenvolvimento de um sistema de telecomunicações. A equipe não sabia muito sobre o processo de desenvolvimento adotado.

UUCP(Unadjusted Use Case Point) = 160	Número de Atores = 5 atores de complexidade média. Número de Casos de uso = 15 casos de uso de complexidade média.
TCF(Technical Complexity Factor) = 1	Todos os fatores tiveram atribuição de valor 3.
EF(Environmental Factor) = 1,175	O fator Familiaridade com o processo de desenvolvimento orientado a objetos adotado teve atribuição de valor 1. O fator Requisitos Estáveis teve atribuição do valor 2 O fator Experiência em Orientação a Objetos teve atribuição do valor 2. Os demais fatores tiveram atribuição do valor 3
UCP = UUCP * TCF * EF = 188.0	Multiplicando o valor de UCP por 20 horas/homem dá um total de 3760horas Os projeto levou 5400 horas para ser realizado o que representa $5400/188.0 = 28,723..$

3.2 Estudos de Caso apresentados no artigo de Viviane Heimberg e Everaldo Artur Grahl.

Este artigo relata a análise de 3 projetos: Sistema de Cálculo em Folha de Pagamento(Projeto 1), Sistema Contábil (Projeto 2) e o sistema de Cartão-Ponto(Projeto 3). Para cada projeto foi realizada as fases de concepção e a primeira iteração da fase de elaboração do Processo Unificado. Segue abaixo a tabela com os resultados da estimativa realizada após a primeira iteração da fase de elaboração:

Projetos	Atores	Casos de Uso	FCT	FCA	PCUNA	PCUA	Horas Estimadas
Projeto 1	4	5	1	0,81	72	58,68	1173,6
Projeto 2	6	8	1,02	0,81	87	72,32	1446,46

Projetos	Atores	Casos de Uso	FCT	FCA	PCUNA	PCUA	Horas Estimadas
Projeto 3	7	4	1,03	0,81	64	53,72	1074,5

Observação: Este artigo utilizou uma nomenclatura traduzida e que difere um pouco da proposta pelo método do Karner:

FCT equivale a TCF

FCA equivale a EF

PCUNA equivale a UUCP

PCUA equivale a IUCP

Para o cálculo das horas estimadas foi utilizado o índice de 20 horas/homem. Porém, após verificar os resultados os membros das equipes de desenvolvimento e os coordenadores dos projetos acharam que a estimativa estava muito acima dos obtidos em projetos semelhantes. Os coordenadores identificaram que 20 horas/homem por total de tempo de unidade de PCUA(UCP) era um número muito alto e não representava corretamente uma boa média para todos os tipos de tempos por nível de complexidade de casos de uso.

Além disso, a metodologia de desenvolvimento de sistemas da empresa **utiliza uma camada de código que abstrai grande parte da geração de código básica dos seus sistemas**. Sendo assim, optaram por reduzir a quantidade de horas/homem para uma média de 10 horas e ajustaram o pesos dos casos de uso e o resultado foi o seguinte:

Projetos	Atores	Casos de Uso	FCT	FCA	PCUNA	PCUA	Horas Estimadas
Projeto 1	4	5	1	0,81	54	43,74	437,4
Projeto 2	6	8	1,02	0,81	76	61,5	615,6
Projeto 3	7	4	1,03	0,81	57	46,17	461,7

Observação: Este artigo utilizou uma nomenclatura traduzida e que difere um pouco da proposta pelo método do Karner:

FCT equivale a TCF

FCA equivale a EF

PCUNA equivale a UUCP

PCUA equivale a UCP

4 EXTRAINDO MAIS DADOS A PARTIR DA ESTIMATIVA

- **Valores obtidos a partir da Métrica**

- UCP (Ajustados): Veja o item 2.4 deste documento.
- Esforço Estimado: Veja o item 2.5 deste documento.
- Produtividade: Veja o item 2.5 deste documento.

- **Valores Pré-definidos**

- Mês: representa a quantidade de dias úteis por mês.
- Horas/Dia: Representa as horas úteis (trabalhadas) por dia.
- Horas/Mês: Representa o número de horas trabalhadas em um mês. Resulta da multiplicação do número de dias úteis mês pelo número de horas de trabalho diário.
- Tempo Disponível: Tempo dado em meses, determinado pelo usuário como prazo para a realização do projeto como um todo. É relevante salientar que este tempo independe da estimativa oferecida pela métrica. Esse valor deve ser discutido com o Gerente de Projetos e deve atender às necessidades do cliente.

- **Distribuição de Esforço para o Projeto**

- **Pessoas Alocadas na Fase Contratação:** Representa o número de pessoas alocadas para a realização da fase de contratação, é possível modificar este valor de acordo com a preferência do usuário.

- **Estimativa por Subprojetos (Desconsiderando a Fase de Contratação)**

- Unidades: quantidade de casos de uso agrupadas por tipo (simples/médio/complexo).

- Subprojetos: Divide o projeto em Subprojetos, informando a quantidade de casos de uso simples, médio e/ou complexo que o o compõe.
- **Tempo Estimado para realizar cada Fase (Considerando o RH Disponível)**
 - Recursos Humanos Disponíveis para realizar cada fase (por Subprojeto): número de Recursos Humanos que estará disponível para a execução de cada fase.

5 CONCLUSÃO

Os estudos de casos relatados neste documento demonstram a importância da adoção de uma técnica de medição de software. A técnica adotada pode ser ajustada ao longo do seu uso tendo como base dados histórico e aplicação de novos padrões de desenvolvimento.

6 REFERÊNCIAS

KARNER, G. Use Case Points: resource estimation for Objectory projects. Objective Systems SF AB (copyright owned by Rational/IBM), 1993.

HEIMBERG, V; GRAHL,E : Estudo de Caso de Aplicação da Métrica de Pontos de Casos de Uso numa Empresa de Software, www.inf.furb.br/seminco/2005/artigos/130-vf.pdf.

Andrade, Edméia Leonor Pereira. Pontos de Caso de Uso e Pontos de Função na gestão de estimativa de tamanho de projetos de software orientados a objetos <http://www.bfpug.com.br/Artigos/UCP/Tese%20Edmeia.zip>