



Business Intelligence

PUC
RIO

Sergio Renato Pinto Lima

Estimativa do tamanho de um sistema, através de Pontos de Caso de Uso, considerando incertezas e avaliação de risco

Monografia de Final de Curso

Monografia apresentada ao Departamento de Engenharia Elétrica da PUC/Rio como parte dos requisitos para a obtenção do título de Especialização em Business Intelligence. Aprovada pela Comissão Examinadora abaixo assinada.

Orientador:

Prof. Dr. Juan Guillermo Lazo Lazo

Rio de Janeiro, 23 nov. 2011

Dedicatória

Para minha esposa e amiga Evelyn, meus filhos Carlos Frederico e Ana Carolina e meu neto Gabriel Lima minha inspiração no meio do caminho.
Amo vocês.

Agradecimentos

Esta monografia tem o intuito de tentar realizar uma tarefa que é de suma importância na atividade de TI, no que pese a necessidade de se dimensionar temporal e economicamente projetos de TI.

Agradeço ao Programa de Pós-Graduação em BI Master da PUC, pelo suporte e pelo conjunto de conhecimentos que me foram confiados.

Algumas pessoas foram de um valor incomensurável para a realização deste trabalho. Dentre elas, quero agradecer ao meu orientador Prof. Dr. Juan Guillermo Lazo Lazo. Agradeço-lhe pela compreensão e pelo apoio. Sua calma me deu a tranquilidade necessária para a produção da pesquisa, enquanto sua sobriedade me fez refletir sobre a necessidade de um rigor metódico e disciplinado na buscas das respostas.

Resumo

Esta monografia trata sobre o uso de modelos estatísticos, relativos ao tratamento da incerteza, nos diagramas de Casos de Uso que são largamente utilizados na análise de sistemas orientados a objetos e que descrevem as funcionalidades de um sistema a partir dos requerimentos dos usuários. O uso de modelos estatísticos visa aprimorar a técnica de estimativa por Pontos de Caso de Uso, elaborada por Gustav Karner da Objectory (posteriormente Rational Software e atualmente IBM) em 1993. A métrica de Pontos de Caso de Uso baseia-se em duas técnicas bastante difundidas: a técnica de Pontos de Função e a metodologia MK-II. Este trabalho apresenta uma abordagem no uso da incerteza, com o objetivo de melhorar a qualidade quantitativa da análise de custo e fornecer uma estimativa do risco do projeto, no desenvolvimento de projetos em ambientes que utilizem técnicas de orientação a objetos.

ABSTRACT

This monograph deals with the use of statistical models, for the treatment of uncertainty in Use Case diagrams that are widely used in the analysis of object-oriented systems and describe the features of a system from the user requirements. The use of statistical models aims to improve the estimation technique by Use Case Points, developed by Gustav Karner of Objectory (later Rational Software and now IBM) in 1993. The metric of Use Case Points is based on two very popular techniques: the technique of Function Points and MK-II methodology. This paper presents an approach in the use of uncertainty, with the aim of improving the quality of quantitative cost analysis and risk of project, project development environments using object oriented techniques.

Sumário

Dedicatória.....	9
Agradecimentos	10
Resumo.....	11
Sumário.....	12
Índice de Figuras	14
Índice de Tabelas.....	15
1. Introdução	16
1.1 Motivação.....	10
1.2 Objetivos do Trabalho	11
1.3 Descrição do Trabalho	12
1.4 Organização da Monografia.....	13
2. Descrição do Problema	11
2.1 Medição de Sistema e Estimativa de Custo	11
2.1.1 Experiência do Projetista	11
2.1.2 Comparação	16
2.1.3 Modelos de Custo	16
2.1.4 Técnica de Pontos de Função	17
2.1.5 Análise por Ponto de Função MKII.....	19
2.1.6 A Linguagem Unificada de Modelagem (UML).....	21
2.1.7 Emprego de Diagramas de Sequencia para se Demonstrar Complexidade	22
2.2 Casos de Uso e Estimativas por Caso de Uso	22
2.2.1 A História do Caso de Uso	22
2.2.2 Atores e Metas	23
2.2.3 O Modelo Gráfico de Caso de Uso	23
2.2.4 Cenários e Relacionamentos	24
2.3 Casos de Uso e Diferentes Técnicas de se Dimensionar Projetos de Sistemas	24
2.3.1 Mapeando Casos de Uso na Análise por Ponto de Função.....	24
2.3.2 Estimativa de Caso de Uso e Linhas de Código	25
2.3.3 Casos de Uso e Pontos de Função.....	25
2.3.4 Experiência Internacional.....	25
3. Metodologias.....	27
3.1 A Técnica de Pontos de Caso de Uso	27
3.1.1 Qualificando Atores e Casos de Uso	27
3.1.2 Determinantes Técnicos e Ambientais.....	28
3.1.3 Problemas com a Contagem de Casos de Uso	29
3.1.4 Gerando Estimativas Baseadas em Pontos de Caso de Uso (PCU).....	30
3.1.5 Produzindo Casos de Uso	30
3.1.6 A Descrição Textual dos Casos de Uso.....	31
3.1.7 Compondo os Casos de Usos	31
3.1.8 Contando Extensões e Inclusões de Casos de Uso	32
3.2. Incerteza na Medida.....	32

3.2.1 Como é avaliada a Incerteza	33
3.2.2 Gestão de Riscos.....	33
3.2.3 Gerenciamento de Riscos em Projetos.....	39
3.2.4 Método Monte Carlo (MC).....	40
4. Arquitetura do Sistema Proposto	44
5. Resultados	52
6. Conclusões e Trabalhos Futuros	56
Referências Bibliográficas.....	57
Apêndice 1 – Atores.....	60
Apêndice 2 – Fatores Técnicos.....	61
Apêndice 3 – Fatores Ambientais	63

Índice de Figuras

Figura 1: Distribuição Triangular - Distribuição Contínua.....	41
Figura 2: Distribuição Triangular - Função de Densidade de Probabilidade	42
Figura 3: Distribuição triangular para a variável Segurança.....	43
Figura 4: Tela do programa @Risk, configurada para 10.000 simulações.....	44
Figura 5: Tela do programa @Risk configurada para simulação Monte Carlo	45
Figura 6: Distribuição triangular com mínimo de 1 e máximo de 3	46
Figura 7: Distribuição triangular com mínimo de 0 e máximo de 5	47
Figura 8: Sumário da estatística, do campo Esforço Total em Horas, gerado pelo @Risk, após 10.000 iterações	54

Índice de Tabelas

Tabela 1: Número de Pontos de Função Não Ajustados Por Tipo de Função	18
Tabela 2: Características Gerais do Sistema.....	18
Tabela 3: Grau de Influência.....	19
Tabela 4: Características gerais das aplicações.....	20
Tabela 5: Peso dos Atores.....	27
Tabela 6: Peso de Casos de Uso por número de entidades.....	28
Tabela 7: Peso de Casos de Uso por número de transações.....	28
Tabela 8: Fatores Técnicos.....	29
Tabela 9: Fatores Ambientais	30
Tabela 10: 21 cenários de sistemas a serem desenvolvidos	45
Tabela 10: Aba Atores	47
Tabela 12: Aba Casos de Uso	48
Tabela 13: Aba Fatores Técnicos	49
Tabela 14: Aba Fatores Ambientais.....	50
Tabela 15: Aba Parâmetros	51
Tabela 16: Tabela contendo o número de horas calculado com a métrica original de Karner e com a métrica acrescida de incerteza...52	
Tabela 17: Tabela contendo o número de horas calculado com a métrica original de Karner e com a métrica acrescida de incerteza e classificada de forma ascendente com base na variação percentual.53	
Tabela 18: Tabela contendo o número de horas calculado com a métrica original de Karner e com a métrica acrescida de incerteza...53	
Tabela 19: Tabela contendo o número de horas calculado com a métrica original de Karner e com a métrica acrescida de incerteza para 10.000 e 100.000 iterações.	54
Tabela 20: Descrição dos Atores	60
Tabela 21: Descrição dos Fatores Técnicos.....	61
Tabela 22: Descrição dos Fatores Ambientais.....	63

1. Introdução

A partir da década de 70, com a popularização dos sistemas de médio porte e da microinformática, houve uma explosão na demanda no desenvolvimento de sistemas, que se espalhou por todos os segmentos da economia, bem como por todas as áreas do conhecimento humano. Neste contexto, a informática, como um todo, começou a se especializar cada vez mais ativamente, ou seja, se na década de 70 existiam, basicamente, os especialistas em teleprocessamento, banco de dados e sistemas aplicativos, nos dias de hoje o nível de especialização criou funções do tipo arquiteto de banco de dados, especialista em rede, Analista de Sistemas para ambiente C ou D (ex. Java, .NET, etc).

Neste contexto tão capilarizado e mais do que nunca, as organizações padronizadoras mundiais começaram a se preocupar em estabelecer padrões e racionalizar, desde como deveria ser feito o levantamento inicial no interessado em ter o seu sistema desenvolvido, até na sugestão de quais artefatos lhe deveriam ser entregues ao término do trabalho. Porém, o principal problema que persiste nesse relacionamento técnico versus cliente esbarra sempre na necessidade de se quantificar o tempo necessário para se desenvolver a atividade solicitada. Além disso, qual a quantidade de recursos humanos a serem alocados nessa empreitada, que, na prática, permite avaliar o custo do projeto a ser desenvolvido.

Aliás, alguns autores ponderam que esta é uma das atividades mais críticas no relacionamento de um desenvolvedor com o seu cliente e dizem mais: esta etapa da interação pode ser a fonte do sucesso ou do fracasso de uma empresa de desenvolvimento de sistemas.

Daí surge à necessidade de se criar formas de avaliação e quantificação das atividades de desenvolvimento de sistemas, que, a partir das necessidades vocalizadas pelo cliente, possam ser traduzidas em homens/hora com a menor possibilidade de erro e com a maior flexibilidade possível, para se adequar às mudanças de escopo e ao próprio aprendizado do cliente durante o processo de desenvolvimento.

A utilização de métricas, somada à experiência acumulada das equipes de desenvolvimento, pode auxiliar na geração de técnicas e ferramentas que favoreçam, com o mínimo de erro possível, a determinação da quantidade de recursos humanos e, por conseguinte, do custo de um projeto de sistema.

Por outro lado, é sabido que qualquer medição está sujeita a erros e imperfeições. Algumas delas são devido a efeitos aleatórios, tais como diferenças das experiências dos técnicos envolvidos nos projetos, maior ou menor facilidade de uso das ferramentas de desenvolvimento e testes ou mesmo da capacidade do usuário em explicitar as suas necessidades. Outras envolvem as características intrínsecas, relativas ao próprio sistema a ser desenvolvido, tais como se ele é um sistema distribuído ou não, se tem níveis de performances mínimos exigidos, se tem níveis complexos de instalação, se requer altos níveis de manutenção, se requer características especiais de segurança, etc.

Este trabalho apresenta a utilização de métodos estatísticos de análise de incerteza, visando uma melhor acurácia, na métrica da Análise de Custo no desenvolvimento de projetos, baseados no método de Pontos de Caso de Uso.

1.1 Motivação

O que motivou o desenvolvimento do presente trabalho foi a necessidade de se preencher a lacuna existente no cálculo do esforço no desenvolvimento de projetos de sistemas, utilizando-se a técnica de Ponto de Casos de Uso, conforme estabelecido originalmente no trabalho de Karner (KARNER, 1993) e a inexistência, neste mesmo trabalho, de se considerar qualquer tipo de incerteza nos dados que são informados e calculados, permitindo associar o risco associado a um projeto.

Se por um lado a métrica de Pontos de Caso de Uso ajuda ao especialista a dimensionar o esforço a ser despendido no desenvolvimento de um projeto de software, por outro lado ignora toda e qualquer incerteza e os riscos implícitos neste mesmo desenvolvimento.

Como ignorar os riscos inerentes ao dia a dia nas atividades onde o homem está envolvido? De fato, os riscos envolvidos em qualquer projeto são tão grandes que existem organizações que pensam diuturnamente nos meios para gerenciá-los e mitigá-los.

A presente monografia visa associar a métrica de Pontos de Caso de Uso, a análise de incerteza e ao cálculo do risco associados ao desenvolvimento de sistemas.

1.2 Objetivos do Trabalho

A presente monografia pretende acrescentar a medida da incerteza (risco), nos cálculos realizados pela métrica de Pontos de Caso de Uso, desenvolvido por Karner e a quantificação da estimativa de risco associado ao projeto.

Para tanto, seguimos os seguintes passos:

- 1) Definimos o método para análise quantitativa das variáveis com incerteza;
- 2) Selecionamos a melhor distribuição de probabilidade para as incertezas que pretendemos analisar;
- 3) Implementamos a métrica original, estabelecida por Gustav Karner, através de uma planilha Excel;
- 4) Criamos uma segunda planilha Excel com a métrica original, acrescida da colocação, em cada um dos campos da métrica, das distribuições de probabilidade para representar as incertezas;
- 5) Fizemos uso da simulação Monte Carlo para a geração de cenários de comportamento das variáveis com incerteza, utilizando-se uma distribuição triangular de probabilidade, via o software @Risk da Palisade;
- 6) Simulamos um conjunto de sistemas que foram calculados pelos dois métodos e estimamos o risco do projeto; e
- 7) Ao final, comparamos os valores calculados em horas, para cada conjunto idêntico, e determinamos se houve variação nos valores encontrados.

1.3 Descrição do Trabalho

O desenvolvimento da presente monografia envolveu 6 etapas: estudo da métrica de Pontos de Casos de Uso, a sua comparação com os métodos de Pontos de Função e MKII, o estudo da incerteza e do risco, o estudo da simulação Monte Carlo, o desenvolvimento da métrica de Pontos de Caso de Uso e da mesma métrica incluindo o cálculo da incerteza e risco e o estudo de casos.

O estudo da métrica de Pontos de Casos de Uso para o dimensionamento do esforço a ser despendido, surge da necessidade de se criar formas de avaliação e quantificação das atividades de desenvolvimento de sistemas, que, a partir das necessidades vocalizadas pelo cliente, possam ser traduzidas em recursos (homens/hora) com a menor possibilidade de erro e com a maior flexibilidade possível, para se adequar às mudanças de escopo e ao próprio aprendizado do cliente durante o processo de desenvolvimento.

O estudo dos métodos de Pontos de Função e do método MKII, para a avaliação do esforço no desenvolvimento de sistemas, busca compreender onde estas técnicas apresentam vantagens ou desvantagens, quando comparadas a métrica de Pontos de Casos de Uso preconizada por Gustav Karner.

Já o estudo da incerteza e do risco visa apresentar e ressaltar que não é suficiente se informar se um determinado caso de uso é de alta, média ou baixa complexidade, ou ainda se a experiência da equipe com orientação a objetos é pouca, suficiente ou grande, pois todas estas informações contém um alto nível de insegurança. E esta insegurança (incerteza) precisa ser necessariamente mensurada e agregada aos valores do esforço calculado em qualquer métrica, seja ela Pontos de Casos de Uso, Pontos de Função ou MKII.

O estudo da simulação Monte Carlo visa apresentar um método que permite, dentre outras coisas, gerar aleatoriamente N sucessivos cenários em termos de custo ou tempo (variável aleatória), que serão criados, a partir de um modelo estatístico, que vem a ser uma distribuição de probabilidade, no caso presente uma distribuição triangular, para uma determinada variável com incerteza no Projeto. Cada amostra ou cenário corresponde a uma iteração do método. Desse modo, o método de Monte Carlo fornece uma estimativa do valor de um tempo ou custo esperado assim como um erro para esta estimativa.

Após as pesquisas bibliográficas, desenvolveram-se duas planilhas Excel com o modelo de Pontos de Casos de Uso sem e com incerteza, onde esta última faz uso da simulação Monte Carlo.

Por fim, passou-se a etapa de estudos de caso, onde o modelo original de Pontos de Casos de Uso foi avaliado, comparativamente, com o seu paradigma usando a simulação Monte Carlo. Esta fase envolveu a simulação de projetos de desenvolvimento de sistemas com diversos tamanhos e complexidades.

1.4 Organização da Monografia

A presente monografia está dividida em cinco capítulos adicionais, descritos a seguir:

O capítulo 2 apresenta a descrição do problema, relacionado às diversas métricas de estimativa de esforço no desenvolvimento de um sistema de software, incluindo a técnica de Pontos de Função, Pontos de Função MKII, a técnica de Pontos de Casos de Uso e o seu mapeamento entre as diversas técnicas abordadas.

O capítulo 3 descreve mais detidamente a métrica de Pontos de Casos de Uso, os conceitos relacionados à Incerteza, ao Risco no desenvolvimento de Projetos, às formas de avaliação de incertezas, ao método Monte Carlo.

O capítulo 4 descreve as técnicas empregadas para se desenvolver estimativas de esforço de desenvolvimento de projetos, utilizando a métrica de Pontos de Casos de Uso associada à análise da incerteza e risco envolvidos, nestas estimativas.

O capítulo 5 detalha os resultados obtidos e compara os valores obtidos pela métrica de Pontos de Casos de Uso com e sem o cálculo da incerteza e do risco.

Finalmente, o capítulo 6 descreve as conclusões do trabalho e identifica possíveis trabalhos futuros.

2. Descrição do Problema

Medir e estimar são dois aspectos ou estágios do procedimento de estimativa. Diversas abordagens têm sido utilizadas com maior ou menor frequência pelos desenvolvedores (KITCHENHAM *et al.*, 1997).

Da mesma forma que medimos, precisamos ter em mente que qualquer medida guarda imperfeições. No caso presente, ela pode estar associada a efeitos aleatórios, tais como a capacidade dos profissionais que irão desenvolver a aplicação, a mudança do escopo do projeto no curso do seu desenvolvimento e a qualidade da liderança. Outras imperfeições estão associadas à questões inerentes ao projeto, tais como a necessidade de alocação da equipe em meio período, a dificuldade na programação, a características especiais de segurança, aos diversos níveis de portabilidade, manutenibilidade, usabilidade e tantas outras que são requeridas na análise de Pontos por Caso de Uso.

Estas imperfeições, quando vistas pelo ponto de vista técnico, são conhecidas incertezas da medição. A definição do termo incerteza (da medição) é obtida da versão adotada para o Vocabulário Internacional de Termos Básicos e Genéricos em Metrologia (INMETRO, 2000) é:

“Um parâmetro associado ao resultado de uma medição, que caracteriza a dispersão de valores que poderiam ser razoavelmente atribuídas ao mensurando”.

2.1 Medição de Sistema e Estimativa de Custo

O tamanho de um sistema pode ser definido por atributos internos: funcionalidade e complexidade, podendo ser medido estaticamente sem a execução ou o desenvolvimento do sistema. A possibilidade de reuso de código é um aspecto do tamanho. Extensão é o tamanho físico do produto e pode ser medido pela especificação, desenho e pelo código. A complexidade refere-se tanto à eficiência, quanto à complexidade das funcionalidades (FENTON & PFLEEGER, 1997).

2.1.1 Experiência do Projetista

A opinião dos projetistas é feita com base nas experiências passadas. Em geral, a opinião de um técnico experiente resulta em estimativas acuradas, todavia isto é inteiramente dependente da vivência do técnico (BOEHM, 1981).

A vantagem é que o método incorpora o conhecimento das diferentes experiências entre os projetos passados (FENTON & PFLEEGER, 1997). A desvantagem deste método é que as estimativas dependem da efetiva experiência do técnico neste tipo de projeto e que eles podem ser tendenciosos ou não, ao fazerem uma boa análise.

2.1.2 Comparação

A comparação é a forma mais utilizada pelos técnicos experientes: o estimador compara o projeto proposto, com um ou mais projetos. Diferenças e similaridades são identificadas e utilizadas para ajustar as estimativas. O estimador, tipicamente, identifica o tipo da aplicação, estabelece uma previsão inicial e, então, refina sua previsão no contato com os termos do projeto. A acurácia da analogia é dependente das informações históricas de projetos anteriores (FENTON & PFLEEGER, 1997).

Mais uma vez, a incerteza se apresenta, pois apesar de projetos se parecerem, eles nunca são idênticos, logo precisamos avaliar esta incerteza.

2.1.3 Modelos de Custo

Modelos de custo são algoritmos que descrevem medidas de entrada, normalmente uma medida do tamanho do produto, com algumas medidas de saída, tais como esforço ou duração do projeto. Modelos de custo fornecem estimativas diretas de esforço e se apresentam de duas formas: equações matemáticas e tabelas com pareceres (KITCHENHAM *et al.*, 1997). Equações matemáticas utilizam tamanho, como variável de entrada, e esforço, como variável de saída. Eles frequentemente incluem um número de fatores de ajuste. Estes fatores de ajuste influenciam a produtividade e estão relacionadas, de forma subjetiva, com experiência do programador: muito bom, bom, médio, fraco, muito fraco.

As vantagens dos modelos de custo são que eles podem ser utilizados por pessoas sem experiência. A desvantagem é que a fórmula tem que ser atualizada, para permitir a troca dos métodos de desenvolvimento. Os modelos assumem que o futuro é igual ao passado e que seus resultados se aplicam à média dos projetos.

Originalmente os diversos modelos de custos foram desenvolvidos no passado, baseados em modelos de regressão.

2.1.4 Técnica de Pontos de Função

A técnica de pontos de caso de uso é baseada na Análise por Ponto de Função (APF), desenvolvida por Allan Albrecht e na menos conhecida Análise por Ponto de Função MK II, que faz uma adaptação e melhoria da técnica proposta por Albrecht. Para se entender a técnica de Pontos de Caso de Uso, é necessário se conhecer as duas técnicas em que ele se baseia APF e MK II. A técnica de Pontos de Caso de Uso é mais semelhante com a técnica MK II, do que da tradicional técnica por Pontos de Função.

O Manual de Práticas de Contagem, do Grupo de Usuários Internacional (sigla em inglês IFPUG), identifica cinco tipos de funções, que são contados e avaliados:

1. Arquivo Lógico Interno (ALI)
2. Arquivo de Interface Externa (AIE)
3. Entrada Externa (EE)
4. Saída Externa (SE)
5. Consulta Externa (CE)

Arquivo Lógico Interno (ALI)

Um ALI é uma entidade lógica e persistente, a respeito da qual dados serão mantidos. Os ALIs baseiam-se em requisitos lógicos dos usuários e são independentes da implementação ou dos meios de armazenamento, tais como tabelas ou bancos de dados. Um ALI é contado com base em uma avaliação do número de campos de dados não recursivos do usuário e do número de tipos de elementos de registros lógicos nele contidos.

Arquivo de Interface Externa (AIE)

Um AIE é uma entidade lógica e persistente, que é requerida para referência ou validação pelo sistema sendo contado, mas que é mantido por outro aplicativo. Isto é, um arquivo de interface externa deve ser um arquivo lógico interno para um outro aplicativo. De forma semelhante a um ALI, um AIE é avaliado com base no número de campos de dados não recursivos do usuário e no número de tipos de elementos de registros lógicos. Arquivos de Interface Externa também são partes dos requisitos lógicos dos usuários.

Entrada Externa (EE)

Uma entrada externa é um processo lógico do negócio, que mantém os dados em um ou mais arquivos lógicos internos, ou é um processo de controle que direciona o sistema para atender os requisitos de negócio do usuário. Uma entrada externa é contada com base no número de campos de dados do usuário envolvidos e na soma dos ALI e AIE participantes do processo. Um exemplo de EE seria “Incluir empregado” em um aplicativo de recursos humanos.

Saída Externa (SE)

Uma saída externa é um processo lógico do negócio, que gera dados para um usuário ou para outro aplicativo externo ao sistema. Exemplos típicos de

Saídas Externas incluem relatórios de usuários, geração em arquivos externos. As Saídas Externas também fazem parte dos requisitos lógicos dos usuários.

Consulta Externa (CE)

Uma consulta externa é o último tipo de função lógica de usuário, contada através de pontos de função. Uma CE consiste em um par gatilho-resposta (ou pergunta-resposta), através do qual a pergunta ou solicitação de dados entra no aplicativo, vinda de fora (tipicamente a partir de um usuário ou de outro aplicativo). Os dados são recuperados para atender à solicitação e enviados para fora. Mais uma vez, as Consultas Externas fazem parte dos requisitos lógicos dos usuários de um sistema.

Cada tipo de função (ALI, AIE, EE, SE, CE) é avaliado, recebendo complexidade Baixa, Média ou Alta, utilizando-se um método direto (delineado no Manual de Práticas de Contagem).

A tabela 1 do IFPUG resume o número de pontos de função não ajustados atribuídos a cada tipo de função:

Tabela 1: Número de Pontos de Função Não Ajustados Por Tipo de Função

Tipo de Função	Baixa	Média	Alta
EE	3	4	6
SE	4	5	7
CE	3	4	6
ALI	7	10	15
AIE	5	7	10

O passo final na contagem de pontos de função envolve o ajuste da contagem, através de um Fator de Ajuste de Valor (FAV), o qual avalia restrições de negócio adicionais do sistema não consideradas pelos cinco tipos de funções.

O FAV é baseado em 14 características gerais de sistema, que determinam a funcionalidade geral da aplicação que está sendo contada, conforme apresentado na tabela 2.

Tabela 2: Características Gerais do Sistema

Fator	Descrição
T1	Comunicação de dados
T2	Processamento distribuído de dados
T3	Desempenho
T4	Configuração intensamente utilizada
T5	Taxa de transação
T6	Atualização on-line
T7	Eficiência do usuário final
T8	Atualização on-line

T9	Processamento complexo
T10	Reutilização
T11	Facilidade de instalação
T12	Facilidade de operação
T13	Múltiplas localidades
T14	Facilidade de alteração

O nível (grau) de influência varia em uma escala de 0 a 5 conforme a tabela 3.

Tabela 3: Grau de Influência

Nível	Influência
0	Nenhuma influência
1	Influência mínima
2	Influência moderada
3	Influência média
4	Influência significativa
5	Influência forte

Após serem classificados os quatorze itens, de acordo com o nível de influência (NI), aplica-se a fórmula:

$$\mathbf{FAV = (NI * 0,01) + 0,65}$$

Para se determinar os Pontos de Função Ajustados para um projeto que envolva desenvolvimento, utiliza-se a seguinte fórmula:

$$\mathbf{PFD = (PFB + PFC) * FAV}$$

Onde:

PFD = Pontos de Função de Desenvolvimento

PFC = Ponto de Função de Conversão

PFB = Ponto de Função Bruto da Aplicação antes do projeto de manutenção

A mensuração funcional de tamanho e pontos de função não é uma panaceia, simplesmente oferece um processo objetivo e replicável para avaliar o tamanho lógico do sistema com base em requisitos funcionais dos usuários (DEKKERS, 1998).

2.1.5 Análise por Ponto de Função MKII

A análise pela técnica de ponto de função MKII é uma variação da análise de pontos de função, usado inicialmente na Inglaterra. A técnica foi proposta por Charles SYMONS para se obter um melhor controle dos processos internos complexos.

SYMONS observou uma série de problemas com a contagem de pontos de função original, entre a escolha dos valores complexos, médios e simples. Isto significava que muitos itens complexos não eram adequadamente pesados.

Pontos de função MKII, assim como a técnica original de pontos de função não eram muitos proveitosos para se estimar projetos de sistemas, que incluíam muitos cálculos embutidos ou realizados em tempo real, projetos que tenham um grande número de algoritmos ou projetos para a internet (RULE, 2001).

No lugar dos cinco tipos de componentes definidos na técnica de pontos de função, o MKII vê os sistemas como uma coleção de transações lógicas, onde cada transação consiste de uma entrada, um processamento e uma componente de saída (SYMONS, 1991). Um tipo de transação lógica é definido, como uma única combinação entrada/processamento/saída, por exemplo: criar um cliente, atualizar uma conta, consultar o status de um pedido e gerar um relatório.

Por causa da semelhança entre os dois conceitos, as duas abordagens podem ser combinadas para produzir melhores estimativas sobre certas circunstâncias.

A técnica de ponto de função MKII modifica a técnica da análise de pontos de função, por estender a sua lista de fatores de ajustes técnicos. Estes fatores foram descartados há alguns anos atrás, porque eles não faziam mais sentido no modelo de desenvolvimento de sistemas modernos. Para ambas as técnicas, FPA e MKII FPA os fatores de ajuste foram desacreditados como não realistas. Por isto muitos profissionais têm ignorado os fatores de ajustes e trabalhado usando os pontos de função não ajustados (RULE, 2001).

A relação das características gerais da aplicação é apresentada na tabela 4, para mostrar os fatores T15, T16, T17 e T18 adaptados da técnica de casos de uso de Karner, adicionalmente aos fatores originais propostos por Albrecht, T1 a T14 (RIBU, 2001).

Tabela 4: Características gerais das aplicações

Fator	Descrição
T1	Comunicação de dados
T2	Processamento distribuído de dados
T3	Desempenho
T4	Configuração intensamente utilizada
T5	Taxa de transação
T6	Atualização on-line
T7	Eficiência do usuário final
T8	Atualização on-line
T9	Processamento complexo
T10	Reutilização
T11	Facilidade de instalação
T12	Facilidade de operação

T13	Múltiplas localidades
T14	Facilidade de alteração
T15	Requerimentos de outras aplicações
T16	Segurança, privacidade e facilidade de auditoria
T17	Necessidade de treinamento do usuário
T18	Usado por terceiros

Esta lista pode ser comparada com os fatores técnicos utilizados no método de casos de uso, para se verificar que muitos deles são os mesmos.

2.1.6 A Linguagem Unificada de Modelagem (UML)

A Linguagem Unificada de Modelagem (UML em inglês) é utilizada para desenho e análise de sistemas orientados a objeto. A linguagem é baseada nas antigas linguagens de modelagem tais como Técnicas de Modelagem de Objeto e Engenharia de Software Orientada a Objeto (JACOBSON, 1995).

Segundo Ribu, na análise e desenho orientado a objeto, os casos de uso descrevem os requerimentos funcionais. O diagrama de caso de uso foi incorporado ao UML e é parte integrante da notação UML. O padrão da UML não discute o conteúdo ou descreve um caso de uso.

Utilizando a UML, o desenvolvedor pode construir as visões a serem seguidas, através do uso de diagramas:

- A visão funcional descreve os requerimentos com as funcionalidades básicas do sistema. Diagramas de caso de uso descrevem a visão estática e seus relacionamentos. Diagramas de atividade descrevem a visão dinâmica funcional, sendo utilizada para mostrar atividades, fluxos de trabalho, processamentos condicionais e como o trabalho é realizado na empresa.

- A visão estrutural estática define a estrutura estática do sistema e é representada pelos diagramas de classe e diagramas de objetos.

- A visão comportamental descreve o comportamento temporal do sistema usando diagramas, tais como diagramas de colaboração e diagramas de sequência, explicitando sequências de interações entre objetos em diferentes cenários.

Estados UML:

Um caso de uso é um tipo de classificador representando uma unidade coerente de funcionalidades providas pelo sistema, um subsistema, ou uma classe manifestada por sequências de mensagens trocadas entre o sistema e um ou mais atores externos, juntamente com as ações a serem realizadas pelo sistema.

O comportamento de um caso de uso pode ser descrito de diversas maneiras, dependendo do que for mais conveniente, sendo, frequentemente, textos simples, mas estados de máquinas, operações e métodos são exemplos de outros caminhos, que descrevem o comportamento de casos de uso.

O diagrama de estado é utilizado para se mostrar como um elemento, normalmente uma classe, troca de estado no curso do tempo, realiza transições permitidas e quais são as condições para estas transições. O diagrama de estado representa o comportamento das entidades dinâmicas pela especificação de suas responsabilidades no recebimento de instâncias de evento. Normalmente, este diagrama é utilizado para descrever o comportamento de classes, apesar de um estado poder descrever o comportamento de outros modelos de entidade, tais como casos de uso, atores, subsistemas, operações ou métodos.

Um diagrama de atividades é um caso especial de diagrama de estado, no qual todos os estados são ações ou estados de sub atividades, nos quais todas as transições são monitoradas até a finalização de ações ou de sub atividades no estado principal.

Um diagrama de classe é um gráfico de classificação dos elementos conectados por vários relacionamentos estáticos. Um diagrama pode conter interfaces, pacotes, relacionamentos e nivelar instâncias, tais como objetos e ligações. Uma interação definida em um contexto de colaboração especifica os detalhes das comunicações que devem ser realizadas até a conclusão de uma tarefa particular.

Uma comunicação é especificada como uma mensagem onde são definidos os papéis da instância remetente e da recebedora, assim como as ações que causaram a comunicação. As instruções da comunicação, são também, especificadas pela interação.

Segundo o Object Management Group um diagrama de sequencia apresenta uma interação, que é um conjunto de mensagens entre partes, visando a colaboração, para se obter uma operação ou resultado esperado.

2.1.7 Emprego de Diagramas de Sequencia para se Demonstrar Complexidade

Conforme Ribu, quando não existem casos de uso detalhados, a complexidade dos casos de uso pode ser representada pela contagem das transações dos diagramas de sequencias. Diagrama de sequencias diz respeito à visão comportamental, de acordo com o padrão UML. Eles são de um menor nível de detalhamento do que casos de uso. Deve-se contar o que fazer e não como fazer.

2.2 Casos de Uso e Estimativas por Caso de Uso

2.2.1 A História do Caso de Uso

Enquanto trabalhava na Ericson, no final dos anos 60, Ivar Jacobson idealizou o que mais tarde seria chamado de casos de uso (RIBU, 2001). Naquele

tempo, a Ericson modelava os seus sistemas utilizando um conjunto de blocos interconectados, mais tarde chamados de ‘subsistemas’ em UML. Os blocos eram fundidos, através da especificação dos ‘casos de tráfego’ previamente especificados, mais tarde conhecidos como casos de uso (COCKBURN, 2000).

Jacobson saiu da Ericson em 1987 e criou a Objectory AB em Estocolmo, onde ele e seus associados desenvolveram um processo produto chamado ‘Objeto’. Uma técnica de diagramação foi desenvolvida para conceituar o caso de uso. Em 1992, Jacobson idealizou a metodologia de sistema OOSE (abreviatura em inglês para Engenharia de Software Orientada a Objeto), que é uma metodologia para se produzir um caso de uso e onde eles envolvem todos os estágios do desenvolvimento. Isto inclui a análise, desenho, validação e teste (JACOBSON, 1992).

Em 1993, Gustav Karner desenvolveu o método de Pontos de Casos de Uso para sistemas orientados a objeto.

Em 1994, Alistair Cockbun construiu o modelo conceitual de Atores e Metas enquanto escrevia o guia de casos de uso para a IBM Consulting Group. Ele gerou as bases de como se estruturar e escrever casos de uso.

2.2.2 Atores e Metas

O termo ‘caso de uso’ implica ‘nos caminhos pelo qual um usuário utiliza o sistema’(COCKBURN 2000). Isto é, uma coleção das sequencias possíveis de interação entre o sistema sob construção e os atores externos, relacionados a uma determinada meta. Atores são pessoas ou sistemas computacionais e o sistema é uma entidade singular que interage com os atores.

A proposta do caso de uso é encontrar a meta imediata de um ator, como se fosse um pedido. Para se alcançar uma meta, algumas ações precisam ser realizadas (RICHTER, 2001). Todos os atores tem um conjunto de responsabilidades. Uma ação conecta uma meta de um ator com outras responsabilidades (COCKBURN, 1997).

Um ator primário é um ator que necessita assistência do sistema para realizar uma meta. Um ator secundário supre o sistema de ajuda para se alcançar à meta. Quando um ator primário dispara uma ação, a meta é alcançada se o segundo ator fornecer a sua ajuda (COCKBURN, 1997).

2.2.3 O Modelo Gráfico de Caso de Uso

Segundo Ribu, o modelo de caso de uso é um conjunto de casos de uso que representam o total das funcionalidades do sistema. O modelo completo também especifica as entidades externas tal como os usuários humanos e outros sistemas

que usam estas funcionalidades. A UML provê duas notações gráficas para definir um modelo funcional de sistema:

- O diagrama de caso de uso descreve uma visão estática das funcionalidades e dos seus relacionamentos com entidades externas e entre si. Palitos representam os atores e elipses representam os casos de uso.
- O diagrama de atividades confere uma visão dinâmica destas funções.

2.2.4 Cenários e Relacionamentos

Um cenário é uma instância do caso de uso, uma sequencia específica de ações que ilustram comportamentos (CHEESMAN & DANIELS, 2000). O cenário principal de sucesso descreve o que acontece na maioria dos casos comuns quando nada de errado acontece. Ele é quebrado em passos do caso de uso e são escritos em linguagem natural ou descrevem um estado ou um diagrama de atividades.

Cenários diferentes podem ocorrer e o caso de uso junta estes diferentes cenários. Casos de uso incluem relacionamentos entre eles mesmos. Desde que o caso de uso represente funções do sistema, estes relacionamentos indicam relacionamentos correspondentes entre estas funções do sistema.

2.3 Casos de Uso e Diferentes Técnicas de se Dimensionar Projetos de Sistemas

Diferentes técnicas de se dimensionar projetos de sistema orientados a objeto foram propostas nos últimos anos (RIBU, 2001). Algumas destas técnicas são apresentadas a seguir.

2.3.1 Mapeando Casos de Uso na Análise por Ponto de Função

Uma técnica para se mapear casos de uso em Pontos de Análise de Função é descrito por Thomas Fetke *et al.* O autor propõe o mapeamento dos casos de uso diretamente no modelo de ponto de função, usando um conjunto de regras concisas, que suportem o processo de medição. Os mapeamentos destas regras, são baseados no padrão definido no Manual de Práticas de Contagem do Grupo de Usuários de Pontos de Função Internacional (GUPFI, sigla em inglês).

Desde que o conceito de atores no modelo de caso de uso é mais amplo que o conceito de usuários externos da aplicação na Análise por Ponto de Função (APF), não existe um mapeamento um por um de ator para usuário externo da aplicação, mas cada usuário do sistema é definido como um ator. Da mesma maneira, todas as aplicações que se comunicam com o sistema podem ser consideradas como um ator. Isto corresponde ao método de ponto de caso de uso de Karner.

O nível de detalhamento do modelo de caso de uso pode variar, e pode não prover nenhuma informação de como se contar um caso de uso específico de acordo com as regras de ponto de função (RIBU, 2001). Para isso, na técnica de Karner os casos de uso devem ser descritos de tal forma a facilitar a contagem de transações.

A Análise por Ponto de Função pode ser utilizada no paradigma de orientação a objeto (FETKE *et al.*, 1997).

2.3.2 Estimativa de Caso de Uso e Linhas de Código

John Smith (SMITH, 1999), da Rational Software, descreve um método de estimativa, baseada em casos de uso transladados em linhas de código. Apesar de não existir nenhuma pesquisa a respeito sobre este método, algumas ferramentas, tais como ‘Estimativa Profissional’, da Software Productivity Center Inc., e a ‘CostXpert’ da Marotz Inc., produzem estimativas de esforço por caso de uso, calculadas pelo número de linhas de código (RIBU, 2001).

2.3.3 Casos de Uso e Pontos de Função

De acordo com as observações de David Longstreet da Software Metrics a aplicação de pontos de função auxilia a determinação se o caso de uso foi escrito em um nível adequado de detalhamento. Se for possível descrever como os dados passam do ator para dentro da fronteira do sistema ou como os dados fluem de dentro da fronteira da aplicação para o ator, então o nível de detalhamento está correto, de outra forma o caso de uso precisa de um detalhamento maior (KARNER, 1993).

Ao se adotar a técnica de caso de uso e a técnica de pontos de função, a qualidade dos documentos de requerimentos pode ser melhorada. Desta forma, a estimativa pode ser aperfeiçoada.

2.3.4 Experiência Internacional

No final do ano de 1993, Martin Arnold e Peter Pedross desenvolveram uma técnica de pontos de caso de uso inspirado em Karner, para o Instituto Banco Suíço. A proposta era medir o tamanho dos sistemas, baseados nas especificações de requerimentos. Nesta técnica, pontos de caso de uso são comparados diretamente com pontos de função e a técnica de pontos de função é utilizada para calibrar o método de pontos de caso de uso.

A qualidade dos documentos de requerimentos e as medidas dos pontos de caso de uso foram analisadas, no intuito de testar e calibrar a técnica de pontos de caso de uso. As análises mostraram que as especificações de requerimentos, com

o uso de casos e cenários podem ser utilizadas para se medir o tamanho de um sistema, e que as medidas podem, normalmente, ser feitas em questão de horas.

O trabalho de Bente Anda *et al.* (2001) demonstraram que o uso da técnica de pontos de caso de uso faz estimativas razoavelmente acuradas para projetos de empresas que utilizam historicamente o método em projetos.

O método foi utilizado em três projetos similares, tanto em tamanho, quanto em funcionalidades. No sentido de se estabelecer as vantagens gerais do método, os autores sugerem que é interessante se tentar a utilização dele em diferentes projetos e em diferentes companhias.

3. Metodologias

3.1 A Técnica de Pontos de Caso de Uso

Uma breve estimativa do esforço, baseada em casos de uso, pode ser feita quando existe algum conhecimento do domínio do problema, tamanho do sistema e arquitetura do estágio pelo qual é feita a estimativa (RIBU, 2001). A técnica de ponto de caso de uso é uma técnica para estimar e dimensionar o esforço de desenvolvimento de um sistema, baseado em contar casos de uso conhecido como Pontos de Caso de Uso.

3.1.1 Qualificando Atores e Casos de Uso

Pontos de caso de uso podem ser contados a partir da análise de caso de uso do sistema (KARNER, 1993). O primeiro passo é qualificar se os atores são simples, médios ou complexos. Um ator simples representa outro sistema com uma Interface definida, um ator médio é uma interação com outro sistema através de um protocolo tipo TCP/IP e um ator complexo pode ser uma pessoa interagindo através de uma interface gráfica ou uma página WEB. Um fator de peso é associado a cada um dos tipos de atores, conforme a tabela 5, abaixo:

Tabela 5: Peso dos Atores

Tipo de Ator	Peso	Descrição
Ator Simples	1	Outro sistema acessando através de uma interface de aplicação
Ator Médio	2	Outro sistema interagindo, através de um protocolo de comunicação, como TCP/IP ou FTP
Ator Complexo	3	Um usuário interagindo, através de uma interface gráfica

O peso total não ajustado (PTNA) é calculado pela contagem da quantidade de atores que fazem cada coisa (dentro da escala de complexidade), multiplicando cada total pelo seu fator de peso e somando os produtos.

Cada caso de uso é definido como simples, médio ou complexo, dependendo do número de transações em cada descrição de caso de uso, incluindo cenários secundários. Uma transação é um conjunto de atividades realizadas em conjunto ou não. A contagem do número de transações pode ser feita pela contagem dos passos do caso de uso. Karner propôs que inclusões e extensões dos casos de uso, não sejam contadas. A complexidade de um caso de uso ficou definida, conforme apresentado na tabela 6, da seguinte maneira:

Tabela 6: Peso de Casos de Uso por número de entidades

Tipo de Caso de Uso	Número de Entidades	Peso
Simple	até 3	5
Médio	de 4 à 7	10
Complexo	7 ou mais	15

Outro mecanismo para se medir a complexidade de casos de uso é a contagem das classes, que podem ser utilizadas no lugar das transações, uma vez que tenham sido determinadas quais classes implementam um caso de uso específico. Os pesos são iguais aos relatados anteriormente (SCHNEIDER & WINTERS, 1998), conforme a tabela 7, abaixo:

Tabela 7: Peso de Casos de Uso por número de transações

Tipo de Caso de Uso	Número de Transações	Peso
Simple	Até 3	5
Médio	de 4 à 7	10
Complexo	7 ou mais	15

Cada caso de uso é então multiplicado pelo seu fator de peso e os produtos são somados para se obter o peso de caso de uso não ajustado (CUNA).

O PTNA é adicionado ao CUNA, para se obter os pontos de caso de uso não ajustados (PCUNA):

$$PCUNA = PTNA + CUNA$$

3.1.2 Determinantes Técnicos e Ambientais

Conforme Karner, o método emprega também uma técnica de multiplicação de fatores correspondente ao fator de Ajuste de Complexidade Técnica do método de Análise por Ponto de Função, e um multiplicador por fator ambiental, para se quantificar os requerimentos não funcionais desde a facilidade de uso à motivação do programador.

Vários fatores que influenciam a produtividade são associados com pesos e valores e são dados a cada fator, dependendo do grau de influência. O valor 0 significa sem influência, 3 é médio, e 5 significa altamente influente (ver tabela 8 e tabela 9).

Os fatores de ajuste são multiplicados pelos pontos de caso de uso não ajustados para produzir os pontos de caso de uso ajustados, produzindo uma estimativa do tamanho do sistema.

O Fator de Complexidade Técnica (FCT) é calculado pela multiplicação do valor de cada fator (T1 – T13) pelo seu peso, quando então se adicionam todos estes números, para se obter o chamado cálculo total FatorT. A seguinte fórmula é aplicada:

$$FCT = 0.6 + (0.01 * \text{FatorT})$$

O Fator Ambiental (FA) é calculado pela multiplicação do valor de cada fator (F1 – F8) pelo seu peso, adicionando-se os produtos para se obter o somatório, denominado de FatorE. A seguinte fórmula é utilizada:

$$FA = 1.4 + (-0.03 * \text{FatorE})$$

A fórmula para cálculo dos Pontos de Caso de Uso (PCU) ajustados é o seguinte:

$$PCU = PCUNA * FCT * FA$$

3.1.3 Problemas com a Contagem de Casos de Uso

Muitas variações de estilos de casos de uso podem dificultar a medição da complexidade dos casos de uso (SMITH, 1999). Descrições textuais livres podem acarretar em especificações ambíguas (ARNOLD & PEDROSS, 1998). Levando-se em conta que existe um grande número de interpretações dos conceitos de casos de uso, SYMONS (2001) concluiu que uma forma de resolver este problema seria ver a transação lógica MKII, como um caso específico de um caso de uso e usar esta abordagem frente aos requerimentos que são mensuráveis e que tenham uma grande chance de unificar a interpretação.

Para calcular o fator de complexidade técnica do sistema, utilizamos a tabela 8 de fatores técnicos.

Tabela 8: Fatores Técnicos

Fator	Descrição	Peso
T1	Sistema Distribuído	2
T2	Tempo de Resposta	2
T3	Eficiência	1
T4	Complexidade dos Processamentos Internos	1
T5	Reusabilidade de código	1
T6	Facilidade de Instalação	0,5
T7	Facilidade de uso	0,5
T8	Portabilidade	2
T9	Facilidade de Mudança	1
T10	Usuários Simultâneos	1
T11	Características Especiais de Segurança	1

T12	Solução de Terceiros	1
T13	Requer Treinamento Especial	1

A tabela 9 mostra os fatores ambientais previstos pela metodologia de Pontos de Caso de Uso e seus pesos associados.

Tabela 9: Fatores Ambientais

Fator	<i>Descrição</i>	Peso
F1	Familiaridade com o RUP	1,5
F2	Experiência na aplicação em desenvolvimento	0,5
F3	Experiência em Orientação a Objetos	1
F4	Presença de analista experiente	0,5
F5	Motivação	1
F6	Requisitos estáveis	2
F7	Desenvolvedores em meio expediente	-1
F8	Linguagem de programação difícil	2

3.1.4 Gerando Estimativas Baseadas em Pontos de Caso de Uso (PCU)

Karner propôs um fator de 20 horas por ponto de caso de uso para uma estimativa por projeto. Segundo alguns autores, experiências de campo demonstraram que este valor varia de 15 a 30 horas por ponto de caso de uso. Schneider e Winters sugerem um refinamento da proposta de Karner, baseada no nível de experiência da equipe e da estabilidade do projeto.

O número de fatores ambientais de F1 até F6 acima de 3 são contados e adicionados ao número de fatores de F7 a F8 inferiores a 3. Se o total é 2 ou menor, eles propõe 20 horas por PCU; se o total é 3 ou 4, o valor é de 28 horas por PCU. Quando o total excede a 4 é recomendado que mudanças sejam feitas no projeto, de forma a ajustá-lo a valores inferiores.

3.1.5 Produzindo Casos de Uso

Os casos de uso de um sistema em construção devem ser escritos em níveis adequados de detalhe. Eles devem possibilitar a contagem de transações nas descrições dos casos de uso, de forma a definir a complexidade do caso de uso (RIBU, 2001). O nível de detalhamento da descrição e da estrutura do caso de uso tem impacto na precisão das estimativas baseadas em caso de uso. O modelo de caso de uso pode conter um número variável de atores e casos de uso e estes números afetariam as estimativas (ANDA *et al.*, 2001).

3.1.6 A Descrição Textual dos Casos de Uso

Os detalhes dos casos de uso devem ser explicitados em suas descrições textuais, escritas em linguagens naturais, ou nos diagramas de estado ou nos diagramas de atividade (RIBU, 2001). Uma descrição de caso de uso deve conter um nome e/ou um número que o identifique, o nome do ator que o origina e uma breve descrição do seu objetivo e uma sequência numerada de passos que descrevem o cenário de sucesso (CHEESMAN & DANIELS, 2000).

O cenário principal de sucesso descreve o que acontece na maioria dos casos, quando nada de errado acontece. Os passos são desenvolvidos exatamente na sequência apresentada. Cada passo pode ter associado um ponto de extensão com comportamentos alternativos. Por exemplo:

 Descrição de um Caso de Uso Para um Sistema de Ponto

Caso de Uso Número 1

Nome: Registro de Horas

Ator Iniciante: Empregado

Ator Secundário: Sistema Gerenciador de Projeto

Sistema de Gerenciamento de Ponto

Meta: Registro de horas trabalhadas por cada empregado em todos os projetos em que ele participe

Pré-condição: Nenhuma

CENÁRIO DE SUCESSO PRINCIPAL

1. O Sistema apresenta o calendário (Assumido: Semana Corrente)

2. O Empregado escolhe o período de tempo

3. Inclui o Caso de Uso 'Busca Projetos Válidos'

4. O Empregado seleciona o Projeto

5. O Empregado registra a hora gasta no Projeto

Repetir do item 4 até acabar

6. O Sistema atualiza o contador de tempo

EXTENSÕES

2o. Período de tempo inválido

O Sistema envia uma mensagem de erro e fica disponível para novas solicitações

 Este caso de uso consiste de 6 passos e de um passo de extensão (RIBU, 2001).

3.1.7 Compendo os Casos de Usos

A Linguagem Unificada de Modelagem (UML) não entra em detalhes sobre como o modelo de caso de uso deve ser estruturado, nem como cada caso de uso deve ser documentado (RIBU, 2001). De qualquer forma, um nível mínimo de detalhes deve ser combinado, tanto quanto necessário, para se estabelecer à captura de todas as funcionalidades do sistema para os casos de uso. Segundo Ivar Jacobson, se nós tentarmos descrever um caso de uso que contenha uma

grande quantidade de variações de eventos, nosso texto pode ficar difícil de ser entendido (JACOBSON, 1995).

‘Cenários explosivos’ devem ser evitados (BERARD, 1997). Escrever muitas páginas de descrição torna difícil controlar o que está acontecendo. Cenários explosivos podem ser evitados utilizando-se as técnicas de inclusão de casos de uso, extensão e variação. As ações de um caso de uso incluído são acionadas quando ele é nomeado em um passo de um caso de uso. Comportamentos alternativos e de falhas em um passo são gerenciados por uma extensão de cenário.

Muitas técnicas de organização podem ser utilizadas para se combinar cenários principais e extensões. Escrever casos de uso de extensão e inclusão é mais efetivo do que escrever fluxos alternativos (COCKBURN, 2000; CHEESMAN & DANIELS, 2000). Esta maneira, esta de acordo com os princípios de uma boa análise e desenho orientado a objeto (RICHTER, 2001).

De acordo com Alistair Cockburn, os melhores casos de uso têm de 3 a 8 passos, provavelmente porque as pessoas não pensam em termos de processos que tenham mais de 10 passos intermediários (COCKBURN, 2000).

3.1.8 Contando Extensões e Inclusões de Casos de Uso

Apesar de Karner recomendar que inclusões e extensões de casos de uso não devam ser contadas, a funcionalidade descrita nestes casos de uso deve ser implementada (ANDA *et al.*, 2001). Se estes casos de uso contêm funcionalidades adicionais, é necessário incluí-las na contagem. Bente ANDA *et al.* deixa claro que não se deve seguir sempre a recomendação de Karner. Segundo ainda Bente ANDA *et al.* em alguns projetos, extensões e inclusões de casos de uso contêm importantes funcionalidades.

3.2. Incerteza na Medida

A incerteza de uma medição não é outra coisa senão a sua qualidade (FIDÉLIS).

Importante notar que o resultado de uma medição só poderá ser considerado “completo” se a incerteza da medição estiver informada (FIDÉLIS). No nosso caso não é suficiente se informar se um determinado caso de uso é de alta, média ou baixa complexidade, ou ainda se a experiência da equipe com orientação a objetos é pouca, suficiente ou grande, pois todas estas informações contêm um alto nível de insegurança.

Se dissermos que o projeto apresenta poucas ou algumas características especiais de segurança, qual o nível de certeza que temos nesta informação?

Possivelmente a única que temos é que o nível de segurança deve estar mais fortemente situado entre estas fronteiras.

Resta claro, que todos os parâmetros informados na métrica de pontos de Caso de Uso, estabelecida por Karner, guardam incertezas que, como tal, representam riscos na mensuração dos projetos, cuja expectativa de tempo de desenvolvimento desejamos mensurar.

3.2.1 Como é avaliada a Incerteza

Dois valores são necessários para caracterizar uma incerteza. Um é o intervalo. O outro é o nível de confiança ou probabilidade de abrangência, que estabelece o quão seguro estamos de que o “valor verdadeiro” está neste intervalo (FIDÉLIS).

Por ser uma consequência de efeitos aleatórios nem sempre conhecidos ou facilmente mensuráveis, a incerteza é normalmente expressa como uma quantidade, ou seja um intervalo em torno do resultado.

Os resultados de medições repetidas são distribuídos em torno da média, conforme a distribuição de probabilidade escolhida para representar o fenômeno estudado.

No presente trabalho levamos em conta a incerteza técnica, dada pelos diversos fatores analisados pela métrica de pontos de caso de uso (UPC, sigla em inglês), tais como: habilidade da equipe, jornada de trabalho, características intrínsecas do projeto (segurança, manutenibilidade, portabilidade, solução de terceiros etc), conhecimento dos usuários, dentre outros, para minimizar a incerteza econômica representada pela quantidade de horas calculada pela UPC e minimizar os impactos financeiros negativos que ocorrem, quando a empresa perde porque calculou o valor do projeto inferior aos seus custos de desenvolvimento ou porque o cliente não fecha o contrato por achar o projeto muito caro.

3.2.2 Gestão de Riscos

A gestão de riscos abrange todos os processos envolvidos na identificação, avaliação e julgamento dos riscos, a atribuição de responsabilidade, tomando ações para mitigar ou antecipá-los, e o monitoramento e avaliação dos progressos realizados (THE RISK MANAGEMENT GUIDE, 2011). O risco é um fator importante a ser considerado durante a gestão de um projeto.

Risco pode ser definido como incerteza do resultado (se oportunidade positiva ou ameaça negativa). Assumir riscos é inevitável se o projeto pretende atingir os seus objetivos (THE RISK MANAGEMENT GUIDE, 2011). A gestão de risco é necessária para se administrar a exposição de um projeto aos riscos, isto

é, a probabilidade de riscos específicos que ocorrem e o impacto potencial se não ocorreu.

A gestão do risco não é um processo linear, e sim, o equilíbrio de uma série de elementos entrelaçados, que interagem uns com os outros e que devem estar em equilíbrio com o todo.

A gestão de risco, ao nível de qualquer projeto se concentra em manter resultados indesejados a um mínimo aceitável. Decisões sobre o gerenciamento de risco a este nível constituem uma parte importante do processo de negócios (THE RISK MANAGEMENT GUIDE, 2011).

O objetivo é gerenciar a exposição, tomando medidas para mantê-la a um nível aceitável. Gerir riscos envolve ter:

- Acesso a informação confiável e atualizada sobre os riscos; Processos de decisão apoiada por um quadro de análise de risco;
- Processos para monitorar os riscos; e
- O equilíbrio certo de controle para lidar com esses riscos.(THE RISK MANAGEMENT GUIDE, 2011)

Risco, em geral, pode ser definido como a probabilidade de ocorrência de acontecimentos desfavoráveis (THE RISK MANAGEMENT GUIDE, 2011).

Risco sistemático ou de mercado é um tipo de risco que afeta todas as empresas em geral e não é passível de ser reduzido pela diversificação. Diz respeito à incerteza da inflação, da política monetária e orçamental, às mudanças conjunturais (THE RISK MANAGEMENT GUIDE, 2011).

No caso dos projetos de desenvolvimento de software, os mesmos podem acabar acarretando aumento de custos que por sua vez podem eliminar o lucro, ou mesmo gerar imensos prejuízos financeiros.

No nosso caso, prever o tamanho de um sistema a ser desenvolvido, envolve analisar um conjunto de variáveis que contêm uma carga de risco muito elevada, pois envolve, muitas vezes, a experiência profissional (maior ou menor) de quem está levantando os dados que servirão de base, para a utilização da métrica de Pontos de Caso de Uso, tais como:

Definição dos atores envolvidos no uso do sistema final, onde o profissional precisa definir se os mesmos são:

- Atores simples - são aqueles que caem nas categorias "sistemas externos." Um exemplo seria a interface a um sistema de legado. Estes tipos de atores têm uma interface bem-definida e são totalmente previsíveis.
- Atores médios - são aqueles que caem na categoria "hardware devices". Um exemplo seriam os leitores de código de barra ou o

timers. Embora estes atores sejam previsíveis, requerem mais esforço em controlá-los e estão normalmente mais inclinados ao erro.

- Atores complexos - são aqueles que caem na categoria "humano". Um exemplo seria Balconista de Serviço, o Fornecedor, e o Freguês. Estes tipos de atores são bem difíceis de controlar e são totalmente imprevisíveis. Embora um GUI ou interface regular tenha consistências e imponha edições e controles, há mais complexidade em trabalhar com um ator desconhecido.

Definição da complexidade de cada um dos casos de uso levantados, onde o profissional precisa definir se os mesmos são:

- Simples - São processos simples de negócios.
- Médio - são algoritmos matemáticos ou processos que exigem algum processamento específico.
- Complexo - são processos em real time, sincronização, etc.

Definição dos fatores técnicos envolvidos no desenvolvimento do projeto, onde o profissional deverá definir as seguintes características:

O sistema irá usar componentes distribuídos em diversos servidores ?

- não usará
- irá usar, mas geograficamente estão próximos e não irá influenciar na aplicação.
- irá usar e isso será levado em consideração no momento do projeto.
- irá usar e ao desenvolver, o cuidado deverá ser dobrado.

Qual a necessidade de preocupar-se com boas respostas de performance nesta aplicação ?

- a aplicação é bastante simples e estática.
- a aplicação é simples, mas não há nada no desenvolvimento que possa provocar impacto na sua performance.
- a aplicação tem um compromisso bastante grande com os tempos de respostas das transações
- a aplicação é um sistema de missão crítica e é de fundamental importância um bom nível de performance.

A aplicação terá muitos serviços on-line ?

- nenhum serviço on-line.
- poucos serviços on-line.
- muitos serviços on-line.
- a aplicação é totalmente on-line.

Qual a complexidade dos processamentos internos?

- a aplicação é bastante simples e não há processos complexos.
- a aplicação é simples, mas existem alguns processamentos complexos, como cálculos, algoritmos inteligentes, etc.

- existem vários processos complexos que poderão exigir performance de máquina, como sincronização, atualização online de conteúdo, classes dinâmicas, etc.
- a aplicação é toda complexa e irá exigir uma dedicação especial de máquina e acompanhamento de um líder técnico na criação de cada um destes processos.

Quanto dos componentes desta aplicação poderão ser reutilizados?

- nenhum componente.
- 10 a 20% dos componentes.
- 21 a 40% dos componentes.
- mais que 40% dos componentes.

Qual a complexidade de instalação desta aplicação?

- praticamente nenhuma, será somente publicada em um webserver e o usuário poderá acessar de qualquer browser.
- fácil, embora o usuário acesse de um browser qualquer, será necessário ter instalado na máquina cliente alguns plugins e/ou dll's especiais.
- o usuário deverá instalar a aplicação na sua máquina, mas o script de instalação irá encarregar-se de configurar o equipamento.
- o usuário terá que instalar e configurar alguns itens do seu equipamento.

Ao desenvolver a aplicação, o desenvolvedor deverá preocupar-se com a usabilidade desta?

- não.
- a preocupação deverá existir, mas o negócio não é complexo, portanto qualquer que seja a lógica apresentada, a aplicação será intuitiva.
- o desenvolvedor deverá preocupar-se, por o negócio é complexo e não intuitivo.
- a usabilidade da aplicação é um dos fatores mais importantes deste projeto, uma vez que a produtividade do usuário final está diretamente ligada à aplicação. Neste caso deverá ser feito até estudos de ergonomia.

A aplicação irá "rodar" em diversos ambientes?

- não. A plataforma será pré-definida.
- Existem 2 a 3 possibilidades de plataformas.
- Não há uma plataforma definida, mas não é uma exigência do cliente que a aplicação funcione nas plataformas mais antigas.
- Existem diversas plataformas de sistemas operacionais, browsers, equipamentos e até conflitos com outras aplicações deverão ser consideradas.

Qual a preocupação com a manutenibilidade desta aplicação?

- a aplicação não sofrerá manutenção constante, mas as boas práticas de desenvolvimento de código serão apreciadas.
- a aplicação não sofrerá manutenção constante, mas é bastante importante seguir um padrão de nomes de variáveis, documentação no código, etc. A documentação atualizada também deverá ter atenção especial.
- a documentação do projeto deverá estar sempre atualizado, pois esta aplicação deverá sofrer manutenção constante, uma vez que já estão previstas novas fases de desenvolvimento com manutenções evolutivas.
- a aplicação é um sistema de missão crítica e a agilidade na manutenção é fator muito importante.

A aplicação terá muitos usuários simultâneos ?

- não. Esta é uma aplicação *stand alone*
- poucos usuários simultâneos.
- muitos usuários simultâneos.
- muitos usuários simultâneos, e isto deverá ser levado em conta no momento da arquitetura lógica e física. É recomendado também, executar testes de carga antes de homologar a aplicação.

Existem características especiais de segurança nesta aplicação ?

- nenhuma.
- poucas. Os requisitos de segurança são comuns.
- algumas. A aplicação irá exigir alguns procedimentos especiais para garantir a segurança.
- Esta é uma aplicação de missão crítica e a segurança é fator fundamental.

O projeto envolve solução de terceiros ?

- não
- sim, será utilizada soluções, mas não oferece grandes riscos para o projeto.
- sim, será utilizada solução de terceiros e esta deverá ser mantida para a sua aplicação.
- sim, serão utilizadas mais de uma solução.

Requer treinamento especial na solução ?

- não. A arquitetura, solução e negócio são de total domínio.
- sim. A aplicação irá utilizar recursos novos.
- sim. Além de utilizar novos recursos, o negócio também não é de domínio.
- sim. Recomenda-se acompanhamento de um consultor especializado.

- Definição de aspectos, relativos à equipe que irá atuar no desenvolvimento do projeto, onde o profissional precisa definir se:

A equipe tem Familiaridade com o RUP?

- não.
- parte de um processo.
- sim. Existe um processo que prevê e orienta todas as fases do desenvolvimento.
- o processo utilizado é inclusive inspecionado, garantindo que nenhuma fase importante seja ignorada.

A equipe tem experiência na aplicação em desenvolvimento?

- não.
- parte da equipe não tem experiência, mas terá o suporte de um líder técnico.
- poucos não tem experiência, mas não representará nenhuma dificuldade.
- é de domínio de toda a equipe.

A equipe tem experiência em Orientação a Objetos?

- não.
- parte da equipe não tem experiência, mas terá o suporte de um líder técnico.
- poucos não tem experiência, mas não representará nenhuma dificuldade.
- é de domínio de toda a equipe.

A equipe conta com a presença de um Analista experiente (liderança)?

- não.
- existem problemas com parte da equipe, como por exemplo a distância geográfica.
- não existem problemas do tipo.
- a equipe funciona como uma "engrenagem".

A equipe está motivada?

- não.
- a motivação é uma preocupação na equipe.
- alguns membros da equipe estão motivados e outros não.
- sim, está motivada.

A equipe está preparada para o impacto caso haja mudança de requisitos (requisitos estáveis)?

- não. Problemas desta natureza sempre provocam desgastes e retrabalho.
- gestão de mudanças é uma preocupação da equipe, e é por este motivo que existe um esforço maior na fase de levantamento.
- mudanças de requisitos nunca foi um problema, pois a equipe está treinada e preparada para gerenciar mudanças

- O desenvolvimento é feito em espiral, prevendo vários releases durante o projeto. Desta forma, as mudanças não criam impacto nenhum no processo.

A equipe será alocada em meio-período, compartilhando a atenção com outros projetos (desenvolvedores em meio expediente)?

- não.
- uma pequena parte da equipe.
- grande parte da equipe.
- toda a equipe.

A equipe tem dificuldade com a linguagem de programação?

- não. Toda a equipe é formada por profissionais seniores.
- grande parte da equipe é formada por profissionais seniores.
- alguns componentes da equipe estão em treinamento (estagiários)
- a maioria da equipe é formada por estagiários.

Como podemos constatar, o profissional precisará contar com uma experiência muito grande ou com uma equipe de especialistas para poder definir com a maior precisão possível o ambiente em que o desenvolvimento do projeto se dará.

De qualquer maneira, os riscos inerentes ao mau dimensionamento destas respostas poderão significar um maior ou menor risco, no esforço necessário, ao efetivo desenvolvimento do sistema, objeto da métrica de Pontos de Casos de Uso. O que pode resultar, ao invés de lucro, em prejuízo para a empresa desenvolvedora.

3.2.3 Gerenciamento de Riscos em Projetos

Por definição, Projetos são basicamente processos únicos com limitação de escopo, tempo e recursos. O fato de os Projetos serem únicos e de possuírem limitações exige do Gerenciamento de Projetos um planejamento consistente em um ambiente de incertezas, ou seja, em um ambiente probabilístico (Fernandes, 2005).

Nos tempos atuais da Tecnologia da Informação, cabe aos gerentes de projetos, engenheiros de softwares, arquitetos de sistemas ou aos analistas de sistemas a difícil missão de gerenciar os riscos associados ao desenvolvimento dos projetos de software. Para tanto, devem utilizar os seguintes processos, conforme o *Project Management Body of Knowledge* (PROJECT MANAGEMENT GUIDE, 2004):

- Planejamento do gerenciamento de riscos.
- Identificação de riscos.
- Análise qualitativa de riscos.
- Análise quantitativa de riscos.

- Planejamento de respostas a riscos,
- Monitoração e controle de riscos.

É na análise quantitativa em particular que se tenta traduzir em termos efetivamente numéricos – e assim melhor avaliar – os riscos relacionados a um Projeto. Para tal são usadas técnicas de simulação, ou seja, são geradas amostras aleatórias segundo um modelo da probabilidade de um risco para o Projeto em termos de tempo ou custo (FERNANDES, 2005).

Neste trabalho, optamos por utilizar o método Monte Carlo para simular, de forma aleatória, os diversos parâmetros, relacionados à métrica de Pontos de Caso de Uso, visando minimizar o risco associado ao tempo para o desenvolvimento de projetos de software.

3.2.4 Método Monte Carlo (MC)

O interesse nos projetos de software é determinar o tempo de desenvolvimento e, por contingência, o seu custo. Devido às incertezas nas variáveis, estas são consideradas como variáveis aleatórias. Assim define-se uma distribuição e probabilidade para representar o comportamento dessas variáveis. Neste caso, o método da simulação Monte Carlo permite obter amostras ou cenários de forma aleatória das distribuições de probabilidade que representam o comportamento de cada variável.

Cada amostra corresponde a uma iteração do método. Desse modo, o método de MC fornece uma estimativa do valor de um tempo ou custo esperado assim como um erro para esta estimativa, o qual é inversamente proporcional ao número de iterações ou amostras da variável aleatória. O erro total é dado por:

$$\varepsilon = \frac{3\delta}{\sqrt{N}}$$

onde δ é o desvio padrão da variável aleatória e N é o número de iterações. Ou seja, quanto maior o número de iterações, menor será o erro. Em outras palavras, a essência da simulação MC é: Estabelecer uma distribuição de probabilidade (modelo) à qual responde uma variável aleatória (tempo ou custo) para o risco analisado (FERNANDES, 2005).

Neste caso, as variáveis aleatórias são múltiplas, pois múltiplas são as variáveis com incerteza identificáveis pela métrica em um dado projeto. Vale à pena destacar também, que as variáveis aleatórias são independentes e a sua correlação é não-linear. Em resumo, significa que os riscos simulados são independentes e que um não influencia no outro ou esta influência é mínima.

2.3.1.1 Teorema do Limite Central

O somatório das variáveis aleatórias também só é possível devido ao Teorema do Limite Central, que diz: sob condições gerais, a função de distribuição acumulada (FDA) de uma soma de variáveis aleatórias independentes aproxima-se à FDA de uma variável aleatória gaussiana apesar da FDA das variáveis aleatórias individuais poderem estar longe de serem gaussianas. Ou seja, pouco importa a distribuição de probabilidade de cada variável aleatória independente correspondente a cada risco analisado, o somatório delas resulta sempre em uma distribuição normal (isso para um número considerável de variáveis aleatórias). Isto explica por que as distribuições normais aparecem com frequência na prática (FERNANDES, 2005).

A lição do Teorema do Limite Central é, portanto, que o resultado da simulação MC multivariada responde a uma curva normal de distribuição de probabilidade (CNDP) e FDA, sob pena de tornar a simulação sem efeito caso contrário (FERNANDES, 2005).

2.3.1.2. Distribuição Triangular

Outro importante detalhe a ser levado em conta para a qualidade dos resultados obtidos com a simulação é a escolha do modelo, ou seja, da melhor distribuição de probabilidade para cada risco analisado (FERNANDES, 2005).

Como cada projeto de desenvolvimento de software guarda características que diferem um dos outros, resolvemos usar a distribuição triangular ou Beta-Pert para representar a probabilidade das variáveis com incerteza, previstas na métrica de Pontos de Caso de Uso.

As figura 1 e 2 apresentam uma distribuição triangular com as suas características:

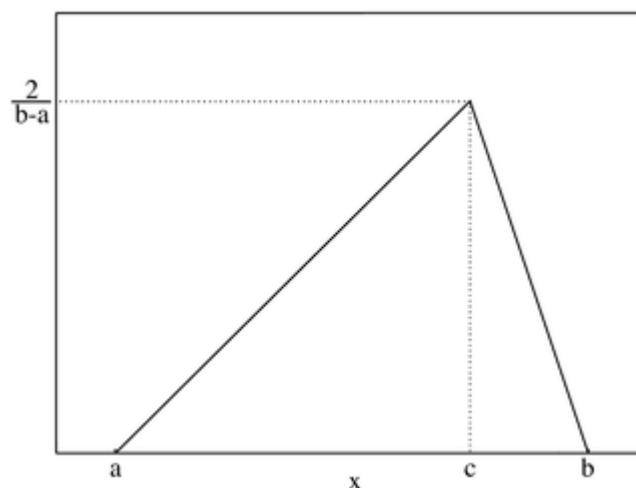


Figura 1 Distribuição Triangular - Distribuição Contínua

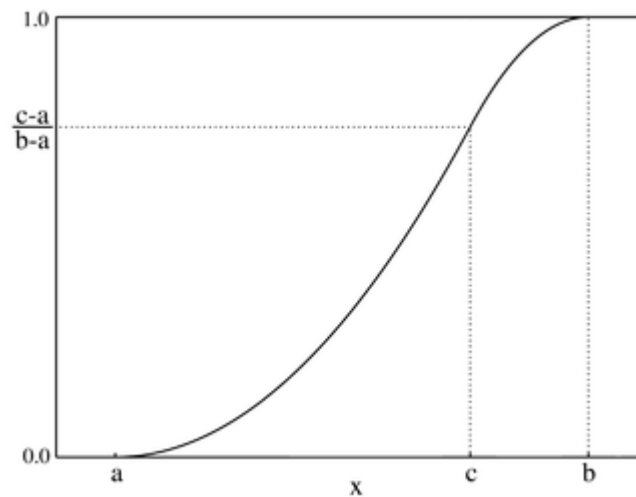


Figura 2 Distribuição Triangular - Função de Densidade de Probabilidade

A distribuição triangular é uma distribuição contínua definida no limite $x \in [a, b]$ com a função de densidade de probabilidade

$$P(x) = \begin{cases} \frac{2(x-a)}{(b-a)(c-a)} & \text{for } a \leq x \leq c \\ \frac{2(b-x)}{(b-a)(b-c)} & \text{for } c < x \leq b \end{cases}$$

e a função de distribuição

$$D(x) = \begin{cases} \frac{(x-a)^2}{(b-a)(c-a)} & \text{for } a \leq x \leq c \\ 1 - \frac{(b-x)^2}{(b-a)(b-c)} & \text{for } c < x \leq b, \end{cases}$$

onde $c \in [a, b]$ é a moda (WEISSTEIN).

A média é

$$\mu = \frac{1}{3}(a + b + c),$$

A distribuição triangular utiliza 3 parâmetros: um valor mínimo, outro mais provável e um terceiro máximo. A figura 3 mostra o emprego da distribuição triangular para a simulação de um risco associado, à variável que define o nível de segurança do sistema pretendido e que pode variar entre:

- 0 - nenhuma.
- 1 a 2 - poucas. Os requisitos de segurança são comuns.
- 3 a 4 - algumas. A aplicação irá exigir alguns procedimentos especiais para garantir a segurança.
- 5 - Esta é uma aplicação de missão crítica e a segurança é fator fundamental.

Na figura 3 a distribuição variou entre o mínimo de 0 e o máximo de 5, sendo que o especialista determinou inicialmente que o valor mais provável era 1:

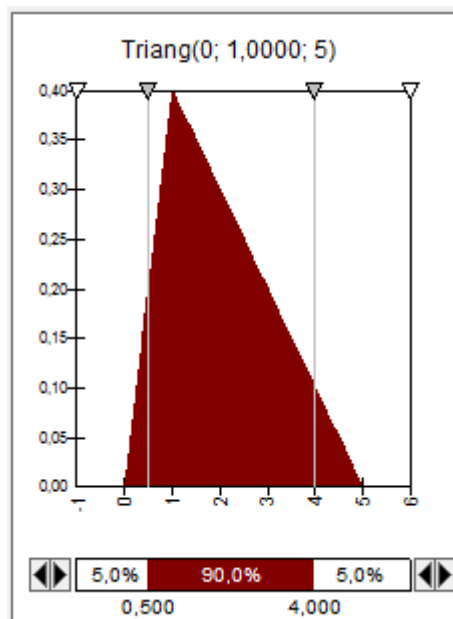


Figura 3 Distribuição triangular para a variável Segurança.

No presente trabalho, utilizamos a simulação Monte Carlo com 10.000 iterações para cada um dos parâmetros atribuíveis a métrica de Pontos de Caso de Uso, conforme descritos no item 2.2, todos eles utilizando-se de uma distribuição triangular. A cada iteração é obtido o somatório destas variáveis e ao final destas 10.000 iterações é utilizado a média dos valores contabilizados pela variável Esforço Total, como a provável quantidade de horas a ser utilizada no desenvolvimento do projeto em questão. Para tanto, foram utilizados os programas @Risk, da empresa Palisade Corporation, em associação com o programa Excel da Microsoft.

4. Arquitetura do Sistema Proposto

Para que pudéssemos avaliar as diferenças calculadas com a utilização da métrica de Pontos de Caso de Uso com e sem incerteza, desenvolvemos duas planilhas Excel, sendo a primeira utilizando a métrica básica proposta por Karner e a segunda incorporando a simulação Monte Carlo na avaliação de incertezas das medições, através de uma distribuição de probabilidade contínua triangular, utilizando para tanto o software @Risk versão 4.5.7.

Na planilha com a métrica descrita por Karner, definimos os Atores, os Fatores Técnicos e os Fatores Ambientais, conforme descrito nos Anexos 1, 2 e 3, respectivamente.

Na planilha com a métrica, acrescida da incerteza, estes valores foram recalculados 10.000 vezes utilizando-se a Simulação Monte Carlo, com a semente escolhida randomicamente, via @Risk, conforme parametrização apresentada nas figuras 4 e 5, abaixo:

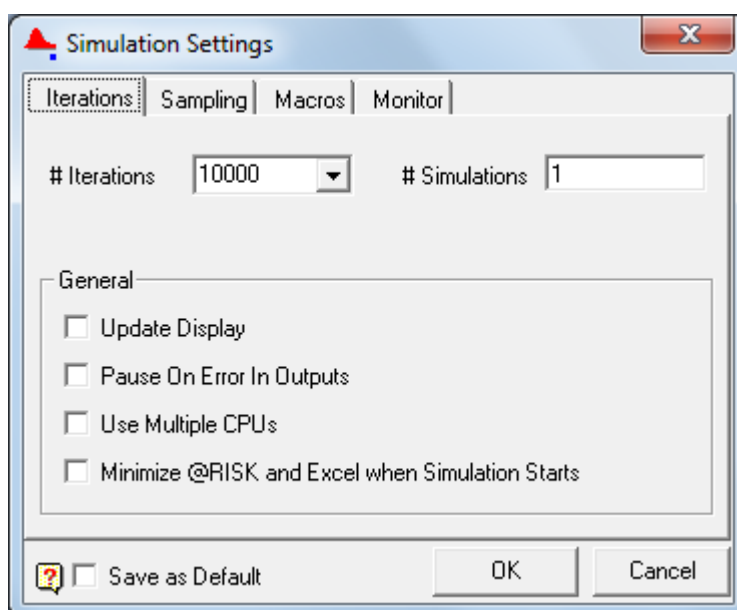


Figura 4 Tela do programa @Risk, configurada para 10.000 simulações

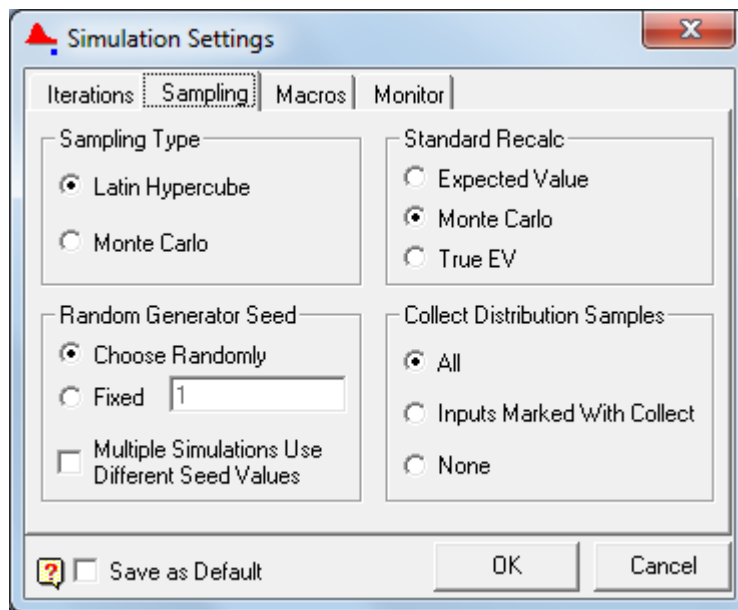


Figura 5 Tela do programa @Risk configurada para simulação Monte Carlo

Para que pudéssemos avaliar as diferentes previsões de esforço no desenvolvimento de sistemas, consideramos um conjunto de variações de níveis de complexidade, nos casos de uso avaliados. Desta forma, poderemos comparar o grau de acréscimo ou decréscimo, no conjunto de horas dimensionadas pela métrica tradicional, proposta por Karner, da métrica com o cálculo da incerteza na medição.

Para tanto, foram gerados vinte e um diferentes cenários hipotéticos de sistemas a serem desenvolvidos, onde o número de casos de uso e suas complexidades apresentam variações entre si e cujo resumo pode ser visto na tabela 10:

Tabela 10: 21 cenários de sistemas a serem desenvolvidos

Sistema	Caso de Uso Simples	Caso de Uso Médio	Caso de Uso Complexo
1	1		
2		1	
3			1
4	1	1	
5	1		1
6		1	1
7	1	1	1
8	2		
9		2	
10			2
11	2	2	
12	2		2
13		2	2
14	2	2	2

15	3		
16		3	
17			3
18	3	3	
19	3		3
20		3	3
21	3	3	3

Na planilha que agrega o cálculo da incerteza, na métrica de Karner, para a previsão do esforço de desenvolvimento de um determinado projeto, foram utilizadas distribuições triangulares, conforme abaixo descritas.

Na avaliação dos Atores e Casos de Uso, foi utilizada uma distribuição triangular com o valor mínimo de 1, o máximo de 2 e o valor provável gerado aleatoriamente, através da simulação Monte Carlo, conforme figura 6:

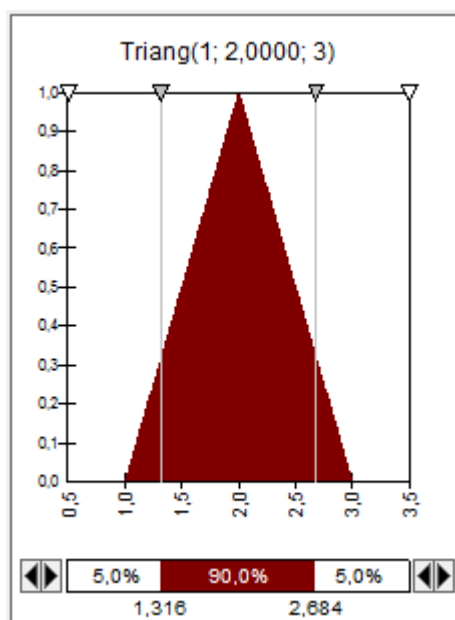


Figura 6 Distribuição triangular com mínimo de 1 e máximo de 3

Na avaliação dos fatores Técnicos e Fatores Ambientais foi utilizada uma distribuição triangular com o valor mínimo de 0, o máximo de 5 e o valor provável gerado aleatoriamente, através da simulação Monte Carlo, conforme apresentado na figura 7:

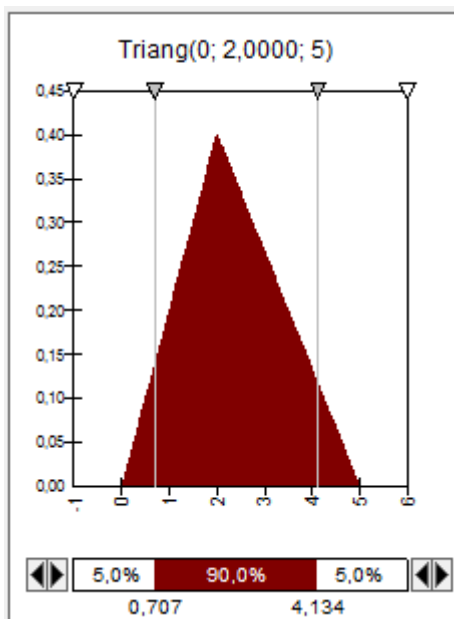


Figura 7 Distribuição triangular com mínimo de 0 e máximo de 5

A planilha Excel desenvolvida, foi composta de 5 abas, assim definidas:

1) Atores:

Tabela 11: Aba Atores

Identificação		Tipo com Incerteza			Tipo Informado			Coluna	Função
D	Atores	Simplex	Médio	Complexo	Simplex	Médio	Complexo		
	Administrador		1				1	3	2
	Usuário Normal		1				1	3	2
	Web Service	1			1			1	1
	Sistema para leitura de código de barras		1			1		2	2
	Total	1	3	0	1	1	2		

Definição das colunas:

- 1) Atores – recebe o nome dos Atores definidos para o projeto.
- 2) Tipo com Incerteza – recebe em uma das colunas, indicadas por Simplex, Médio e Complexo, o resultado calculado na coluna Função. Ou seja, a classificação do tipo do ator pode variar dependendo da coluna calculada pela distribuição triangular, gerada na coluna Função.
- 3) Tipo informado – permite que o especialista informe a qualidade do Ator que está sendo explicitada, segundo a métrica definida por Karner, descrita no Anexo 1.

- 4) Coluna – Contém o número da coluna de 1 a 3 (1=Simple, 2=Médio e 3=Complexo), conforme informado pelo especialista no Tipo Informado e é o valor provável (segundo valor, na distribuição triangular), calculado na célula Função.
- 5) Função – guarda a distribuição triangular a ser calculada pelo @Risk (=ARRED(RiskTriang(1;Valor provável; 3);0)).
- 6) Total – guarda o somatório das colunas contidas em Tipo com Incerteza e Tipo Informado. Servirá de entrada, na Aba Parâmetros para se calcular o Peso Total Não Ajustado (PTNA), através da fórmula, prevista na métrica de Karner como:

$$PTNA = (\text{num. de casos simples} + (\text{num. de casos médios} * 2) + (\text{num. de casos complexos} * 3))$$

2) Casos de Uso:

Tabela 12: Aba Casos de Uso

Identificação	Tipo Com Incerteza			Tipo Informado			Coluna	Função
	Casos de Uso	Simple	Médio	Complexo	Simple	Médio		
Caso de uso simples		1			1		1	2
Caso de uso médio		1				1	2	2
Caso de uso complexo		1				1	3	2
Total	0	4	0	1	2	1		

Definição das colunas:

- 1) Casos de Uso – recebe a descrição dos nomes dos casos de Uso definidos para o projeto.
- 2) Tipo com Incerteza – recebe em uma das colunas, indicadas por Simple, Médio e Complexo, o resultado calculado na coluna Função. Ou seja, a classificação do tipo de caso de uso pode variar dependendo da coluna calculada pela distribuição triangular, gerada na coluna Função.
- 3) Tipo informado – permite que o especialista informe a característica do caso de uso que está sendo explicitada, segundo a métrica definida por Karner.
- 4) Coluna – Contém o número da coluna de 1 a 3 (1=Simple, 2=Médio e 3=Complexo), conforme informado pelo especialista, no Tipo Informado e é o valor provável (segundo valor, na distribuição triangular), calculado na célula Função.

- 5) Função – guarda a distribuição triangular a ser calculada pelo @Risk (=ARRED(RiskTriang(1;Valor provável; 3);0)).
- 6) Total – guarda o somatório das colunas contidas em Tipo com Incerteza e Tipo Informado. Servirá de entrada na Aba Parâmetros para se calcular os Casos de Uso Não Ajustado (CUNA), através da fórmula, prevista na métrica de Karner como:
- $$\text{CUNA} = (\text{num. de casos de uso simples} * 5) + (\text{num. de casos de uso médio} * 10) + (\text{num. de casos de uso complexos} * 15)$$

3) Fatores Técnicos:

Tabela 13: Aba Fatores Técnicos

Fatores Técnicos	Pontuação (0 - 5)	Base para Cálculo	Atribuído	Função
Sistemas Distribuídos	1	2	0	1
Performance	0	0	4	0
Eficiência Usuário Final(serviços online)	2	2	3	2
Complexidade dos Processamentos Internos	2	2	1	2
Reusabilidade	2	2	5	2
Complexidade de Instalação	0	0	1	0
Usabilidade	3	1,5	2	3
Portabilidade	1	2	0	1
Manutenibilidade	1	1	5	1
Usuários Simultâneos	1	1	2	1
Características Especiais de Segurança	3	3	3	3
Solução de Terceiros	2	2	0	2
Treinamento Especial	2	2	0	2
	Total	20,5		

Definição das colunas:

- 1) Fatores Técnicos – recebe a descrição de cada um dos Fatores Técnicos, definidos pela métrica de Pontos de Caso de Uso.
- 2) Pontuação – recebe a pontuação calculada na coluna Função, que varia de 0 a 5. Ou seja, a pontuação atribuída a cada um dos Fatores Técnicos pode variar, dependendo da coluna calculada pela distribuição triangular, gerada na coluna Função.

- 3) Base para Cálculo – guarda o valor obtido pela multiplicação da pontuação encontrada na coluna Pontuação vezes o peso relativo do Fator Técnico em questão, conforme a tabela 13.
- 4) Atribuído – permite que o especialista atribua um valor ao Fator Técnico, conforme a sua experiência e segundo a métrica definida por Karner, mostrada no Anexo 2 e é o valor provável (segundo valor, na distribuição triangular).
- 5) Função – guarda a distribuição triangular a ser calculada pelo @Risk (=ARRED(RiskTriang(0;Valor provável; 5);0)).
- 6) Total – guarda o somatório das colunas da coluna Base para Cálculo. Servirá de entrada na Aba Parâmetros para se calcular o fator de Complexidade Técnica, através da fórmula, prevista na métrica de Karner como:

$$FCT = 0.6 + (0.01 * (\text{valor da coluna Total}))$$

4) Fatores Ambientais:

Tabela 14: Aba Fatores Ambientais

Fatores Ambientais	Pontuação (0 - 5)	Base para Cálculo	Atribuído	Função
Familiaridade com o RUP	2	3	2	2
Experiência com a Aplicação em desenvolvimento	3	1,5	4	3
Experiência em Orientação a Objetos	3	3	5	3
Presença de analista experiente	3	1,5	2	3
Motivação	5	10	5	5
Requisitos estáveis	3	-3	5	3
Desenvolvedores em meio expediente	4	4	5	4
Linguagem de programação difícil	0	0	0	0
Total		20		

Definição das colunas:

2. Fatores Ambientais – recebe a descrição de cada um dos Fatores Ambientais, definidos pela métrica de Pontos de Caso de Uso.
3. Pontuação – recebe a pontuação calculada na coluna Função que varia de 0 a 5. Ou seja, a pontuação atribuída a cada um dos Fatores Ambientais pode variar dependendo da coluna calculada pela distribuição triangular, gerada na coluna Função.

4. Base para Cálculo – guarda o valor obtido pela multiplicação da pontuação encontrada na coluna Pontuação vezes o peso relativo do Fator Técnico em questão, conforme a tabela 14.
5. Atribuído – permite que o especialista atribua um valor ao Fator Técnico, conforme a sua experiência, e segundo a métrica definida por Karner, mostrada no Anexo 3, e é o valor provável (segundo valor, na distribuição triangular).
6. Função – guarda a distribuição triangular a ser calculada pelo @Risk (=ARRED(RiskTriang(0;Valor provável; 5);0)).
7. Total – guarda o somatório das colunas da coluna Base para Cálculo. Servirá de entrada na Aba Parâmetros para se calcular o fator de Complexidade Técnica, através da fórmula, prevista na métrica de Karner como:

$$FA = 1.4 + (-0.03 * (\text{valor da coluna Total}))$$

5) Parâmetros:

Tabela 15: Aba Parâmetros

PCU	52,3952
Produtividade	10:00:00
Esforço Total em Horas	523:57:07

Definição das Linhas:

- 1) PCU – A fórmula para cálculo dos Pontos de Caso de Uso (PCU) ajustados é o seguinte:

$$PCUNA = (PTNA + CUNA)$$

$$PCU = PCUNA * FT * FA$$
- 2) Produtividade - Karner propôs um fator de 20 horas por ponto de caso de uso para uma estimativa por projeto. Segundo alguns autores, experiências de campo demonstraram que este valor varia de 15 a 30 horas por ponto de caso de uso. Schneider e Winters sugerem um refinamento da proposta de Karner, baseada no nível de experiência da equipe e da estabilidade do projeto. Para efeitos de testes, consideramos o valor da produtividade em 10 horas.
- 3) Esforço total em horas – Quantidade de horas aproximadas para o desenvolvimento total do projeto. Obtido através da seguinte fórmula:

$$\text{Esforço total do projeto} = PCU * \text{Produtividade.}$$
 Na tabela onde foram efetuados os cálculos, levando-se em conta a incerteza, esta célula foi indicada para o @Risk como contendo a saída observada, conforme $=RiskOutput() + E16 * E17$.

5. Resultados

Como observado a partir da tabela 16, os resultados obtidos nas diversas simulações de sistemas a serem desenvolvidos, conforme apresentado nas linhas especificadas pela coluna Sistema, apresentam uma grande variação percentual (para mais ou para menos), quando se agrega o cálculo da incerteza à métrica de Pontos de Casos de Uso, original de Karner.

No caso dos valores com incerteza, a menor variação negativa foi de -12,57% e a maior de 46,19%, conforme destacado na tabela 17. Cabe ressaltar, que estas variações, se constatadas na prática, significam um término antecipado com 12,57% de possibilidade de acontecer ou em outra situação, a possibilidade do projeto demorar mais 46,19% para terminar do que o tempo previsto pela métrica proposta por Karner.

Ambas as situações demonstram o risco associado ao projeto que, se no primeiro caso caracterizaria um lucro adicional, por outro lado, a segunda situação apresenta a perspectiva de um risco de atraso (e prejuízo) de quase 50% na estimativa sem a incerteza.

Tabela 16: Tabela contendo o número de horas calculado com a métrica original de Karner e com a métrica acrescida de incerteza.

Sistema	Caso de Uso Simples	Caso de Uso Médio	Caso de Uso Complexo	Número de Horas Segundo Karner	Número de Horas Médias com Incerteza	Varição %
1	1			130	164	26,15
2		1		176	183	3,98
3			1	223	201	-9,87
4	1	1		223	264	18,39
5	1		1	269	282	4,83
6		1	1	316	301	-4,75
7	1	1	1	362	382	5,52
8	2			176	245	39,20
9		2		269	282	4,83
10			2	362	320	-11,60
11	2	2		362	445	22,93
12	2		2	455	482	5,93
13		2	2	548	519	-5,29
14	2	2	2	641	681	6,24
15	3			223	326	46,19
16		3		362	382	5,52
17			3	501	438	-12,57
18	3	3		501	625	24,75
19	3		3	641	681	6,24
20		3	3	781	737	-5,63
21	3	3	3	967	981	1,45

Tabela 17: Tabela contendo o número de horas calculado com a métrica original de Karner e com a métrica acrescida de incerteza e classificada de forma ascendente com base na variação percentual.

Sistema	Caso de Uso Simples	Caso de Uso Médio	Caso de Uso Complexo	Número de Horas Segundo Karner	Número de Horas Médias com Incerteza	Varição %
17			3	501	438	-12,57
10			2	362	320	-11,60
3			1	223	201	-9,87
20		3	3	781	737	-5,63
13		2	2	548	519	-5,29
6		1	1	316	301	-4,75
21	3	3	3	967	981	1,45
2		1		176	183	3,98
5	1		1	269	282	4,83
9		2		269	282	4,83
7	1	1	1	362	382	5,52
16		3		362	382	5,52
12	2		2	455	482	5,93
14	2	2	2	641	681	6,24
19	3		3	641	681	6,24
4	1	1		223	264	18,39
11	2	2		362	445	22,93
18	3	3		501	625	24,75
1	1			130	164	26,15
8	2			176	245	39,20
15	3			223	326	46,19

Outro ponto interessante observado se encontra apresentado na tabela 18. Nela destacamos os valores em horas dos projetos simulados, contendo os valores médios, mínimos, máximos, valores mínimos e máximos com 5% de possibilidade de acontecer. Estes valores foram obtidos a partir do Sumário de Estatísticas, listado pelo @Risk a cada conjunto de iterações (no nosso caso 10.000), conforme exemplificado na figura 8.

Tabela 18: Tabela contendo o número de horas calculado com a métrica original de Karner e com a métrica acrescida de incerteza.

Sistema	Número de Horas Segundo Karner	Número de Horas Médias com Incerteza	Varição %	Número de Horas Mínima	Número de Horas Máxima	5% de Possibilidade de terminar antes em horas	5% de possibilidade de terminar depois em horas
1	130	164	26,15	83	336	139	265
2	176	183	3,98	71	325	127	242
3	223	201	9,87	83	336	139	265
4	223	264	18,39	117	500	187	350
5	269	282	4,83	123	495	200	375
6	316	301	-4,75	124	529	221	391
7	362	382	5,52	188	662	280	493

8	176	245	39,20	119	464	167	334
9	269	282	4,83	135	506	207	365
10	362	320	-11,60	154	515	230	413
11	362	445	22,93	239	761	332	572
12	455	482	5,93	245	873	357	619
13	548	519	-5,29	248	851	394	653
14	641	681	6,24	344	1203	527	853
15	223	326	46,19	147	667	229	435
16	362	382	5,52	183	641	287	486
17	501	438	-12,57	161	727	326	558
18	501	625	24,75	299	1067	479	790
19	641	681	6,24	356	1126	517	860
20	781	737	-5,63	419	1191	573	913
21	967	981	1,45	580	1510	769	1211

Name	Worksheet	Cell	Minimum	Mean	Maximum	x1	p1	x2	p2	x2-x1	p2-p1	Errors	
Output 1	Esforço Total em Horas	Parâmetros	E18	23,98061	40,86396	64,35302	31,94583	5%	50,57515	95%	18,62931	90%	0

Figura 8 Sumário da estatística, do campo Esforço Total em Horas, gerado pelo @Risk, após 10.000 iterações

No intuito de se observar a influência do aumento do número de iterações, no cálculo da incerteza, pegou-se os sistemas de menor e maior variação percentual (12,57 e 46,19, respectivamente) e submetemos a um novo cálculo, aumentando o número de iterações de 10.000 para 100.000. Os resultados podem ser vistos na tabela 19.

Neste caso, restou comprovado que o aumento do número de iterações em nada alterou o resultado final, obtido com o conjunto de 10.000 iterações.

Tabela 19: Tabela contendo o número de horas calculado com a métrica original de Karner e com a métrica acrescida de incerteza para 10.000 e 100.000 iterações.

Sistema	Caso de Uso Simples	Caso de Uso Médio	Caso de Uso Complexo	Número de HorasSegundo Karner	Média com Incerteza 10.000 Iterações	Média com Incerteza 100.000 iterações
17			3	501	438	438
15	3			223	326	326

Finalmente, dado os resultados coletados, ficou comprovado que da mesma forma que medimos, precisamos ter em mente que qualquer medida guarda imperfeições. No caso presente, ela pode estar associada a efeitos aleatórios, tais como a capacidade dos profissionais que irão desenvolver a aplicação, a mudança do escopo do projeto no curso do seu desenvolvimento, a qualidade da liderança. Outras imperfeições estão associadas a questões inerentes ao projeto, tais como a

necessidade de alocação da equipe em meio período, a dificuldade na programação, a características especiais de segurança, aos diversos níveis de portabilidade, manutenibilidade, usabilidade e tantas outras que são requeridas na análise de pontos por caso de uso.

Desta forma, é muito importante que associemos heurísticas para cálculo de incerteza e risco na métrica de Pontos de Casos de Uso, com o objetivo de melhorar a qualidade quantitativa da análise do esforço, no desenvolvimento de projetos em ambientes que utilizem técnicas de orientação a objetos.

6. Conclusões e Trabalhos Futuros

A vantagem evidente da métrica por Casos de Uso sobre outras técnicas é que ela demonstra resultados muito mais cedo, no ciclo de desenvolvimento. Além disso, a técnica utiliza-se de um tipo de documento essencial em metodologias dirigidas por caso de uso, como o RUP, de forma que é possível calcular prontamente mudanças nas estimativas do sistema a cada pequena alteração de requisitos, apenas refazendo alguns cálculos.

Apesar dos ganhos, a técnica apresenta alguns problemas, tais como as diferenças visíveis de estimativa lançadas por analistas diferentes, baseando-se na visão e experiência pessoal de cada um, quanto ao nível de granularidade dos Casos de Uso analisados.

De uma forma ou de outra, a metodologia de medição por Pontos de Caso de Uso é uma técnica interessante para equipes que carecem de uma métrica formal de lançamento de estimativas ou que não conseguiram bons resultados utilizando-se de outros mecanismos de estimativa.

A sugestão para trabalhos futuros é a utilização de previsão por Redes Neurais, a partir do momento que as equipes consigam montar um bom histórico de projetos desenvolvidos, que tenham utilizado a métrica de Pontos de Casos de Uso.

Referências Bibliográficas

1. Anda, B.; Dreiem, H.; Jørgensen, M; Sjøberg, D; . *Estimating Software Development Effort based on Use Cases* – Experience from Industry. In International Conference on the Unified Modeling Language (UML2001), 4. Proceedings, Toronto, Canadá, Oct. 1-5, 2001, p.487-502.
2. Boehm, B.W. *Software Engineering Economics*. Prentice Hall, 1981.
3. Cheesman, J.; Daniels, J. *UML Components*, A simple Process for Specifying Component-based Software. Addison-Wesley, 2000.
4. Cockburn, A. *Structuring use cases with goals*. Journal of Object-Oriented Programming, v.10, n.5, p. 35-40, (Set. 1997) e V.10, n.7, p. 56-62, (Nov. 1997).
5. Dekkers, C.A. **Entendendo a Terminologia**. IT Metrics Strategies, 1998.
6. Cockburn, A. *Writing Effective Use Cases*. Addison-Wesley, 2000.
7. EURACHEM, *Quantifying Uncertainty in Analytical Measurement*. Laboratory of the Government Chemist, Londres (1995). (ISBN 0-948926-08-2).
8. Fenton, N.E.; Pfleeger, S.L. *Software Metrics. A Rigorous and Practical Approach*. Cambridge University Press, 1997.
9. Fernandes, E.M da G.P. **Estatística Aplicada**. Universidade do Minho, Braga, 1999. Disponível em <<http://www.norg.uminho.pt/emgpf/documentos/Aplicada.pdf>>. Acesso em 08 nov. 2011.
10. Fernandes, C.A.B.de A., **Gerenciamento de Riscos em Projetos**. Brasil. Disponível em <<http://www.bbbrothers.com.br/scripts/Artigos/MonteCarloExcel.pdf>>. Acesso em .08 nov. 2011.
11. Fetke, T; Abran, A.; Ngyen, T. (1997). *Mapping the oojacobson approach into function point analysis*. Proceedings of Tools, vol. 27. 11p. Disponível em <www.cs.unibo.it/~cianca/wwwpages/ids/lettura/Fetke.pdf>. Acesso em 10 Mar. 2005.
12. Fidélis, G.C. **Incerteza de Medição**. Centro de Educação, Consultoria e Treinamento. Disponível em <http://www.cect.com.br/Incerteza%20de%20Medicao_cect.pdf>. Acesso em 11 nov. 2011.

13. INMETRO- **Vocabulário internacional de termos fundamentais e gerais de metrologia**- 2a. edição. Brasil. 2000 (ISBN 85-87090-90-9).
14. Jacobson, Christerson, and Overgaard. **Object-oriented Software Engineering: A Use Case-driven Approach**. Addison-Wesley, 1992.
15. Jacobson, I. **The Object Advantage: Business Process Reengineering With Object Technology**. Addison-Wesley, 1995.
16. Jaynes, E.T. **Probability Theory: The Logic of Science**. Preprint: Washington University, (1996). Disponível em <<http://bayes.wustl.edu/etj/prob/book.pdf>>. Acesso em . 08 nov. 2011.
17. Karner, G. **Metrics for Objectory**. Dissertação (Mestrado em Computação e Ciência da Informação). Universidade de Linköping, Suécia, 1993.
18. Kitchenham, B.; Linkman, L.; Law, D. **A methodology for evaluating software engineering methods and tools**. IEE Computing & Control Journal, 1997, p. 120-126.
19. Longstreet, D. (1995). **Use cases and function points**. Copyright Longstreet Consulting Inc. (14 mar. 2005). Disponível em <www.softwaremetrics.com/Articles/usecases.htm>. Acesso em 08 nov. 2011.
20. MARTIN, PEDROSS, Arnold; PEDROSS, Peter. **Software Size Measurement and Productivity Rating in Large Scale Software Development** Department, IEEE, 1998
21. PROJECT MANAGEMENT INSTITUTE (PMI). **A Guide to the Project Management Body of Knowledge (PmBok)** – Third Edition, Project Management Institute, Inc., 2004).
22. Ribu, K. **Estimating Object-Oriented Software Projects with Use cases**. Dissertação (Mestrado em Informática). Universidade de Oslo, Oslo, Noruega. 2001. 147p.
23. Richter, C. **Designing Flexible Object-Oriented Systems with UML**. Macmillan Technical Publishing, 2001.
24. Rule, P.G. **Using measures to understand requirements**. Software Measurement Services Ltd, 2001.
25. Schneider & Winters. **Applying use Cases**. Addison-Wesley, 1998.
26. Smith, J. **The estimation of effort based on use cases**. Rational Software, (1999). Disponível em <<http://www.uml.org.cn/RequirementProject/pdf/finalTP171.pdf>>. Acesso em 08 nov. 2011.

27. Symons, C.R. *Software Sizing and Estimating*, MKII FPA. John Wiley and Sons, 1991.
28. Symons, C.R. (2001). *Come back function point analysis (modernised)-all is forgiven!*. Software Measurement Services Ltd, (15 Mar. 2005). Disponível em <<http://www.measuresw.com/library/Papers/Symons/fesmafpa2001/paper.html>>. Acesso em 08 nov. 2011.
29. *The Risk Management Guide - A to Z and FAQ Reference*. Disponível em <<http://www.ruleworks.co.uk/riskguide/>>. Acesso em 08 nov. 2011.
30. Weisstein, E. W, *Triangular Distribution em MathWorld*. Disponível em <<http://mathworld.wolfram.com/TriangularDistribution.html>>. Acesso em 08 nov. 2011.

Apêndice 1 – Atores

Listamos abaixo os atores que foram definidos para a utilização da métrica de pontos de Caso de Uso definidos por Karner.

Tabela 20: Descrição dos Atores

Identificação		Tipo Informado		
D	Atores	Si mples	M édio	Com plexo
	Administrador			1
	Usuario Normal			1
	Web Service	1		
	Sistema para leitura de código de barras		1	
Total		1	1	2

Onde a qualificação dos diversos atores encontra-se assinalado na coluna Tipo Informado, sendo eles:

1) Ator Administrador é considerado um ser humano com responsabilidades gerenciais, conforme Karner este usuário é considerado complexo;

2) Ator Usuário Normal é considerado um ser humano chefiado pelo Administrador e que usa as funcionalidades por ele autorizada, conforme Karner este usuário é considerado complexo;

3) Ator Web Service enquadra-se na categoria sistema externo e conforme Karner é considerado simples; e

4) Ator Sistema para leitura de Código de Barras enquadra-se na categoria de dispositivos de hardware e conforme Karner é considerado de média complexidade.

Apêndice 2 – Fatores Técnicos

Listamos abaixo os atores que foram definidos para a utilização da métrica de pontos de Caso de Uso definidos por Karner.

Tabela 21: Descrição dos Fatores Técnicos

Fatores Técnicos	Pontuação (0 - 5)
Sistemas Distribuídos	0
Performance	4
Eficiência Usuário Final (serviços online)	3
Complexidade dos Processamentos Internos	1
Reusabilidade	5
Complexidade de Instalação	1
Usabilidade	2
Portabilidade	0
Manutenibilidade	5
Usuários Simultâneos	2
Características Especiais de Segurança	3
Solução de Terceiros	0
Treinamento Especial	0

Onde a classificação dos diversos fatores encontra-se assinalado na coluna Pontuação, sendo eles:

1) Sistemas Distribuídos igual a zero ou que o sistema não irá usar componentes distribuídos em diversos servidores;

2) Performance igual a 4, ou seja a aplicação tem um compromisso bastante grande com os tempos de respostas das transações;

3) Eficiência do Usuário Final (serviços on-line) igual a 3, ou seja, a aplicação tem muitos serviços on-line.

4) Complexidade dos Processos Internos igual a 1, ou seja, a aplicação é simples, mas existem alguns processamentos complexos, como cálculos e algoritmos inteligentes;

5) Reusabilidade igual a 5, ou seja, mais de 40% dos componentes poderão ser reutilizados;

6) Complexidade de Instalação igual a 1, ou seja, fácil, embora o usuário acesse de um browser qualquer, será necessário ter instalado na máquina cliente alguns plugins e/ou dll's especiais;

7) Usabilidade igual a 2, ou seja, a preocupação do desenvolvedor quanto a usabilidade deverá existir, mas o negócio não é complexo, portanto qualquer que seja a lógica apresentada, a aplicação será intuitiva;

8) Portabilidade igual a 0, ou seja, plataforma será pré-definida;

9) Manutenibilidade igual a 5, ou seja, a aplicação é um sistema de missão crítica e a agilidade na manutenção é fator muito importante;

10) Usuários Simultâneos igual a 2, ou seja, poucos usuários simultâneos;

11) Características Especiais de Segurança igual a 3, ou seja, a aplicação irá exigir alguns procedimentos especiais para garantir a segurança;

12) Solução de Terceiros igual a 0, ou seja, a aplicação não irá utilizar nenhuma solução de terceiros; e

13) Treinamento Especial igual a 0, ou seja, a arquitetura, solução e negócio são de total domínio não requerendo treinamento especial.

Apêndice 3 – Fatores Ambientais

Listamos abaixo os fatores ambientais que foram definidos para a utilização da métrica de pontos de Caso de Uso definidos por Karner.

Tabela 22: Descrição dos Fatores Ambientais

Fatores Ambientais	Pontuação (0 - 5)
Familiaridade com o RUP	2
Experiência com a Aplicação em desenvolvimento	2
Experiência em Orientação a Objetos	1
Presença de analista experiente	2
Motivação	5
Requisitos estáveis	2
Desenvolvedores em meio expediente	4
Linguagem de programação difícil	0

Onde a classificação dos diversos fatores encontra-se assinalado na coluna Pontuação, sendo eles:

- 1) Familiaridade com o RUP igual a 1, ou seja, a equipe terá de obedecer parcialmente, durante o desenvolvimento, a um processo formal;
- 2) Experiência com a Aplicação em desenvolvimento igual a 4, ou seja, poucos não tem experiência, mas não representará nenhuma dificuldade adicional;
- 3) Experiência em Orientação a Objetos igual a 5, ou seja, é de domínio de toda a equipe;
- 4) Presença de Analista Experiente igual a 2, ou seja, existem problemas com parte da equipe, como por exemplo a distância geográfica;
- 5) Motivação igual a 5, ou seja, a equipe está motivada para o desenvolvimento do sistema e já fizeram outros projetos em conjunto;
- 6) Mudança de Escopo igual a 2, ou seja, a gestão de mudanças é uma preocupação da equipe, e é por este motivo que existe um esforço maior na fase de levantamento;
- 7) Alocação da Equipe em Meio Período igual a 5, ou seja toda a equipe trabalhará parcialmente neste projeto; e

- 8) Dificuldade na Programação igual a 0, ou seja, toda a equipe é formada por profissionais seniores sem nenhuma dificuldade de programação.