

UNIVERSIDADE FEDERAL DE PERNAMBUCO

GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

CENTRO DE INFORMÁTICA

2005.2

ANÁLISE E DEFINIÇÃO DE MÉTRICAS PARA O PROCESSO DE GERÊNCIA DE RISCOS PARA PROJETOS DE SOFTWARE

TRABALHO DE GRADUAÇÃO EM ENGENHARIA DE SOFTWARE

Aluno – Saulo Lopes da Silva Oliveira, slo@cin.ufpe.br.

Orientador – Hermano Perrelli de Moura, hermano@cin.ufpe.br.

Co-Orientadora – Cristine Martins Gomes de Gusmão, cmgg@cin.ufpe.br
(Doutoranda).

06 de fevereiro de 2006.

UNIVERSIDADE FEDERAL DE PERNAMBUCO

GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
CENTRO DE INFORMÁTICA

2005.2

SAULO LOPES DA SILVA OLIVEIRA

ANÁLISE E DEFINIÇÃO DE MÉTRICAS PARA O PROCESSO DE
GERÊNCIA DE RISCOS PARA PROJETOS DE SOFTWARE.

ESTE TRABALHO FOI APRESENTADO À GRADUAÇÃO EM CIÊNCIA DA
COMPUTAÇÃO DO CENTRO DE INFORMÁTICA DA UNIVERSIDADE
FEDERAL DE PERNAMBUCO COMO REQUISITO PARCIAL PARA OBTENÇÃO
DO GRAU DE BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

ORIENTADOR: PROF. HERMANO PERRELLI DE MOURA
CO-ORIENTADORA: CRISTINE MARTINS GOMES DE GUSMÃO

06 de Fevereiro de 2006.

Dedico,

Aos meus pais
Severino e Lenize

A minha irmã
Lorena

© Saulo Lopes da Silva Oliveira, 2006.
Todos os direitos reservados.

AGRADECIMENTOS

Aos meus pais, Severino e Lenize, minha irmã Lorena, minha namorada Sarah e todos meus familiares pelo amor, carinho, apoio, compreensão e permanente incentivo.

À aluna de doutorado Cristine Gusmão pela sua amizade, co-orientação e constante disponibilidade no decorrer deste trabalho.

Ao professor Hermano Perrelli pela orientação e validação de cada etapa deste trabalho.

A Márcio e Marcelo, integrantes do grupo de pesquisa RISCTEC, do qual faço parte, pelas sugestões dadas na elaboração deste trabalho.

E, finalmente, a todos aqueles que contribuíram direta ou indiretamente para a realização deste trabalho.

Saulo Lopes da Silva Oliveira

RESUMO

A gerência de riscos é responsável por descobrir que fatores podem levar um projeto ao fracasso, para que se possa definir ações que evitem o acontecimento desses fatores. Cada vez mais, as organizações vêm percebendo a importância da gestão de riscos. Entretanto, há ainda uma lacuna quanto a informações sobre o processo, principalmente quanto à definição de métricas. Baseado nisso, este trabalho busca tentar reduzir a falta de informações sobre métricas através da demonstração das já existentes, além da tentativa de proposição de novas métricas, a fim de tornar a forma de gerir riscos de uma organização mais eficiente.

SUMÁRIO

1. INTRODUÇÃO	9
1.1. Contexto do Trabalho	9
1.2. Motivação	11
1.3. Objetivos	13
1.3.1. Objetivos Específicos	13
1.4. Estrutura do Trabalho	14
2. ESTUDO SOBRE MÉTRICAS	16
2.1. Introdução à Métricas	16
2.1.1. Classificação das Métricas	17
2.2. Métricas de Tamanho de Software	18
2.2.1. Pontos de Função	19
2.2.2. Pontos de Casos de Uso	22
2.2.2.1. Casos de Uso	22
2.2.2.2. Cálculo de Pontos de Casos de Uso	24
2.3. Métricas em Gerência de Risco	28
2.3.1. Classificação dos Riscos	29
2.3.2. Atividades da Gerência de Risco	31
2.3.3. Métricas	34
3. PONTOS DE RISCO	36
3.1. Introdução	36
3.2. Cálculo de Pontos de Risco	37
3.3. Cálculo de Pontos de Risco Não Ajustados	37
3.4. Fatores Caracterizadores do Projeto	41
3.5. Fórmula de Ajuste	44
3.6. Aplicações	45
4. VALIDAÇÃO	47
4.1. mPrime	47
4.1.1. Principais Funcionalidades	49
4.2. Implementação da Métrica	52
5. CONCLUSÃO	55
5.1. Trabalhos Futuros	56

LISTA DE FIGURAS

Figura 1 – Exemplo de um modelo de Caso de Uso.....	23
Figura 2 – Figura Representando o Ciclo da Gestão de Riscos	34
Figura 3 – Mapeamento entre Casos de Uso e Riscos	38
Figura 4 – Visão Geral do mPrime integrado ao ms-project	48
Figura 5 – Visão do Checklist do mPrime	50
Figura 6 – Exemplo de Riscos Retornados pelos Componentes de Inteligência Artificial	51
Figura 7 – Visão de Parte da Caracterização de Projeto.....	53
Figura 8 – Visão da Exposição dos Riscos através das cores.....	53
Figura 9 – Resultado da métrica de Pontos de Risco Implementado no mPrime.....	54

LISTA DE QUADROS

Quadro 1 – Resultado da Pesquisa do Relatório CHAOS.....	10
Quadro 2 – Exemplo de Cálculo dos Pontos de Função Não Ajustados	20
Quadro 3 – Fatores Técnicos	20
Quadro 4 – Classificação dos Atores do Sistema	25
Quadro 5 – Classificação dos Casos de Uso do Sistema.....	25
Quadro 6 – Fatores de Complexidade Técnica do Sistema.....	26
Quadro 7 – Fatores de Complexidade Ambiental do Sistema.....	27
Quadro 8 – Classificação dos Riscos do Sistema	40
Quadro 9 – Exemplo de Escala de Impacto	40
Quadro 10 – Fatores Caracterizadores do Projeto.....	41

1. INTRODUÇÃO

1.1. Contexto do Trabalho

A presença de softwares na realização de uma atividade, desde a mais simples a mais complexa, se torna cada vez mais necessário, como afirma Pressman [PRESSMAN 1995]:

“Softwares se tornaram uma força fundamental. É o motor que direciona a tomada de decisões em negócios. Serve como base para o investimento científico moderno e para a solução de problemas da engenharia. Encontra-se em todas as áreas de atividade: transportes, médicos, telecomunicações, militar, processos industriais, entretenimento, produtos de escritório...”

Uma outra maneira de perceber o aumento da necessidade de softwares atualmente, é o notável crescimento da participação da indústria de software na economia global, tornando-se verdadeiramente um grande segmento. Por exemplo, estima-se que o mercado de comunicação e software movimentou em torno de 320 bilhões de euros [ANON 1998b]. Enquanto nos Estados Unidos, o setor de tecnologia da informação responde por um quarto do crescimento economia [ANON 1998a].

Entretanto, o desenvolvimento de software é bastante complexo, um dos motivos para que a indústria de software tenha dificuldades em suprir a demanda existente. Os softwares são produzidos fora do custo planejado, normalmente entregues com atraso e muitas vezes sem as funcionalidades esperadas. Estas dificuldades enfrentadas pelas organizações caracterizam a chamada “Crise do Software” (Software Crisis).

A Engenharia de Software responsável por propor melhores processos, ferramentas e técnicas para um desenvolvimento mais eficiente, é uma área de estudo bastante jovem, possui um pouco mais de quarenta anos, não havendo ainda o acúmulo necessário de informações o desenvolvimento ter atingido a maturidade. E este pode ser apontado como um dos principais motivos para a “Crise do Software”.

Para demonstrar o tamanho do impacto causado pela “Crise do Software”, pode-se citar o relatório CHAOS [STANDISH 1994], possui este nome devido ao resultado apresentado. Segundo este relatório, 31.1% dos projetos de software falham, 52.7% são modificados e apenas 16.2% são executados com sucesso.

Quadro 1 – Resultado da Pesquisa do Relatório CHAOS

Resultado do Projeto	%
Falha	31.1
Modificado	52.7
Sucesso	16.2

Portanto, é o desafio da Engenharia de Software buscar alternativas que visem solucionar estes problemas de custo, prazo e qualidade esperada de um produto.

1.2. Motivação

A Engenharia de Software por ser uma área de estudo tão jovem sofre pela falta de informações acumuladas para validação de práticas eficazes no desenvolvimento. Logo, uma das preocupações atuais é com a coleta de dados, ou em outras palavras, métricas.

Métricas atuam tanto no nível estratégico como tático da organização. É eficaz para a otimização de processos, como também auxiliam em decisões da gerência de um projeto de desenvolvimento de software. Por exemplo, podem ser utilizadas para estimar o tamanho de um projeto, ou medir a qualidade do mesmo.

Mesmo assim, poucas organizações possuem um programa de métricas, ou mesmo as que o possuem não o fazem de maneira eficaz. Isto ocorre principalmente por que um programa efetivo de métricas requer uma mudança na cultura organizacional que deve ser iniciada na alta gerência, além disso, implementar métricas é um trabalho árduo e em que os resultados apenas aparecem em longo prazo, diferente da cultura atual de desenvolvimento de software, de resultados imediatos.

Outro campo de estudo que sofre com essa cultura de desenvolvimento de software de resultados imediatos é a Gerência de

Riscos. Como se trata de uma área bastante subjetiva, normalmente acaba esquecida por não trazer nenhum aparente resultado prático de imediato. Outro problema encontrado que inibe a da gestão de riscos é que se trata de uma área de estudo bastante recente em software que além de possuir pouca literatura, suas praticas não atingiram uma maturidade a ponto de descobrir o que realmente funciona ou não.

Diferentemente, no mercado financeiro a Gestão de Riscos é amplamente usada, mais além é uma necessidade das empresas que operam neste ramo. Além disso, no mercado financeiro, riscos já atingiram um bom nível de maturidade com vasta literatura e práticas. Resumidamente, esta maturidade só foi atingida quando começou-se a mensurar o que aparentemente é totalmente subjetivo, incerto.

Portanto, este exemplo do mercado financeiro deve ser tomado como incentivo para a aplicação da gestão de riscos em projetos de software. Há uma necessidade de métricas para gestão de riscos, para que a longo prazo a taxa de falha no desenvolvimento seja cada vez menor.

Desta maneira, pode-se melhorar a tomada de decisões por parte da gerência de projetos, descobrir quais práticas realmente funcionam em gerência de risco e dessa maneira otimizar todo o processo e finalmente mensurar o benefício para a organização de aplicar uma gestão de riscos eficiente, como motivação para um contínuo aprimoramento.

1.3. Objetivos

O objetivo deste trabalho é reduzir a deficiência de informação sobre a gestão de riscos através da proposição de métricas. Ou seja, tentar tornar a gestão de riscos algo menos subjetivo para o gerente de projetos.

Essas métricas que venham a ser definida devem ter como princípio a literatura já existente, para que seja proposto algo válido e que venha a ser realmente aplicado em alguma organização.

Finalmente, para que se atinja um resultado realmente concreto é necessário validar as métricas propostas, e isso será feito através da incorporação dessas métricas na ferramenta mPRIME (Multiple Project Risk Management).

O mPRIME é uma ferramenta de gestão de riscos para ambientes de múltiplos projetos de desenvolvimento de software, desenvolvida como *add-in* para o Microsoft Project [LOPES e SOUTO 2005], produzida no Centro de Informática da Universidade Federal de Pernambuco.

1.3.1. Objetivos Específicos

Para atender ao objetivo geral, alguns objetivos específicos devem ser alcançados:

- Definir quais as principais deficiências da gestão de riscos, a fim de se possuir alguma alternativas para definição de métricas.
- Investigar métricas na literatura que possam servir como ponto de partida para resolver algumas das deficiências descobertas no item anterior.
- Adequar as métricas selecionadas no item anterior para o contexto de gerenciamento de riscos.
- Implementação dessas métricas na ferramenta mPRIME, como forma de validação do trabalho.

1.4. Estrutura do Trabalho

Além deste capítulo inicial que explanou sobre o contexto, motivação e objetivos do trabalho, este é composto por mais quatro capítulos:

O capítulo 2 – **Estudo sobre Métricas** – faz uma discussão sobre métricas em engenharia de software, conceitua métricas, as classificações existentes, também traz informações sobre Análise de Pontos por Caso de Uso e métricas existentes na Gerência de Risco.

O capítulo 3 – **Proposta de Trabalho** – propõe algumas novas métricas para o processo de gerenciamento de riscos e aborda como será realizado este trabalho.

O capítulo 4 – **Resultados Obtidos** – valida e mostra resultados obtidos com as métricas apresentadas no capítulo anterior.

O capítulo 5 – **Considerações Finais e Trabalhos Futuros** – traz a conclusão do trabalho, será realizado um resumo do trabalho e também mostrando ainda os trabalhos relacionados com este e futuros trabalhos que venham estender este.

2. ESTUDO SOBRE MÉTRICAS

2.1. Introdução à Métricas

Uma diferença fundamental entre uma ciência bem desenvolvida como física e alguma outra menos desenvolvida é o grau em que as coisas são medidas [ROBERTS 1979]. A Engenharia de Software não é diferente como afirma Pressman [PRESSMAN 1995], Métricas são fundamentais para qualquer engenharia, e engenharia de software não é uma exceção.

Essencialmente, métricas de software tem haver com mensurar o produto software e o processo pelo qual é desenvolvido [MILLS ET AL 1988].

Mensurar pode ser aplicado em processo de software com a intenção de melhorá-lo continuamente, ou dentro de um projeto para auxiliar as estimativas, o controle de qualidade, avaliar a produtividade e controlar o projeto. Finalmente, medidas podem ser usadas pelos engenheiros de software para ajudar a avaliar a qualidade técnica dos produtos e auxiliar em tomadas de decisões táticas enquanto o projeto procede [PRESSMAN 1995]

Apesar de todos os fatores positivos apresentados, métricas ainda não são amplamente usadas em organizações que desenvolvem software. Isto se deve principalmente por que os resultados de um programa de métricas aparece em longo prazo e muitas organizações não estão dispostas a esperar, principalmente quando envolve a aplicação de algum capital. Além do que, exigem uma grande mudança na cultura organizacional para se iniciar a coleta de dados

Uma maneira de se amenizar a ocorrência de falhas nos programas de métricas é a utilização de boas métricas. Os pontos fortes de uma métrica são os seguintes:

- Simples, de fácil definição.
- Fácil obtenção, ou seja, não é viável uma métrica que necessite de muito esforço ou capital para ser obtida.
- Válida, a métrica deve conseguir medir o que se propõe a medir.

2.1.1. Classificação das Métricas

Métricas de software podem ser categorizadas de diversas maneiras. A primeira delas é a divisão em métricas de processo e de produto. Métricas de Produto são medidas do software em qualquer estágio de desenvolvimento, desde a instalação ao sistema instalado. Podem medir a complexidade de um *design*, o tamanho final de um

programa ou número de páginas de documentação produzidas. Métricas de Processo, ao contrário, são medidas do processo de desenvolvimento de software como tempo de desenvolvimento, tipo da metodologia usada, ou a média de experiência de programação da equipe [apostila sei].

Outra classificação é a separação entre métricas objetivas e subjetivas. A primeira, independente de quem seja o observador, o resultado vai ser o mesmo, por exemplo, Linhas de Código. As subjetivas, ao contrário, o resultado pode variar de acordo com o observador, por exemplo, a classificação de um ator de caso de uso em simples, médio e complexo, na métrica de pontos por caso de uso.

Uma outra possibilidade de classificação de métricas é a separação entre métricas primitivas e computadorizadas. As primeiras, o resultado da métrica é obtido diretamente, como Linhas de Código. Já as computadorizadas exigem algum tipo de cálculo para a obtenção do resultado, como por exemplo, a relação da quantidade de Linhas de Código por Pessoa a fim de mensurar a produtividade.

2.2. Métricas de Tamanho de Software

Uma das principais aplicações das métricas de software é a mensuração da produtividade de uma organização. A “saída” do desenvolvimento de software como uma função de esforço e tempo aplicados [PRESSMAN 1995]. Este tipo de métrica é fundamental para o

sucesso de uma empresa de software, já que com essa dimensão é possível aferir informações como prever o tempo e custo de desenvolvimento de um projeto, controlar as atividades que são executadas no projeto.

Neste sentido, a comunidade de Engenharia de Software vem propondo várias possibilidades de métricas, como: Linhas de Código, Análise por Pontos de Função e mais recentemente com a evolução do paradigma orientado a objeto, Análise de Pontos por Caso de Uso.

2.2.1. Pontos de Função

Este modelo foi proposto em 1979, por Albrecht, sua estratégia para determinar o tamanho de um sistema é de acordo com a quantidade de requisitos funcionais, considerando a complexidade de cada um deles.

Para se determinar a quantidade de pontos de função de um projeto, primeiramente são calculados os “Pontos de Funções Não Ajustados” e em seguida é aplicada uma fórmula de ajuste, que é influenciada pelos “Fatores Técnicos” do projeto, para finalmente determinar o resultado.

Os Pontos de Funções Não Ajustados são calculados a partir da divisão do sistema em “entrada externas”, “saídas externas” e “consultas externas” e cada entrada, saída ou consulta recebe um peso de acordo com a sua complexidade. Em seguida, os recursos de dados são classificados em “Arquivos Lógicos Internos”, pertencentes ao sistema, e

“Interfaces Externas”, externos ao sistema, e, da mesma maneira, cada “Arquivos Lógicos Internos” ou “Interfaces Externas” recebe um peso. Finalmente, todos os pontos de cada classificação são somados, como o exemplo mostrado no Quadro 2.

Quadro 2 – Exemplo de Cálculo dos Pontos de Função Não Ajustados

Classificação	SIMPLES			MÉDIO			COMPLEXO			TOTAL
	Número	Peso	Total	Número	Peso	Total	Número	Peso	Total	
Saídas	2	4	8	5	5	25	5	7	35	68
Consultas	5	4	20	6	5	30	6	7	42	92
Entradas	7	3	21	4	4	16	3	6	18	55
Arquivos Internos	4	7	28	1	10	10	8	15	120	158
Interfaces Externas	2	5	10	3	7	21	7	10	70	101
Total de Pontos de Função Não Ajustados										474

O próximo passo é determinar a influência dos “Fatores Técnicos” para se determinar o resultado da estimativa. Um pequeno questionário deve ser respondido com valores no intervalo de zero a cinco, em que zero significa sem influência no projeto e cinco, essencial. O questionário a ser respondido é mostrado no Quadro 3.

Quadro 3 – Fatores Técnicos

Fatores Técnicos	
Comunicação de Dados	Atualização On-Line
Processamento de Dados Distribuídos	Processamento Complexo
Desempenho	Reusabilidade

Utilização de Equipamento	Facilidade de Implantação
Volume de Transações	Facilidade Operacional
Entrada de Dados On-Line	Múltiplos Locais e Organizações do Usuário
Usabilidade	Manutenibilidade

Para a finalização do cálculo dos pontos, a fórmula de ajuste que se utiliza do valor dos “Pontos de Função Não Ajustados” e das respostas do questionário de “Fatores Técnicos” do projeto.

A equação da formula de ajuste é mostrada a seguir:

$$PF = PFNA \times \left[\left(\sum_1^{14} FTi \times 0.01 \right) + 0.65 \right]$$

Onde, PF é a quantidade de pontos de função, PFNA é quantidade de Pontos de Função Não Ajustados e FT é a resposta de um dos fatores técnicos, ou seja todas as respostas do questionário são somadas.

Entretanto, este modelo vem sendo criticado e alguma fraquezas são notadas. Algumas desvantagens são que Pontos de Função não podem ser calculados automaticamente. Não é objetivo já que é necessário fazer várias decisões subjetivas. Isto é toda entrada, saída e interface devem ser classificadas como simples, média ou complexa e os Fatores Técnicos também devem ser classificados por humanos também. Além de não levar em conta qualquer influência de fatores associados a recursos humanos [KARNER 1993].

2.2.2. Pontos de Casos de Uso

2.2.2.1. Casos de Uso

Casos de Uso não foi definido de uma única vez e sim passou por um processo de evolução, até se tornar da forma conhecida hoje. Este processo iniciou-se no fim da década de 60, quando Ivar Jacobson trabalhava para Ericsson e finalizou-se em 1994 com Alistair Cockburn.

Um caso de uso é uma narrativa documentada que descreve uma seqüência de eventos de um ator usando um sistema para completar um processo [JACOBSON 1992]. Os atores são uma entidade externa ao sistema que interage com os casos de uso, podem ser tanto pessoas como outros sistemas.

A maneira mais comum de se representar casos de uso de um sistema é através de um modelo UML (Unified Modeling Language). Neste modelo, todas as funcionalidades de um sistema são representadas através dos vários casos de uso onde também são mostradas as relações desses com seus respectivos atores. Um exemplo pode ser visto na Figura 1, onde a representação gráfica dos atores são pequenos bonecos e a dos casos de uso são elipses.

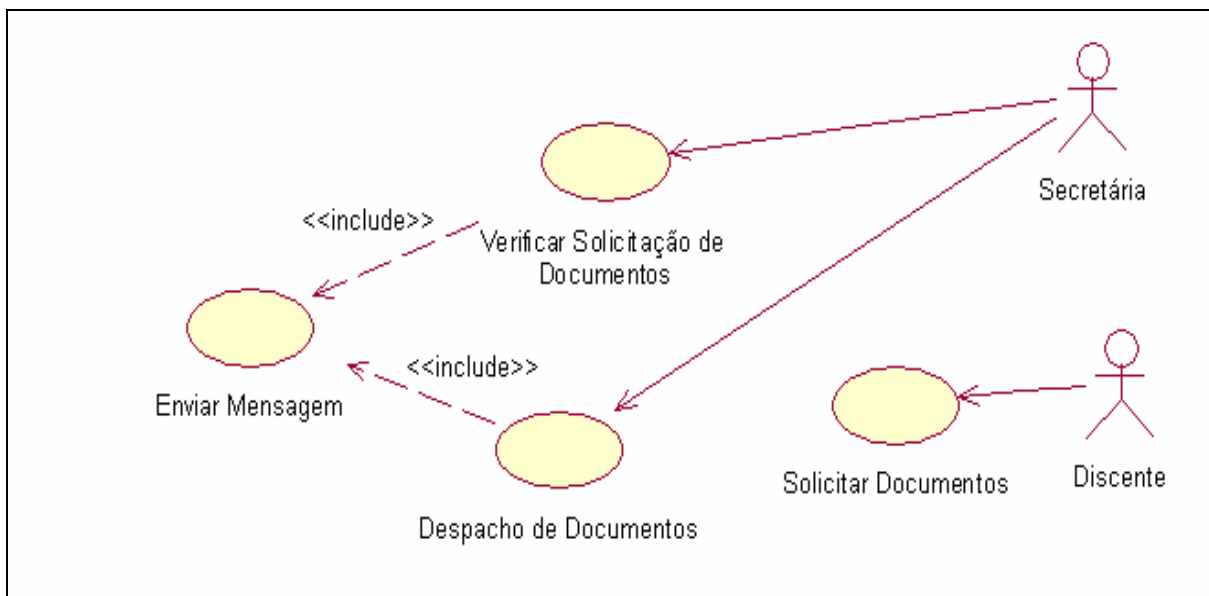


Figura 1 – Exemplo de um modelo de Caso de Uso

Casos de uso também podem possuir relacionamentos uns com os outros. Existem duas formas: inclusão e extensão. A inclusão de um caso de uso em outro, significa que o último deve incluir toda a funcionalidade existente no primeiro. Na Figura 1, há um relacionamento de “include”. Já na relação de extensão, a diferença entre a inclusão resume-se que na inclusão a relação é de obrigação de incluir toda a funcionalidade, enquanto na extensão, opcional.

2.2.2.2. Cálculo de Pontos de Casos de Uso

A técnica de estimativa por Pontos de Caso de Uso foi proposta em 1993 por Gustav Karner, da Objectory (hoje, Rational Software). Ela baseia-se em dois métodos bastante utilizados – o mecanismo de Pontos de Função e uma metodologia conhecida como Mk II, uma adaptação da técnica de Pontos de Função, bastante utilizada na Inglaterra [FREIRE 2006].

O processo de estimativa inicia-se medindo a funcionalidade do sistema baseado no modelo de caso de uso em uma conta chamada Pontos de Casos de Uso Não Ajustados. Fatores Técnicos envolvidos no desenvolvimento destas funcionalidades são avaliados similarmente a Pontos de Função. O último passo na estimativa não é entretanto de Pontos de Funções e é um novo fator chamado Fatores de Ambiente [KARNER 1993].

As principais vantagens dessa abordagem sobre pontos de função é que primeiramente, leva em conta fatores humanos na sua estimativa e outro ponto é que o nível de abstração para se obter uma estimativa é muito maior.

O método de cálculo dos Pontos de Caso de Uso funciona da seguinte maneira:

Primeiramente, são classificados os atores do sistema em Simples, Médio e Complexo e atribuindo pesos 1, 2, 3 para cada classificação, respectivamente. Esta classificação é melhor visualizada no Quadro 4. Em seguida é obtido o Total de Pesos não Ajustados dos Atores somando os produtos das quantidades de atores pelo seu respectivo peso.

Quadro 4 – Classificação dos Atores do Sistema

Classificação dos Atores	Peso
Simples	1
Médio	2
Complexo	3

O segundo passo, é a classificação dos casos de uso em Simples, Médio e Complexo e atribuindo pesos 1, 2, 3 para cada classificação respectivamente. Existem várias abordagens para determinar a complexidade do caso de uso, duas delas são pelo número de transações envolvidas e outra pelo número de classes envolvidas. No final deste passo, obtém-se o Total de Pesos não Ajustados dos Casos de Uso somando os produtos das quantidades de casos de uso pelo seu respectivo peso.

Quadro 5 – Classificação dos Casos de Uso do Sistema

Classificação dos Atores	Peso
Simples	1
Médio	2
Complexo	3

Pode-se então obter o total de Pontos de Casos de Uso Não Ajustados através do somatório do Total de Pesos Não Ajustados dos Atores e dos Casos de Uso.

O próximo passo é determinar o “Fator de Complexidade Técnica”. Inicialmente, deve-se responder um pequeno questionário com valores de 0 a 5, onde 0 a 5, onde 0 é sem influência e 5 é um fator determinante. Após determinar os valores de resposta dos questionários, estes devem ser multiplicados pelos seus respectivos pesos, como mostrado no Quadro 6.

Quadro 6 – Fatores de Complexidade Técnica do Sistema

Fator Técnico	Peso
Sistema Distribuído	2
Desempenho da Aplicação	1
Eficiência do Usuário Final (On-Line)	1
Código Reusável em Outras Aplicações	1
Complexidade de Processamento	1
Facilidade de Instalação	0.5
Facilidade de Uso	0.5
Portabilidade	2
Facilidade de Alterações	1
Concorrência	1
Segurança	1
Acesso Direto a Terceiros	1
Necessidade de Facilidades Especiais de Treinamento para Usuários	1

O total deve ser somado e em seguida ser aplicada a seguinte formula:

$$FCT = 0.6 + (0.01 \times SFT)$$

Onde, FCT é o fator de complexidade técnica do projeto e SFT é o somatório dos fator técnico do projeto.

Em seguida, deve-se determinar o “Fator de Complexidade Ambiental”, seu cálculo é semelhante ao “Fator de Complexidade Técnica”. Inicialmente, deve-se responder um pequeno questionário com valores de 0 a 5, onde 0 a 5, onde 0 é sem influência e 5 é um fator determinante. Após determinar os valores de resposta dos questionários, estes devem ser multiplicados pelos seus respectivos pesos, como mostrado no Quadro 7.

Quadro 7 – Fatores de Complexidade Ambiental do Sistema

Fator Ambiental	Peso
Familiaridade da Equipe com RUP	1.5
Experiência da Equipe	0.5
Experiência da Equipe em Orientação a Objeto	1
Capacidade dos Analistas da Equipe	0.5
Motivação	1
Estabilidade dos Requisitos	2
Estagiários ou Funcionários em Tempo Parcial	-1
Domínio da Tecnologia ou da Configuração do Ambiente	-1.5

O total deve ser somado e em seguida ser aplicada a seguinte formula:

$$FCA = 1.4 + (-0.03 \times SFA)$$

Onde, FCA é o fator de complexidade ambiental do projeto e SFA é o somatório do fator ambiental do projeto.

Finalmente, é determinado a quantidade de Pontos de Casos de Uso, através da equação:

$$PCUA = PCUNA \times FCT \times FCA$$

Onde, PCUA são os Pontos de Casos de Uso do Projeto; PCUNA, Pontos de Casos de Uso Não Ajustados; FCT, Fator de Complexidade Técnica e FCA, Fator de Complexidade Ambiental.

2.3. Métricas em Gerência de Risco

Das muitas disciplinas da engenharia, engenharia de software é mais suscetível aos riscos que muitas outras áreas. Embora projetos, processo e requisitos de software evoluam, ainda sim a complexidade dos produtos é maior, e há um número maior de potenciais fatores de risco que em muitas outras disciplinas [FAIRLEY 1989].

O termo gerenciamento de risco tem sido usado de duas maneiras na literatura de gerência de projetos. De um lado tem sido referida como a disciplina que estuda como identificar, delegar e eliminar riscos

[BOEHM 1989]. Por outro lado é referido como atividade ou processo que tenta identificar o que pode ir errado em um projeto e praticar ações a fim de evitar problemas no futuro [CHARETTE 1989; HALL 1998].

Através de perspectivas globais de negócios muitas organizações estão tornando-se cada vez mais dependentes do sucesso ou do fracasso dos softwares que desenvolvem. Neste contexto, a gerência de riscos não é apenas baseada em boas práticas para o desenvolvimento de software, mas sim, boas práticas para gerir negócios [GUSMAO e MOURA 2005].

Um ponto fundamental é a definição do que é risco. Entretanto, existem várias definições e usos para o termo risco, conforme visto nas subseções anteriores. Ainda que o conceito de risco esteja bastante associado a perigos e impactos negativos, já vem sendo utilizado como "exposição a conseqüências da incerteza", sendo cada vez mais aplicado tanto no gerenciamento de perdas como no de ganhos potenciais.

2.3.1. Classificação dos Riscos

Várias categorizações de riscos são possíveis, uma delas é baseada na área do projeto que foi atingida.

Riscos de Projeto de Software. Define os parâmetros operacionais, organizacionais e contratuais do desenvolvimento de software. O risco de projeto é primeiramente uma responsabilidade da gerência. Risco de projeto inclui limites de recursos, interfaces, relacionamentos com fornecedores ou restrições de contratos.

Riscos de Processo de Software. Neste tipo de risco estão incluídos os problemas técnicos e de gerência. Nos procedimentos de gerência pode-se encontrar riscos em atividades como: planejamento, definição e contratação de equipe de trabalho, garantia de segurança e configuração de gerência. Nos procedimentos técnicos, podem-se encontrar riscos nas atividades: análise de requisitos, projeto, codificação e testes.

Riscos de Produto de Software. Contém as características intermediárias e finais do produto. Estes tipos de riscos têm origens nos requisitos de estabilidade do produto, performance, complexidade de codificação e especificação de testes.

Uma outra abordagem foi proposta pelo SEI (Software Engineering Institute), a sua taxonomia de riscos [CARR ET AL. 1993] sugeriu classificar os riscos em três categorias: Engenharia do Produto, Ambiente de Desenvolvimento e Restrições do Produto.

Engenharia de Produto – composta pelas atividades da Engenharia de Sistema e da Engenharia de Software envolvidas na criação de um sistema que satisfaça requisitos específicos e expectativas do cliente. As atividades incluem a análise e especificação dos requisitos de sistema e de software; a modelagem de software e implementação; a integração de componentes de hardware e de software; e os testes de software e sistema.

Ambiente de Desenvolvimento – endereça o ambiente do projeto em desenvolvimento e o processo utilizado. Incluem processo de

desenvolvimento e sistema, métodos de gerenciamento e ambiente de trabalho.

Restrições de Programa – faz referência aos fatores externos ao projeto. Fatores que normalmente estão fora do controle do projeto e mesmo assim influenciam o sucesso ou constituem fonte de riscos substanciais.

Embora estas atividades sejam tipicamente realizadas na fase de planejamento do projeto, ao longo do projeto serão identificados novos riscos não identificados anteriormente. Esses novos riscos devem ser tratados da mesma forma, isto é, uma vez identificados, serão analisados e deve ser feito o planejamento necessário. Novamente, isto supõe uma política de ampla comunicação entre todas as pessoas que têm algum tipo de relação com o projeto.

2.3.2. Atividades da Gerência de Risco

Diferentemente da definição de riscos em software, as atividades que devem ser executadas para uma gestão de riscos eficiente parecem já ter entrado em um consenso, não só na literatura, como guias e modelos de qualidade existentes.

As atividades são as seguintes:

Planejar a Gerência de Risco. Esta atividade tem a finalidade de definir a estratégia da gestão de riscos, dos recursos necessários para a

realização do processo e por fim, da efetivação das ações consideradas necessárias no plano de Gerência de Riscos.

Identificar Riscos. A identificação dos riscos é a atividade inicial de um projeto de software. Objetiva um levantamento preliminar de todas as possibilidades de riscos existentes no projeto. O aspecto mais importante da atividade de identificação de riscos é compor uma documentação formalizando os dados coletados.

Analisar Riscos. Nesta atividade são caracterizados os aspectos mais importantes de cada risco, com a finalidade de explorar as melhores estratégias de mitigação (eliminação). De forma geral, os riscos são categorizados e priorizados, segundo algum critério específico estabelecido, para tornar a gerência concentrada nos riscos considerados prioritários.

Planejar Respostas aos Riscos. O planejamento é uma atividade, da Gerência de Riscos, que envolve, em geral, a determinação dos riscos a serem gerenciados, dos planos de ação para os riscos sob controle da gerência e dos planos de contingência para os riscos que se encontram além das capacidades de mitigação.

Monitorar Riscos. O monitoramento dos riscos é a observação da efetividade dos planos de ação na execução do desenvolvimento do projeto de software. O objetivo é prover informações precisas e contínuas para habilitar a gerência de risco a atuar de forma preventiva e não reativa aos eventos adversos. Como benefício desta atividade, tem-se a melhor compreensão do andamento do projeto por parte dos membros

das equipes de desenvolvimento. Cada risco monitorado, possui um ciclo de atualização próprio. A frequência de atualização depende dos recursos disponíveis e da rapidez com que o produto se desenvolve.

Controlar Riscos. A atividade de controle dos riscos avalia a situação corrente para determinar eventuais desvios do planejado. O controle dos riscos envolve alteração das estratégias de mitigação, quando se fizer necessário; utilização de ações previamente planejadas de contingência; encerramento de trabalhos relacionados a um determinado risco, quando este deixar de existir, entre outras. A utilização de cronogramas é essencial para a atividade de controle na Gerência de Riscos, pois o controle explícito de tarefas de mitigação de riscos facilita o acompanhamento do progresso e da eficácia destes planos.

Comunicar os Riscos. A comunicação entre as equipes e membros do projeto de software é um dos fatores mais importantes para a realização bem sucedida da gerência de riscos.

Deve-se ressaltar que todas as atividades são baseadas e centradas na comunicação, devendo ser realizadas de forma cíclica e contínua dentro do processo de Gerência de Riscos utilizado.

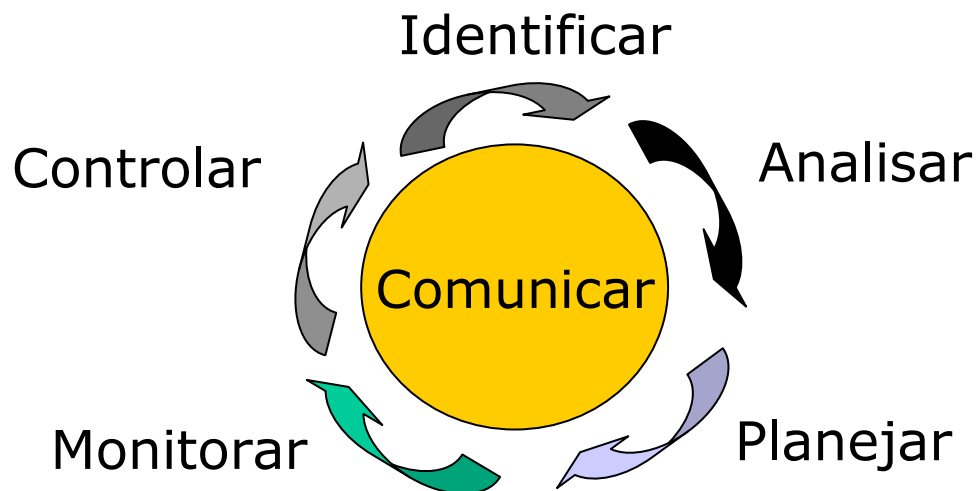


Figura 2 – Figura Representando o Ciclo da Gestão de Riscos

2.3.3. Métricas

Atualmente, ainda existe pouca informação na literatura de Engenharia de Software sobre a Gerência de Riscos e principalmente de métricas para esta gerência. Diferentemente do mercado financeiro, onde a Gerência de Riscos já está consolidada e existem várias métricas e indicadores que auxiliam o processo.

Dentre algumas métricas que foram propostas para software, duas merecem destaque, foram propostas por Boehm, são elas: Exposição do Risco e Influência da Redução de Risco. Ambas são utilizadas na análise do risco e sua priorização [MYERSON].

A Exposição do Risco é definida como:

$$ER = PROB \times IMP$$

Onde, ER é a exposição ao risco, PROB é probabilidade de acontecimento e IMP é o valor do impacto associado ao risco.

Já a Influência da Redução de Risco é definida como;

$$IRR = (REa - REd) / CRR$$

Onde, IRR é a Influência da Redução de Risco, REa é a Exposição do Risco antes da aplicação de ação aos riscos, e REd é a exposição depois da aplicação das ações e CRR é o custo da redução da probabilidade do risco.

Essa medida tem como principal papel medir a relação custo/benefício de se efetuar ações para diminuir chances de acontecimento do problema.

3. PONTOS DE RISCO

3.1. Introdução

Neste capítulo será apresentada a real motivação desse trabalho, a definição de uma métrica que venha auxiliar o processo de Gerência de Riscos em ambientes de desenvolvimento de software.

Decidiu-se que a métrica deveria abordar o problema de definir quantitativamente o nível de risco de um projeto como um todo. Dessa maneira, seria possível comparar dois projetos e definir qual deles é mais arriscado.

A partir de então, iniciou-se uma pesquisa dentro da literatura de Engenharia de Software para analisar métricas, a fim de selecionar algumas que pudessem servir como ponto de partida para a prover as informações desejadas na métrica proposta neste trabalho.

Percebeu-se então, uma certa similaridade entre as métricas existentes para mensurar o tamanho do software e o que se deseja prover nesta métrica. Da mesma maneira deseja-se medir o tamanho, ou melhor definindo, um número que signifique o nível de risco do projeto como um todo, a fim de se possuir uma estimativa de riscos do projeto e

também da produtividade, ou melhor, da eficácia da gestão de riscos de uma organização.

A pesquisa a partir de então partiu para o foco de se selecionar qual a melhor métrica de estimativa existente para se fazer uma analogia com a gestão de riscos. A resposta que se propõe nesse trabalho é Pontos de Casos de Uso, essa escolha foi baseada por ser uma das métricas mais comuns para Projetos Orientados à Objetos, na atualidade, o que facilitaria o uso da métrica aqui proposta e também pela adequação que conseguimos ser mais intuitiva que em com outras métricas.

Devido a essa inspiração na métrica de Pontos de Casos de Uso, resolveu-se intitular a métrica aqui proposta por Pontos de Risco.

3.2. Cálculo de Pontos de Risco

A partir de então, buscaram-se adequações entre casos de uso para riscos, mas que essas adequações não fossem apenas intuitivas mas também que realmente trouxessem alguma melhoria ao processo de gestão de riscos, ou seja resultasse em algo positivo para a organização que a utilizar.

Como mostrado anteriormente na seção 2.2.2.2, o cálculo de Pontos de Casos de Uso de um projeto é efetuado de acordo com o seguinte procedimento: Primeiramente, são calculados os Pontos de Casos de Uso Não Ajustados e em seguida é aplicada uma formula de

ajuste que é influenciada pelos “Fatores de Complexidade Técnica” e “Fatores de Complexidade Ambiental” que atuam no projeto.

Na analogia aqui proposta, o procedimento de cálculo do Pontos de Risco é idêntico. Inicialmente, deve ser calculado os Pontos de Riscos Não Ajustados do projeto, da mesma maneira que em casos de uso, e o resultado final é retornado pela aplicação de uma formula de ajuste que no nosso caso é influenciado pelo que foi nomeado “Fatores Caracterizadores do Projeto”, que é um agrupamento dos Fatores Técnicos e Ambientais do projeto, isto não significa uma perda de precisão já que os “Fatores Caracterizadores do Projeto”, incluem tanto questões técnicas quanto de recursos humanos.

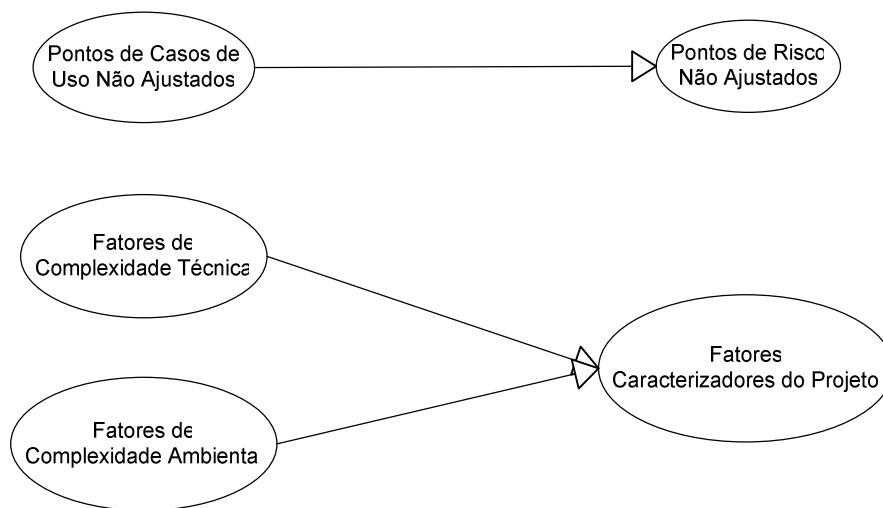


Figura 3 – Mapeamento entre Casos de Uso e Riscos

3.3. Cálculo do Pontos de Risco Não Ajustados

Em Pontos de Casos de Uso, o cálculo do Pontos de Casos de Uso Não Ajustados é feito através da soma do Total de Pesos dos Atores e de Casos de Uso Não Ajustados. Na abordagem proposta por esse trabalho, os Pontos de Risco Não Ajustados são definidos pela soma do “Total de Pesos dos Responsáveis pelo Risco”, mapeando a classificação de Atores, juntamente com a soma do Total de Pesos de Riscos Não Ajustados, que é mapeado ao Total de Pesos de Casos de Uso Não Ajustados.

O Total de Pesos de Responsáveis por Risco Não Ajustado, é determinada pela a classificação dos responsáveis de acordo com seu grau de habilidade em controlar os riscos. Infelizmente, dentro deste trabalho não foi possível determinar uma classificação clara dos responsáveis, não houve como determinar exatamente a influência dos responsáveis na probabilidade de ocorrência de um risco, esse estudo poderá ser feito como trabalho futuro, como será discutido mais a frente. Devido a esse imprevisto, inicialmente, os Pontos de Risco Não Ajustados será igual ao valor do Total de Pesos de Risco Não Ajustados.

Para calcular o Total de Pesos de Riscos Não Ajustados, primeiramente, os riscos são classificados de acordo com sua exposição em: Muito Baixo, Baixo, Médio, Alto, Muito Alto e atribuindo pesos 1, 2, 3, 4, 5; respectivamente. Em seguida chega-se ao resultado final através do somatório dos produtos das quantidades de riscos pelo seu respectivo peso. Isso é melhor demonstrado no Quadro 8.

Quadro 8 – Classificação dos Riscos do Sistema

Classificação	Descrição	Peso
Muito Baixo	Exposição maior que 0.0 e menor que 0.2	1
Baixo	Exposição maior que 0.2 e menor que 0.4	2
Médio	Exposição maior que 0.4 e menor que 0.6	3
Alto	Exposição maior que 0.6 e menor que 0.8	4
Muito Alto	Exposição maior que 0.8 e menor que 1.0	5

Essa classificação em cinco níveis e os valores dos pesos foi inspirada na forma de determinação do impacto mais comumente usado pela comunidade de Engenharia de Software. Normalmente, o impacto é definido estaticamente geralmente em três ou cinco níveis e com uma diferença de 20% entre cada nível. Por exemplo, a escala de impacto mostrada, abaixo:

Quadro 9 – Exemplo de Escala de Impacto

Classificação	Peso
Muito Baixo	0.2
Baixo	0.4
Médio	0.6
Alto	0.8
Muito Alto	1.0

3.4. Fatores Caracterizadores do Projeto

Na abordagem proposta neste trabalho, os fatores técnicos e ambientais foram aglutinados em uma única categoria que foi nomeada “Fatores Caracterizadores do Projeto”, uma vez que contém questões técnicas e humanas.

Quadro 10 – Fatores Caracterizadores do Projeto

Classificação	Respostas	Peso
Tamanho da Equipe	0= Muito pequena (1 a 6 pessoas) 1=Pequena (7 a 20 pessoas) 2=Média (21 a 50 pessoas) 3=Grande (51 a 100 pessoas) 4=Muito grande (mais de 100 pessoas)	1.75
Distribuição Geográfica da Equipe	0= Mesma sala 1=Mesmo prédio, salas diferentes 2=Mesma cidade, mesma empresa, prédios diferentes 3=Mesma cidade, empresas diferentes 4=Cidades diferentes	1.28
Experiência da Equipe no Processo (em média)	0= Nenhum projeto 1=1 projeto 2=2 a 3 projetos 3=4 a 5 projetos 4=Mais de 5 projetos	1.13
Experiência da Equipe na Aplicação	0= Nenhum projeto 1=1 projeto 2=2 a 3 projetos 3=4 a 5 projetos	1.20

	4=Mais de 5 projetos	
Experiência da Equipe com a tecnologia de desenvolvimento.	0= Nenhum projeto 1=1 projeto 2=2 a 3 projetos 3=4 a 5 projetos 4=Mais de 5 projetos	1.50
Criticidade do Projeto (possível consequência de uma falha do sistema)	0= Perda de conforto 1=Prejuízos baixos, perdas facilmente recuperáveis 2=Prejuízos moderados, perdas recuperáveis 3=Prejuízos altos, perdas irre recuperáveis 4=Risco de Vida	1.20
Tamanho do Projeto (investimento no projeto)	0= Menos de R\$ 50.000,00 1=Entre R\$ 50.000,00 e R\$ 150.000,00 2=Entre R\$ 150.000,00 e R\$ 1.000.000,00 3=Entre R\$ 1.000.000,00 e R\$ 3.000.000,00 4=Mais de R\$ 3.000.000,00	0.68
Prioridade (Qualidade x Tempo)	0= 10/90 1=30/70 2=50/50 3=70/30 4=90/10	1.13

O procedimento de cálculo é similar ao fator técnico ou ambiental na abordagem de caso de uso. Inicialmente, deve-se responder um pequeno questionário com valores de 0 a 4. Após determinar os valores de resposta dos questionários, estes devem ser multiplicados pelos seus respectivos pesos, como mostrado no quadro 10, o total deve ser somado e em seguida ser aplicada a seguinte formula:

$$FP = 1.05 + (0.015 \times SFP)$$

Onde, FP é o Fator Caracterizador do Projeto e SFP é somatório retornado a partir da resposta do questionário.

O questionário proposto foi retirado de um modelo de caracterização de projeto proposto por Coelho [COELHO 2003]. Este é composto por questões com cinco possíveis respostas, por isso os valores da resposta variam de zero a quatro, diferentemente de casos de uso, entre zero e cinco. As respostas foram ordenadas de acordo com elevação da criticidade, ou seja 0 é o menor valor em criticidade e 4, o valor máximo. Foram retiradas duas perguntas do modelo original, sobre conograma e orçamento, devido suas respostas serem subjetivas.

Os pesos propostos para cada questão foram definidos empiricamente através de uma pesquisa com alunos da Qualiti que se preparam para a certificação PMP (Project Manager Professional). Resumidamente, a pesquisa consistiu de perguntar qual fatores propostos pelo modelo de caracterização do Projeto de Coelho [COELHO 2003] eram considerados mais críticos para a gerência de riscos.

Já a equação proposta para o cálculo do fator caracterizador do projeto, não possuiu nenhuma forma de validação para chegar a tal resultado. Simplesmente, acho-se razoável que a influência dos fatores caracterizadores em um caso extremo, onde todos as questões obtiverem

respostas máximas em criticidade, poderia ser adicionado 50% mais pontos ao resultado final da métrica. É preciso um estudo mais profundo para a determinação dessa equação.

Um ponto crucial que deve ser considerado quando for proposta uma nova equação é que o Fator de Complexidade Técnica deve ter seus valores reduzidos gradativamente, caso o valor das probabilidades dos riscos se encontrem muito altas. Uma vez que as probabilidades estão muito altas, de certa maneira já há uma preocupação grande com os riscos do projeto, então não há motivo para acrescentar muitos mais pontos aos projetos através dos fatores caracterizadores do projeto.

3.5. Fórmula de Ajuste

Finalmente para se obter o resultado dos Pontos de Risco, é similar a Pontos Por Caso de Uso, utilizando se do resultado dos Pontos de Risco Não Ajustado e o valor dos Fatores Caracterizadores do Projeto, como na equação:

$$PR = PRNA \times FCP$$

Onde, PR é a quantidade de Pontos de Risco, PRNA são os Pontos de Riscos Não Ajustados e o FCP é o Fator Caracterizador do Projeto.

3.6. Aplicações

Depois de demonstrado o procedimento de cálculo, é necessário mostrar as aplicações que esta métrica pode ter dentro de uma organização desenvolvedora de software.

A primeira aplicação, diz respeito a viabilidade de um projeto de software. Se uma organização possui o histórico da quantidade de Pontos de Riscos dos projetos, então é possível determinar um limite de pontos de risco por projeto que organização suporta. Além de que caso seja viável, é possível ter uma previsão de quanto atraso e custo ocorrerá devido a ocorrência de problemas.

Uma segunda aplicação, é possível se determinar a eficiência da gestão de riscos da organização. Se a organização mantiver um histórico da variação dos Pontos de Risco de um projeto, antes e depois das aplicações de ações que visam reduzir a ocorrência é possível definir uma relação entre Pontos de Risco por Ação aplicada, ou por unidade monetária do custo da aplicação das ações. Para tal, basta adaptar a métrica Influência da Redução de Risco, vista na seção 2.3.3, substituindo a exposição pela aplicação de Pontos de Risco antes e depois das ações da gestão de riscos.

Uma outra possibilidade, é que em organizações que executam projetos paralelamente, é possível estabelecer um *ranking* de projetos de

acordo com riscos, utilizam-se da quantidade pontos de risco do projeto, assim percebe-se quais projetos precisam de uma maior atenção.

Uma outra abordagem, em vez de ver os riscos de um projeto como todo, calcular os pontos de risco de acordo com alguma classificação, assim é possível perceber a quantidade de pontos que provem de cada área de projeto auxiliando na análise de riscos. Por exemplo, pode se calcular os pontos de um projeto de acordo com a classificação de riscos de Produto, Projeto e Negocio ou utilizando a categorização do SEI (Software Engineering Institute) [CARR ET AL 1993], Engenharia de Produto, Restrições de Programa e Ambiente de Desenvolvimento.

4. VALIDAÇÃO

Neste capítulo, será apresentada a ferramenta mPrime (Multiple Project Risk Management), onde foi implementada a métrica de Pontos de Risco como forma de validar a métrica proposta no capítulo anterior.

4.1. mPrime

O mPRIME é uma ferramenta de gestão de riscos para ambientes de múltiplos projetos de desenvolvimento de software, desenvolvida como add-in para o MS Project [LOPES e SOUTO 2005]. Sua definição e implementação tiveram por base estudos acadêmicos, em nível de doutorado, mestrado e graduação, do Centro de Informática da Universidade Federal de Pernambuco – CIn-UFPE.

Seu desenvolvimento ocorreu durante a disciplina de Projeto de Desenvolvimento do Centro de Informática – UFPE, no primeiro semestre letivo do ano de 2005.

A ferramenta trabalha com a taxonomia de riscos definida pelo SEI (Software Engineering Institute) [CARR ET AL 1993]. Além disso, é aderente ao processo do CMMI (Capability Maturity Model Integration), oferecendo suporte a todos os processos da gestão de riscos: Planejar a Gerência de

Riscos, Identificação de Riscos, Análise dos Riscos, Planejar Respostas aos Riscos, Monitoramento e Controle.

O mPRIME utiliza componentes de inteligência artificial que auxiliam na execução das atividades de identificação, monitoração e controle dos riscos [GUSMÃO ET AL 2005]. O principal diferencial está na disponibilidade de funções que levantam situações de riscos, através de listas de verificação, facilitando a identificação de riscos de projetos e entre projetos [FERNANDES e BUARQUE 2005].

Além destes componentes inteligentes, a ferramenta mPRIME disponibiliza um conjunto de técnicas de levantamento de riscos baseadas em modelos consolidados na área de gestão [PMI 2000].

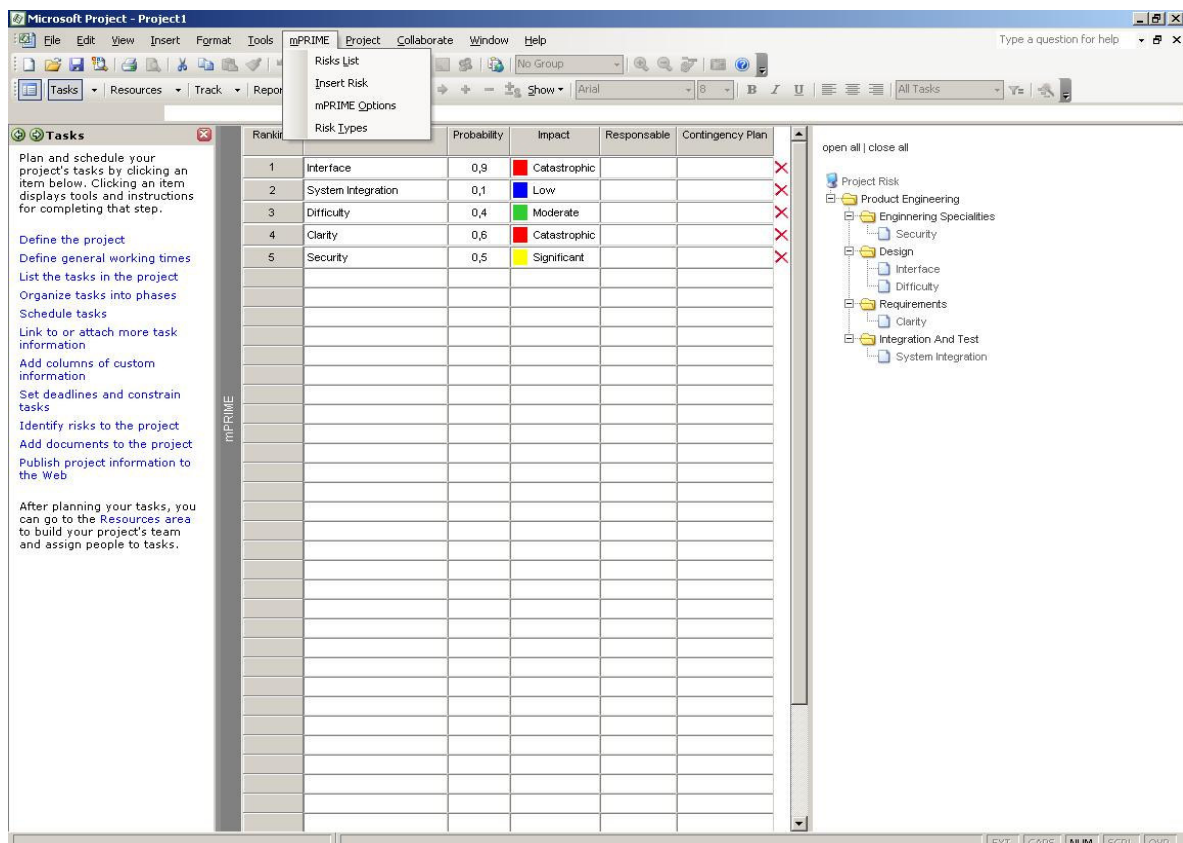


Figura 4 – Visão Geral do mPrime integrado ao ms-project

4.1.1. Principais Funcionalidades

Atualmente, a ferramenta mPRIME encontra-se em uma versão inicial, oferecendo suporte principalmente à atividade de identificação de riscos que se encontra bem mais desenvolvida, possuindo funcionalidades envolvendo técnicas de levantamento de riscos e alguns componentes de inteligência artificial.

Com relação das técnicas de identificação de riscos inseridas na ferramenta, resumem-se à Lista de Verificação que se apresenta de duas formas.

A primeira se trata de uma lista de verificação (checklist), veja figura 5, que foi definida pelo Instituto de Engenharia de Software (Software Engineering Institute – SEI) [CARR ET AL 1993], depois de respondida é retornada uma lista de possíveis riscos ao projeto, veja Figura 6.

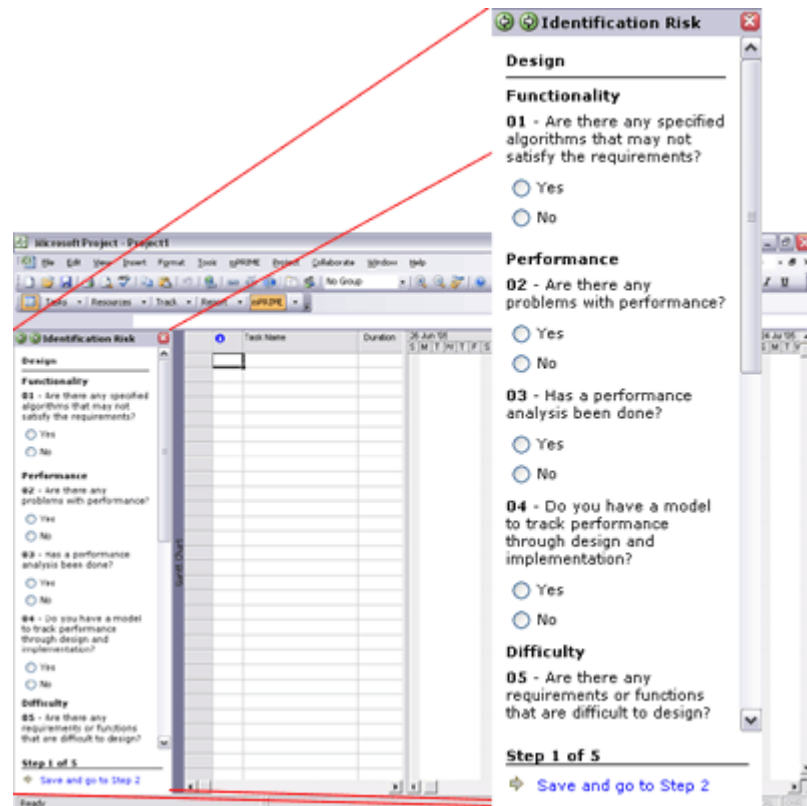


Figura 5 – Visão do Checklist do mPrime

A segunda forma é bastante semelhante à primeira, na verdade é uma derivação da primeira forma da lista de verificação. O principal motivo dessa nova forma se deu principalmente pelo fato da lista definida pelo SEI ser bastante extensa possuindo 194 questões, então de acordo com a disponibilidade de tempo para o usuário responder todo o questionário poderia ser inviável. A solução encontrada, que é a segunda forma da lista de verificação, foi aliar componentes de Inteligência Artificial à Lista, de forma que fossem selecionadas apenas as questões mais relevantes. Esta seleção está baseada na Estrutura Analítica do Projeto – EAP (Work Breakdown Structure – WBS) que está sendo desenvolvido, dessa forma apenas as perguntas mais relevantes seriam

retornadas ao usuário, reduzindo a quantidade de questões da lista de verificação.

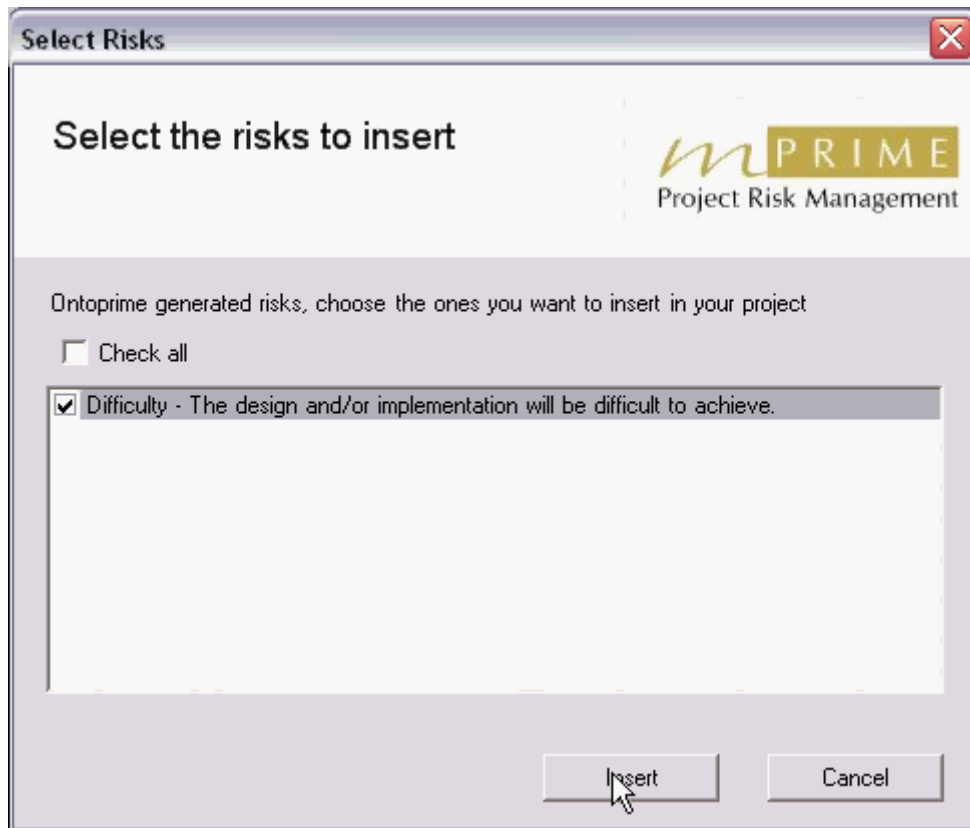


Figura 6 – Exemplo de Riscos Retornados pelos Componentes de Inteligência Artificial

Os componentes existentes, hoje, de inteligência artificial oferecem duas funcionalidades para a identificação de riscos da ferramenta mPRIME. A primeira se trata da leitura da Estrutura Analítica do Projeto, que depois de analisada pela ontologia retorna possíveis riscos do projeto ao usuário. Uma outra possibilidade é a comparação de projetos, ou seja, a ferramenta retorna projetos semelhantes ao que se encontra em desenvolvimento, em seguida o usuário pode selecionar riscos para o

projeto em execução a partir das informações dos riscos de projetos semelhantes.

4.2. Implementação da Métrica

A implementação da métrica teve, resumidamente, duas preocupações, a primeira delas foi estender o mPRIME para que possuísse as informações necessárias para o cálculo da métrica e um segundo passo, a partir dos dados obtidos no primeiro passo calcular para o usuário o resultado da métrica.

Prover informações para o mPRIME foi facilitado, uma vez que os Pontos de Riscos Não Ajustados são classificados a partir da exposição dos Riscos, dado já presente no mPRIME.

Também já se encontrava implementado no mPRIME, o questionário de Fatores Caracterizadores do Projeto, este já servia para a realização de uma funcionalidade de levantamento de riscos através da comparação de projetos.

Nesse primeiro passo, foi necessário fazer uma alteração na parte de lógica da ferramenta para prover o armazenamento das exposições dos riscos, para se ter o acompanhamento dos Pontos de Risco do Projeto ao longo do tempo para se prover informações sobre a eficiência da gestão de riscos.

Project Characteristics

mPRIME
Project Risk Management

Team Size

☒ Very Small 1-6

☐ Medium 21-50

☐ Very Large +100

☐ Small 7-20

☐ Large 51-100

Geographical Team Distribution

☒ Different cities

☐ Same city, same room, different building

☐ Same room

☐ Same city, different enterprise

☐ Same building, different rooms

Cancel < Back Next >

[illegible]

Já a segunda parte da implementação envolveu mudanças na camada lógica da aplicação, para interpretar os dados armazenados a fim de retornar o resultado dos Pontos de Riscos. Em seguida foi implementada a interface gráfica mostrada a seguir com a contagem de Pontos Não Ajustados, por área do projeto, o total de pontos de risco e a relação pontos por ação de resposta aos riscos.



Figura 9 – Resultado da métrica de Pontos de Risco Implementado no mPrime

5. CONCLUSÃO

A gerência de riscos vem ganhando cada vez mais espaço dentro das organizações. Entretanto, por se tratar de uma gerência muito subjetiva, é vital a definição de métricas para auxiliar no processo.

Como a gestão de riscos é bastante recente no desenvolvimento de software existem poucas definições sobre métricas, deixando uma lacuna que muitas vezes pode ser determinante para o fracasso da gerência de riscos ou mesmo de um projeto.

Então na tentativa de diminuir essa falta de métricas, este trabalho propôs uma métrica para o processo de gerência de riscos. Este trabalho procurou definir uma maneira de como quantificar todo risco de um projeto através de um único valor, desta maneira o nível de riscos entre projetos podem ser comparados facilmente.

Logo, foi proposta a Métrica de Pontos de Risco que teve como ponto de partida e inspiração a métrica de Pontos de Casos de Uso, largamente usada em projetos orientados a objetos.

Finalmente, como forma de validação da métrica, esta foi implementada na ferramenta de gerenciamento de riscos, mPRIME.

5.1. Trabalhos Futuros

Existem várias possibilidades de trabalhos futuros a partir desse trabalho. A primeira e mais importante é aplicação da métrica proposta em um projeto real para se avaliar a efetividade dela. Isto se torna bastante facilitado por já se encontrar implementado em uma ferramenta de gerenciamento de riscos.

Outra possibilidade de trabalho é determinar a influência no valor dos Pontos de Riscos Não Ajustados dos Responsáveis pelos Riscos, ou seja, definir uma classificação e pesos que reflitam a influência desse fator de risco, no grau de risco do projeto.

Finalmente, uma outra proposta é a definição de uma nova equação para o Fator Caracterizador do Projeto e então validá-la em uma série de projetos.

REFERÊNCIAS BIBLIOGRÁFICAS

[ANON 1998a] *Emerging Digital Economy*, U.S. Department of Commerce.

[ANON 1998b] *European Information Technology Observatory 98*, European Information Technology Observatory, ISBN 0-94-7486-2.

[BOEHM 1989] BOEHM, B. W., *Tutorial: Software Risk Management*. IEEE Computer Society Press, 1989.

[CARR ET AL 1993] CARR, M. ET AL, *Based Risk Identification. Tecnical report CMU/SEI-93-TR-6*. Software Engineering Institute, Carnegie Mellon University. USA. 1993.

[CHARETTE 1989] CHARETTE, R. N., *Software Engineering Risk Analysis and Management*. McGraw-Hill, New York, 1989.

[COELHO 2003] COELHO, C., *MAPS: UM MODELO DE ADAPTAÇÃO DE PROCESSOS DE SOFTWARE*. Centro de Informática, Universidade Federal de Pernambuco. Recife. Brasil, 2003.

[FAIRLEY 1989] FAIRLEY, R. E., *Risk Management: The Key to Successful Software Projects*. in Proceedings of the 3rd IFAC/IFIP Workshop, F. J. Mowle & P. F. Elzer, eds., Pergamon, Oxford, pp. 45–50, 1989.

[FERNANDES e BUARQUE 2005] FERNANDES, T. e BUARQUE, T., *Manual do Usuário versão 2.0*. In Suppera Solutions Relatório Técnico, Centro de Informática, Universidade Federal de Pernambuco. Recife. Brasil, 2005.

[FREIRE 2006] FREIRE, H., *Calculando Estimativas: O Método de Pontos de Casos de Uso*. Consultada em 05/02/2006, disponível em <<http://www.cnnt.com.br>>.

[GUSMAO e MOURA 2005] GUSMÃO, C.M.G, MOURA, H. *Managing Risk: Methods for Software Systems Development*, Gestão de Riscos para Ambientes de Múltiplos Projetos de Software: Teoria e Prática. In Escola Regional de Minas Gerais – IV ERI MG 2005. Instituto de Informática Pontifícia Universidade Católica de Minas Gerais – PUCMinas.

[GUSMÃO ET AL 2005] GUSMÃO, C.M.G, ET AL, *Ontologia de Domínio de Riscos*. In Suppera Solutions Relatório Técnico, Centro de Informática, Universidade Federal de Pernambuco. Recife. Brasil, 2005.

[HALL 1998] HALL, E. M., *Managing Risk: Methods for Software Systems Development*, Addison–Wesley Pub Co., Reading, 1998.

[JACOBSON 1992] JACOBSON, I., et. al, *Object-Oriented Software Engineering: A Use Case Driven Approach* . Addison-Wesley, Estados Unidos, 1992.

[KARNER 1993] KARNER, GUSTAV, *Resource Estimation for Objectory Projects*. Objective Systems SF AB, 1993, consultada em 05/02/2006, disponível em <<http://www.bfpug.com.br>>.

[LOPES e SOUTO 2005] LOPES, S. e SOUTO, S. *Documento de especificação de projeto versão 1.0*. In Suppera Solutions Relatório Técnico, Centro de Informática, Universidade Federal de Pernambuco. Recife. Brasil, 2005.

[MILLS ET AL 1988] MILLS, E. ET AL, *SEI Curriculum Module SEI-CM-12-1.1*. Software Engineering Institute, Carnegie Mellon University. USA. 1988.

[MYERSON] MYERSON, M., *Risk Management Processes for Software Engineering Models*. Artech House, 1996.

[PMI 2000] PROJECT MANAGEMENT INSTITUTE, *A Guide to the Project Management Body of Knowledge*. 2000, consultada em 05/02/2006, disponível em <<http://www.pmi.org/pmi/publictn/pmboktoc.htm>>.

[PRESSMAN 1995] PRESSMAN, ROGER S., *Software Engineering - A practitioner's aproach*. McGraw Hill, Estados Unidos, 4ª Edição, 1995.

[ROBERTS 1979] ROBERTS, FRED S., *Measurement Theory with Applications to Decision making, Utility, and the Social Sciences*. Encyclopedia of Mathematics and its Applications Addison Wesley Publishing Company, 1979.

[STANDISH 1994] *The CHAOS Report*. Standish Group, Estados Unidos, 1994, consultada em 05/02/2006, disponível em <<http://www.standishgroup.com>>.

