# UNIVERSIDADE FEDERAL DE ITAJUBÁ

**Djalma Brighenti Jr**

# Proposta para o gerenciamento de risco em projetos de desenvolvimento de *software* – Pesquisa-ação na Ford Motor Company Brasil.

Dissertação submetida ao Programa de Pós-Graduação em Engenharia de Produção como requisito parcial à obtenção do título de ***Mestre em Engenharia de Produção***

**Orientador**: Prof. Carlos Eduardo Sanches da Silva, Dr.

**Itajubá, abril de 2005**

BRIGHENTI JR., Djalma. *Proposta para o gerenciamento de risco em projetos de desenvolvimento de software – Pesquisa-ação na Ford Motor Company Brasil.* Itajubá: UNIFEI, 2005. 155p. (Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Engenharia de Itajubá).

Palavras-chave: Gerenciamento de projetos, Gerenciamento de Riscos.

# UNIVERSIDADE FEDERAL DE ITAJUBÁ

**Djalma Brighenti Jr**

Proposta para o gerenciamento de risco em projetos de desenvolvimento de *software* – Pesquisa-ação na Ford Motor Company Brasil.

Dissertação aprovada por banca examinadora em 28 de abril de 2005, conferindo ao autor o título de ***Mestre em Engenharia de Produção.***

**Banca Examinadora:**

Carlos Henrique Pereira Mello, Dr.

Prof. João Bosco Schumann Cunha, Dr.

Prof. Carlos Eduardo Sanches Silva, Dr

**Itajubá, abril de 2005**

Ministério da Educação
**UNIVERSIDADE FEDERAL DE ITAJUBÁ**
Criada pela Lei nº 10.435, de 24 de abril de 2002

# ANEXO I

## PRONUNCIAMENTO DA BANCA EXAMINADORA

A Banca Examinadora, abaixo assinada, nomeada pela Portaria nº 213, de 27 de Abril de 2005, considerando o resultado do Julgamento da Prova de Defesa Pública da Dissertação de Mestrado intitulada: **"Proposta para o gerenciamento de risco em projetos de desenvolvimento de software – Pesquisa-ação na Ford Motor Company Brasil"** apresenta pronunciamento no sentido de que o Coordenador do Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Itajubá solicite ao DRA (Departamento de Registro Acadêmico) a expedição do título de **Mestre em Ciências em Engenharia de Produção**, na área de Concentração **Qualidade Processos e Produto,** satisfeitas as demais exigências regimentais, a **Djalma Brighenti Junior** .

Itajubá, 28 de Abril de 2005.

Dr. Carlos Henrique Pereira Mello
1º Examinador

Prof. Dr. João Bosco Schumann Cunha
2º Examinador - UNIFEI

Prof. Dr. Carlos Eduardo Sanches da Silva
3º Examinador – UNIFEI (Orientador)

# Agradecimentos

Aos colegas da Ford que participaram diretamente desta pesquisa ou indiretamente através de todo o conhecimento que me foi transmitido nestes anos de trabalho em conjunto: muito obrigado !

Ao Prof. Carlos Eduardo, por ter sabido pressionar na hora certa e por estar presente nos momentos necessários: muito obrigado !

Aos meus filhos Daniela e Caio, que ao seu modo souberam entender porque o papai não estava por perto ou disponível para brincar, mesmo em finais de semana ou feriados: muito obrigado !

À minha esposa Claudia, Mestre em Ciências e Doutoranda, por ter me feito trilhar este caminho, por seu incentivo, ajuda e experiência que fizeram dela minha co-orientadora: muito obrigado !

Aos meus pais Sylvia e Djalma, que com sua admiração e orgulho me fizeram seguir este caminho com mais energia e dedicação, e que com seu entusiasmo me contagiaram: um muito obrigado é pouco para expressar todo meu sentimento por vocês.

# Sumário

# Resumo

Projetos de Tecnologia da Informação são normalmente lembrados pelos clientes e usuários como projetos que dificilmente são concluídos no prazo inicialmente previsto, dentro do custo estimado e principalmente com todas as necessidades esperadas atingidas. Existem uma série de fatores que tornam os projetos de TI especialmente difíceis de gerenciar e de ter seus objetivos atendidos. Os projetos de TI envolvem novas tecnologias e uma série de fatores de risco que se repetem. Assim, este trabalho teve o objetivo de propor um processo estruturado para planejamento e controle preventivo de problemas através do gerenciamento dos riscos inerentes aos projetos de tecnologia da informação para diminuir o stress e esforço necessário para os times de projeto concluírem os trabalhos no tempo, custo e com a qualidade esperada. A pesquisa foi realizada na Ford Motor Company Brasil, que possui uma área de Tecnologia da Informação de grande porte e que trabalha rotineiramente com o desenvolvimento de novos projetos de desenvolvimento de sistemas. Para a análise do ambiente atual e o desenvolvimento do modelo de gerenciamento de riscos adaptado à empresa, foi utilizada a técnica de pesquisa *Soft Systems Methodology*. O modelo desenvolvido foi apresentado e validado pela diretoria da área de TI e teve sua implantação recomendada. O Gerenciamento de riscos provou ser uma necessidade para os projetos de desenvolvimento de sistemas devida à complexidade do ambiente em que os projetos são desenvolvidos. A partir de técnicas simples para a identificação dos riscos o seu controle ocorre de forma natural, e a revisão destes riscos nas reuniões de acompanhamento do projeto foca a gerência nos itens mais importantes e onde ela pode efetivamente auxiliar.

Palavras-chave: Gerenciamento de projetos, Gerenciamento de Riscos.

# Abstract

Information technology projects are usually know by their customers and users as projects that seldom are completed on time, within the budget and with all the required specifications. There are several factors that make IT project especially difficult to manage and to meet their goals. IT projects require new technologies and deals with several common risk factors. The objective of this work is to propose a structured process for planning and preventative control of problems through IT project risk management, in order to reduce the project team stress and workload required to deliver the system on time, within budget and with the expected quality. The research was conducted in Ford Motor Company Brazil, which has a large IT department and that is used to work with new system development projects. To understand the current scenario and to develop the risk management model adequate to the company, it was used the research technique Soft Systems Methodology. The resulting model was presented to and validated by the IT department directory and its implementation was recommended. Risk management proved to be a requirement for systems development projects due to the complexity of the environment in which projects are developed. Using simple techniques for risk identification its control is almost natural and the risk review during project status meeting drives management focus to the critical items where they can help.



Key word: Project management, Risk management.

# Lista de figuras

# Lista de quadros

# Lista de tabelas

# Lista de gráficos

# Lista de abreviaturas e siglas

| | |
|---|---|
| 5W 1H | *What, Why, Who, Where, When, How* |
| CMMi | *Configuration management* (gerenciamento de configurações) |
| CMMi | *Capability Maturity Model Integration* |
| COTS | *Commercial off the shelf* (pacote de software comprado) |
| CRM | *Continuous Risk Management* |
| L(UO) | *Loss of unsatisfactory outcome* (perda advinda do resultado insatisfatório) |
| MSP | *Mitigation Strategy Planning* |
| NDS | *Non-developmental software* (software não desenvolvido pelo projeto) |
| P(UO) | *Probability of unsatisfactory outcome* (probabilidade do resultado insatisfatório) |
| PDCA | *Plan, Do, Check , Act* |
| PMBoK | *Project Management Body of Knowledge* |
| PMI | *Project Management Institute* |
| RE | *Risk Exposure* (exposição ao risco) |
| ROI | *Return on Investment* (retorno sobre o investimento) |
| SA-CMM | *Software Acquisition Capability Maturity Model* |
| SDM | *Solution Delivery Methodology* |
| SEI | *Software Engineering* Institute |
| SG | *Specific Goals* |
| SOW | *Statement of Work* (especificação do trabalho) |
| SP | *Specific Practices* |
| SRE | *Software Risk Evaluation* |
| SSM | *Soft Systems Methodology* |
| SW-CMM | *Software Capability Maturity Model* |
| TBQ | *Taxonomy based questionnaire* |
| TI | Tecnologia da Informação |
| TRM | *Team Risk Management* |
| WBS | *Work Breakdown Structure* |

# 1 Introdução

## 1.1 Contexto da pesquisa

De acordo com as estimativas do Standish Group (1999), no ano 2000, cerca de 100 milhões de transações seriam feitas pelas pessoas em todo o planeta a cada segundo. Naquela época o mundo dos negócios já teria mais de um milhão de aplicações críticas para o negócio implantadas. E a cada cinco anos este número de aplicações dobraria: mais transações, mais aplicações e mais interfaces, elevando o número para mais de dois milhões de aplicações críticas no ano de 2004.

Somente nos Estados Unidos seriam iniciados cerca de 300.000 projetos de desenvolvimento de *software* e mais de meio milhão seriam completados nos próximos 12 meses (STANDISH GROUP, 2001).

Projetos de Tecnologia da Informação são normalmente lembrados pelos clientes e usuários como projetos que dificilmente são concluídos no prazo inicialmente previsto, dentro do custo estimado e principalmente com todas as necessidades esperadas atingidas.

Pesquisas realizadas pelo Standish Group (1995) apontaram que nos Estados Unidos, em 1994, 31% dos projetos de desenvolvimento de software seriam cancelados antes de serem completados e cerca de 53% dos projetos iriam custar 189% além da estimativa original.

As estatísticas de 2002 (STANSDISH GROUP, 2003) apesar de terem evoluído em relação a 1994 ainda mostram um cenário bastante negativo, com uma taxa de sucesso de apenas 34% dos projetos. O percentual de projetos cancelados reduziu para 15% e os demais 51% dos projetos tiveram estouro de orçamento (acima de 43% da estimativa inicial) ou perda de prazo (aumento de 82% no prazo original) ou entrega de menos recursos e funções do que o previsto no requisito do projeto (52% das funções iniciais).

Ainda de acordo com o Standish Group (1999), muitos destes projetos irão falhar não por falta de dinheiro ou de tecnologia, mas por falta de habilidades em gerenciamento de projetos.

Parecem, portanto, existir uma série de fatores que tornam os projetos de TI especialmente difíceis de gerenciar e de ter seus objetivos atendidos. Os projetos de TI envolvem novas tecnologias e uma série de fatores de risco que se repetem. Estes projetos demandam uma energia muito grande de todo o time do projeto para poder atender os compromissos assumidos. Desta forma torna-se necessário o entendimento das peculiaridades dos projetos de TI e a análise de metodologias que permitam gerenciar melhor os projetos de TI, principalmente no que se concerne ao gerenciamento de riscos.

Surge o problema de pesquisa desta dissertação que pode ser descrito através da pergunta: Quais as abordagens do gerenciamento de riscos concernentes a projetos de TI? Como implementar o gerenciamento de riscos em projetos de TI?

Posterior à analise das abordagens do gerenciamento de riscos concernentes a projetos de TI, seleciona-se uma abordagem e avalia-se sua aplicação na área de TI da Ford Motor Company Brasil e, em especial, na área de desenvolvimento de sistemas que trabalha predominantemente na condução de novos projetos.

## 1.2 Objetivos

Assim, o objetivo principal deste trabalho é propor um processo estruturado para planejamento e controle preventivo de problemas através do gerenciamento dos riscos inerentes aos projetos de tecnologia da informação para diminuir o stress e esforço necessário para os times de projeto concluírem os trabalhos no tempo, custo e com a qualidade esperada.

Dentro do objetivo principal, desdobram-se objetivos específicos nos seguintes temas:

- Descrever e analisar os métodos de gerenciamento de riscos de projetos
- Identificar os que são mais aplicáveis à área de projetos de Tecnologia da Informação
- Analisar o ambiente de projetos de Tecnologia da Informação na Ford e os elementos de risco mais freqüentes
- Propor e avaliar um processo estruturado para gerenciamento destes riscos na Ford

## 1.3 Metodologia da pesquisa

### 1.3.1 Classificação da pesquisa

Conforme discorre Bryman (1989), na pesquisa-ação o investigador torna-se virtualmente parte da arena sendo estudada com o propósito de resolver problemas organizacionais e a pesquisa-ação é explicitamente preocupada com o desenvolvimento de soluções que podem ser aplicadas na organização.

Yin (1984) apresenta três condições para escolha do método de pesquisa a ser adotado: tipo de questão colocada; grau de controle que o pesquisador tem sobre os eventos; grau de focalização no contemporâneo como oposição a eventos históricos. No caso desta pesquisa as condições propostas por Yin (1984) classificam a pesquisa como pesquisa-ação, pois:

- Tipo de questão: como avaliar os riscos presentes no desenvolvimento de projetos de TI ?
- Grau de controle do pesquisador tem sobre os eventos: participante ativo da pesquisa
- Grau de focalização no contemporâneo como oposição a eventos históricos: a pesquisa tem como objeto um tema atual e presente nos projetos de TI.

Complementando as propostas de Yin (1984) pode-se considerar também a limitação de tempo e a necessidade de validar o método proposto a partir de sua aplicação, assim a pesquisa-ação destaca-se como o método de pesquisa mais adequado.

Para o desenvolvimento da pesquisa-ação foi definida a utilização da Metodologia SSM (*Soft Systems Methodology*), pois esta é uma técnica que permite planejar e implementar mudanças, embora também seja usada para desenvolver novos sistemas.

O quadro 1 classifica a pesquisa a ser realizada nesta dissertação de diferentes maneiras, utilizando-se dos conceitos de Silva e Menezes (2001).

| Classificação da pesquisa | Conceito | Justificativa para a classificação da pesquisa |
|---|---|---|
| Quanto à natureza da pesquisa:<br>• Pesquisa Aplicada | A pesquisa aplicada objetiva gerar conhecimentos para aplicação prática dirigidos à solução de problemas específicos, envolvendo verdades e interesses locais. | A dissertação contribui para que a área de desenvolvimento de sistemas da Ford utilize os conceitos descritos para um melhor gerenciamento dos projetos de TI. |
| Quanto à predominância na forma de abordagem do problema:<br>• Pesquisa Qualitativa | A pesquisa qualitativa considera que existe uma relação dinâmica entre o mundo real e o sujeito, onde o ambiente natural é a fonte direta para coleta de dados e o pesquisador é o instrumento chave. Utiliza-se da interpretação dos fenômenos e da atribuição de significados. | O trabalho analisa o ambiente de desenvolvimento de projetos de TI da Ford e as principais preocupações enfrentadas pelos líderes de projeto (elementos de risco). |
| Quanto aos seus objetivos:<br>• Pesquisa Exploratória | A pesquisa exploratória visa proporcionar maior familiaridade com o problema com vistas a torná-lo explícito ou a construir hipóteses. Envolve levantamento bibliográfico e entrevista com pessoas que tiveram experiência prática com o problema pesquisado; análise de exemplos que estimulem a compreensão. | Os conceitos iniciais da dissertação estão baseados em levantamentos bibliográficos.<br>A exploração da pesquisa-ação permite aprofundar os conhecimentos acerca do processo em análise |
| Quanto aos seus procedimentos técnicos:<br>• Em forma de Pesquisa-Ação | A Pesquisa-Ação é concebida e realizada em estreita associação com uma ação ou com a resolução de um problema coletivo. Os pesquisadores e participantes representativos da situação ou do problema estão envolvidos de modo cooperativo ou participativo. | A dissertação utiliza-se da pesquisa-ação em uma empresa que tem uma área de desenvolvimento de sistemas com uma prática rotineira de desenvolvimento de projetos de TI. |

**Quadro 1** – Classificação da pesquisa

**Fonte:** Baseado em Silva e Menezes (2001)

### 1.3.2 Universo da pesquisa

O universo da pesquisa será a Ford Motor Company Brasil, uma empresa montadora de automóveis com várias plantas no Brasil, sendo uma delas na região do Grande ABC, em São Paulo. Nesta planta estão concentrados os escritórios administrativos da empresa e também a área de Tecnologia da Informação. Essa empresa possui os requisitos necessários para se elaborar o trabalho, uma vez que a mesma apresenta uma área de Tecnologia da Informação de grande porte e que trabalha rotineiramente com o desenvolvimento de novos projetos de desenvolvimento de sistemas.

### 1.3.3 Instrumentos de coleta de dados

O processo de coleta de dados terá como origem a observação direta do pesquisador, devido à sua experiência como gerente de projetos de TI na empresa universo da pesquisa, e a utilização de entrevistas não-estruturadas para o desenvolvimento da metodologia SSM, com:

- Analistas líderes de projeto
- Executivos da área de TI da empresa com larga experiência na gerência de projetos de TI
- Diretor de TI da empresa.

## 1.4 Limitações

O trabalho se limita ao estudo do gerenciamento de riscos em projetos de TI, em especial os projetos de desenvolvimento de sistemas, bem como propor e avaliar um processo estruturado para gerenciamento destes riscos na Ford.

Sendo assim a pesquisa não aborda a cultura organizacional necessária para a implantação deste modelo e também não se propõe a fazer um estudo completo das estratégias a serem utilizadas para responder a estes riscos.

Devido ao método de pesquisa ser a pesquisa-ação, os resultados não podem ser generalizados.

## 1.5 Estruturas

No capítulo 1 – Introdução – serão apresentados o problema de pesquisa, seu contexto, objetivos, metodologia de pesquisa, limitações e estrutura.

No capítulo 2 – Gerenciamento de projetos e TI – será feita uma breve apresentação dos conceitos de gerenciamento de projetos com base no PMBoK (PMI, 2000) e a contextualização dos projetos de TI, apresentando suas peculiaridades em relação a outros tipos de projetos.

No capítulo 3 – Gerenciamento de riscos – será contextualizado o gerenciamento de riscos e analisado como este processo pode ajudar no gerenciamento de projetos de desenvolvimento de sistemas, sendo apresentados e comentados fatores de riscos presentes na literatura. Também serão apresentadas e discutidas várias metodologias clássicas: PMBoK, Barry Bohem, SEI e CMMi.

No capítulo 4 – Processos do gerenciamento de riscos – serão apresentados em detalhes as várias etapas das metodologias de gerenciamento de riscos, analisando o objetivo de cada etapa, as técnicas e ferramentas utilizadas e o resultado esperado.

No capítulo 5 – Pesquisa-Ação – será apresentada a aplicação da metodologia *Soft Systems* na empresa universo da pesquisa, descrevendo as etapas desenvolvidas, seus resultados e a avaliação do processo proposto para gerenciamento de riscos.

E, finalmente, no capítulo 6 – Conclusões – serão apresentadas as conclusões gerais e específicas deste trabalho, bem como sugestões de trabalhos futuros que possam continuar desenvolvendo conhecimento sobre o tema discorrido nesta dissertação.

No Apêndice A – Identificação de riscos baseados em uma taxonomia – será apresentada o conceito da taxonomia dos problemas de desenvolvimento de sistemas e o método de identificação de riscos utilizando um questionário baseado nesta taxonomia.

No Apêndice B – Definições dos grupos da taxonomia – será apresentada a definição das classes, elementos e atributos da taxonomia.

No Apêndice C – TBQ – Questionário completo – será apresentado o questionário completo utilizado no método de identificação de riscos descrito no Apêndice A.

No Apêndice D – TBQ – Questionário resumido – será apresentado o questionário resumido que poderá ser utilizado em algumas situações na identificação de riscos.

# 2 Gerenciamento de Projetos e TI

Conforme discorre Sisk (1998), o gerenciamento de projetos na sua forma moderna começou a se delinear apenas a algumas décadas. Começando no início de 1960, empresas e outras organizações começaram a enxergar o benefício de organizar o trabalho em projetos e a entender a necessidade crítica de comunicar e integrar o trabalho através de múltiplos departamentos e profissões.

Na virada do século 20, Frederick Taylor iniciou seus estudos detalhados do trabalho. Ele aplicou a lógica científica ao trabalho para mostrar que as tarefas poderiam ser analisadas e melhoradas focando em suas partes elementares. Antes de Taylor, a única forma de melhorar a produtividade era demandar mais horas de trabalho duro dos trabalhadores.

Henry Gantt, sócio de Taylor, estudou em detalhe a ordem das operações do trabalho. Seus gráficos de Gantt, com as barras de tarefas e os marcadores de *milestones*, desenham a seqüência e a duração de todas as tarefas do processo. Estes diagramas provaram ser uma ferramenta analítica para os gerentes tão poderosa que ela permaneceu praticamente inalterada por quase cem anos. Foi somente no início da década de 90 quando linhas foram adicionadas às barras de tarefas para mostrar precisamente as dependências entre tarefas.

Após a segunda guerra mundial, a complexidade dos projetos e a diminuição da mão-de-obra disponível durante o período da guerra demandaram novas estruturas da organização. Complexos diagramas de rede chamados de PERT e o método do caminho crítico foram introduzidos, dando aos gerentes um maior controle sobre projetos extremamente complexos (como sistemas militares de armas, com sua enorme variedade de tarefas e numerosas interações em muitos pontos no tempo).

Nas décadas seguintes, a abordagem para o gerenciamento de projetos começou a tomar sua forma moderna. Enquanto que vários modelos de negócio evoluíram neste período, eles compartilhavam uma mesma estrutura básica: que os projetos são gerenciados por gerentes de projetos, que monta um time e assegura a integração e comunicação do fluxo de trabalho horizontalmente através de diferentes departamentos.

Contando hoje com mais de 90.000 associados o Project Management Institute (PMI), que foi fundado em 1969 na Filadélfia, Pensilvânia, USA, se tornou a principal associação profissional em Gerenciamento de Projetos.

Em 1996, o PMI publicou o PMBoK – A Guide to the Project Managment Body of Knowledge (PMBoK Guide), um guia englobando todas as áreas do conhecimento que regem as regras do gerenciamento de projetos. No ano 2000 foi lançada a segunda versão do PMBOK *Guide* e, no início do século 21, mais de 270.000 cópias do guia estavam em circulação (PMI, 2004).

Uma nova versão do guia está em desenvolvimento e deverá ser chamada de PMBOK 3ª ed. De acordo com Buzin (2003), na nova versão os processos básicos de gerenciamento de projeto (início, planejamento, execução, controle e finalização) são abordados de uma forma mais dinâmica, enfatizando a sobreposição, em um ciclo muito familiar aos que trabalham com a qualidade total e o PDCA – *Plan, Do, Check, Act*.

Devido à sua grande aceitação e a ser uma referência de conhecimentos específicos para todos os profissionais da área de gerenciamento de projetos, este capítulo baseia-se no PMBoK (PMI, 2000).

## 2.1 Definição de projeto e de gerenciamento de projeto

Toda organização desenvolve trabalho e o trabalho pode ser contínuo e repetitivo (operacional) ou temporário e único (projeto). Portanto, "projeto é uma empreitada temporária executada para criar um produto ou serviço único".(PMI, 2000)

Desta forma, dois aspectos caracterizam o projeto:

- temporariedade: todo projeto tem um começo e um fim bem definidos
- unicidade: o produto ou serviço criado pelo projeto é diferente de alguma forma de todos os outros produtos ou serviços

Para muitas organizações, os projetos são uma forma de atender às necessidades que não podem ser resolvidas através da operação normal da organização. E os projetos são críticos para a realização dos objetivos estratégicos das organizações, uma vez que os projetos são a forma como a estratégia é implementada.

Exemplos de projetos podem ser:

- desenvolvimento de um novo produto ou serviço
- execução de uma mudança na estrutura ou estilo da organização
- desenho de um novo veículo de transporte
- desenvolvimento ou aquisição de um novo sistema de informação
- construção de um novo prédio
- condução de uma campanha política
- implementação de um novo procedimento de negócio

Como afirmado na definição de projeto, o projeto é sempre temporário e, portanto, tem um início e término definidos. Ser temporário não significa que o projeto seja curto em duração, pois muitos projetos podem durar por vários anos. O término do projeto é alcançado quando:

- os objetivos do projeto foram alcançados
- torna-se claro que os objetivos do projeto não serão ou não poderão ser atingidos ou
- a necessidade de realizar o projeto não existe mais

É importante diferenciar o projeto em si do produto ou serviço criado pelo projeto. Desta forma, ser temporário normalmente não se aplica ao produto ou serviço criado pelo projeto. Normalmente projetos são executados para criar resultados que perdurem por muito tempo.

Os objetivos de projetos e da operação rotineira da organização são fundamentalmente diferentes. O objetivo do projeto é alcançar seu objetivo e fechar o projeto. O objetivo de uma operação contínua é normalmente sustentar o negócio. Projetos são encerrados quando o seu objetivo esperado é alcançado enquanto que atividades "não-projeto" adotam novos objetivos e continuam a ser executadas.

Projetos envolvem executar alguma coisa que ainda não foi feita anteriormente e, portanto, são únicas. Um produto ou serviço pode ser considerado único mesmo que pertença a uma categoria bastante abrangente. Por exemplo, muitos edifícios foram construídos, porém cada um é único em algum aspecto: desenho, proprietários, localização, contratados, etc. A presença de elementos repetitivos não altera a característica fundamental de singularidade de um trabalho de projeto.

Projetos que integram os conceitos de temporariedade e unicidade também apresentam a característica de elaboração progressiva. Como o produto de cada projeto é único, as características que distinguem um produto ou serviço devem ser elaboradas progressivamente. Progressivamente significa "progredindo em passos, continuando em incrementos" enquanto que elaborado significa "desenvolvido com cuidado e detalhe".

## 2.2 O gerenciamento de projetos

*Gerenciamento de projetos é a aplicação de conhecimentos, habilidades, ferramentas e técnicas em atividades do projeto para alcançar os requisitos do projeto* (PMI, 2000).

O gerenciamento de projetos é realizado através do uso de processos como: iniciação, planejamento, execução e controle e fechamento. O time do projeto gerencia os trabalhos do projeto e este trabalho envolve tipicamente:

- demandas competindo por: escopo, tempo, custo , risco e qualidade
- pessoas chave com diferentes necessidades e expectativas
- requisitos identificados

É importante notar que muitos dos processos do gerenciamento de projetos são iterativos por natureza. Isto se deve em parte à existência e necessidade de elaboração progressiva através do ciclo de vida do projeto, ou seja, quanto mais você sabe sobre o projeto, melhor preparado você está para gerenciar o projeto.

## 2.3 Processos de gerenciamento de projetos

O gerenciamento de projetos é uma empreitada interativa. Essas interações normalmente requerem soluções de compromisso entre os objetivos do projeto: melhoria em uma área pode ser alcançada apenas pelo sacrifício da performance de outra área. O sucesso no gerenciamento de projetos exige um gerenciamento ativo destas interações.

A tensão devido à competição entre as demandas por performance (funcionalidades, qualidade, especificações do produto), custo e prazo é apresentada por Williams (1999) através do conceito do "triângulo de ferro" do gerenciamento de projetos (figura 1).

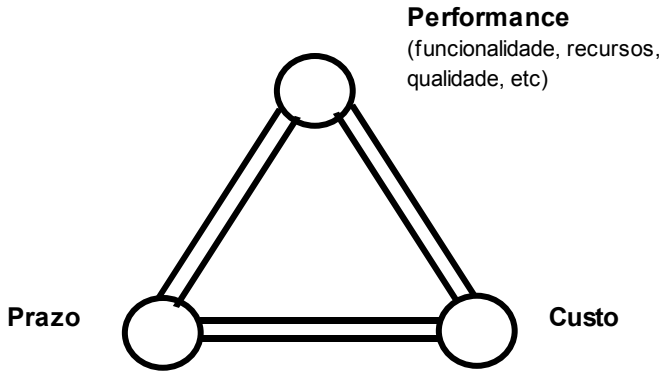


**Figura 1 –** Triângulo do gerenciamento de projetos
**Fonte:** Willians (1999)

Neste conceito, todo projeto deve satisfazer as restrições dos três vértices do triângulo simultaneamente. Porém, o gerenciamento responsável do projeto requer um entendimento de quais restrições são mais importantes para o projeto quando não é possível atender a todas ao mesmo tempo.

Projetos são compostos de processos. De acordo com Humprey (1999), um processo de sofwtare é um conjunto de fases de um projeto, estágios, métodos, técnicas e práticas que são empregadas para desenvolver e manter o software e os artefatos associados a ele, como planos, documentos, modelos, código, casos de teste, manuais, etc. Os processos do projeto são executados por pessoas e normalmente enquadram-se em uma das duas categorias de processos:

- de gerenciamento de projetos: descrever, organizar e completar as tarefas do projeto. Os processos de gerenciamento de projetos são aplicáveis à maioria dos projetos, na maior parte do tempo e serão descritos brevemente a seguir.
- orientados ao produto: especificar e criar o produto do projeto. Processos orientados ao produto são tipicamente definidos pelo ciclo de vida do projeto.

Processos de gerenciamento de projeto e processos orientados ao produto possuem intercessões e interagem ao longo do projeto. Por exemplo, o escopo do projeto não pode ser definido na ausência de um conhecimento básico de como criar o produto.

Os processos de gerenciamento de projetos, segundo o PMBoK (2000), podem ser agrupados em cinco grupos, que são processos de:

- Iniciação: dar autorização para o projeto ou a fase do projeto
- Planejamento*:* definir e redefinir objetivos e selecionar a melhor alternativa de ação para atingir o objetivo que o projeto se propõe
- Execução: coordenar pessoas e outros recursos para desenvolver o plano
- Controle: assegurar que os objetivos do projeto serão atingidos monitorando e medindo regularmente o progresso e identificando variações em relação ao plano de modo a que medidas corretivas possam ser tomadas quando necessário
- Fechamento: formalizar a aceitação do projeto ou fase e concluí-lo de uma forma ordenada

Os grupos de processos são ligados pelos resultados que eles produzem: o resultado de um processo freqüentemente serve de input para outro processo. Entre os grupos de processo centrais os elos são iterativos: o planejamento provê um plano documentado para a execução que, por sua vez, através do controle provê atualizações para o plano inicial. Estas conexões podem ser vistas na figura 2.

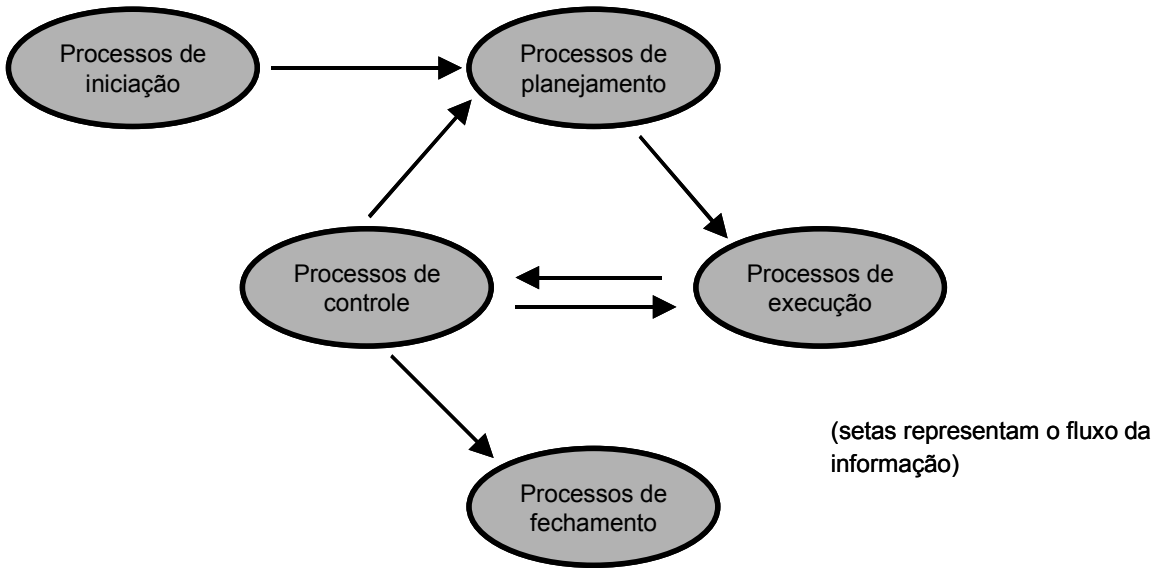


**Figura 2** – Conexões entre os processos

**Fonte**: PMBoK (2000, p.31)

Além disto, os grupos de processos não são eventos isolados e que ocorrem uma única vez: são atividades que tem interseção e sua intensidade varia através de cada fase do projeto, como mostra a figura 3.

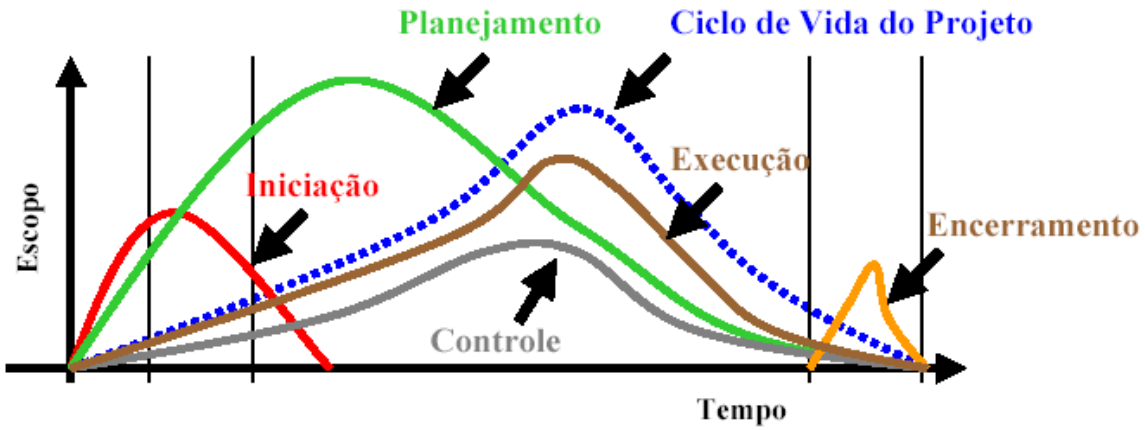


**Figura 3** – Interseções entre os processos em uma fase
**Fonte**: PMBoK (2000, p.31)

Finalmente, os grupos de processos também interagem através das fases de tal forma que o fechamento de uma fase provê inputs para o início de outra fase. Por exemplo, o fechamento de uma fase de *design* requer o aceite do cliente no documento de *design*. Ao mesmo tempo, o documento de *design* define a descrição do produto para a fase seguinte.

Segundo o PMBoK (2000), dentro de cada grupo de processos, os processos individuais são ligados por suas entradas (*inputs*) e saídas (*outputs*). Focando nestas ligações, cada processo pode ser descrito em termo de suas:

- Entradas: documentos ou itens documentáveis que serão trabalhados
- Ferramentas e técnicas: mecanismos aplicados às entradas para gerar as saídas
- Saídas: documentos ou itens documentáveis que são o resultado de um processo

## 2.4 Disciplinas do gerenciamento de projetos

As disciplinas do gerenciamento de projetos descrevem conhecimentos e práticas do gerenciamento de projetos em termos dos componentes de seus processos. Estes processos foram organizados pelo PMBoK (PMI, 2000) em nove áreas de conhecimento denominadas gerência:

- da integração do projeto: descreve os processos requeridos para garantir que os vários elementos do projeto são coordenados apropriadamente. Consiste em desenvolvimento do plano do projeto, execução do plano do projeto e controle integrado de mudanças.
- do escopo do projeto: descreve os processos requeridos para assegurar que o projeto inclui todo o trabalho necessário e apenas o trabalho necessário para completar o projeto com sucesso. Consiste em iniciação, planejamento do escopo, definição do escopo, verificação do escopo e controle de mudanças do escopo.
- do prazo do projeto: descreve os processos requeridos para assegurar a execução do projeto no prazo previsto. Consiste em definição de atividades, seqüenciamento de atividades, estimativa da duração das atividades, desenvolvimento do cronograma e controle do cronograma.
- do custo do projeto*:* descreve os processos requeridos para assegurar que o projeto é completado dentro do orçamento aprovado. Consiste em planejamento de recursos, estimativa de custos, orçamento de custos e controle de custos.
- da qualidade do projeto: descreve os processos requeridos para garantir que o projeto atenda os requisitos para o qual foi aprovado. Consiste em planejamento da qualidade, garantia da qualidade e controle da qualidade.
- dos recursos humanos do projeto: descreve os processos requeridos para fazer o uso mais efetivo das pessoas envolvidas no projeto. Consiste em planejamento da organização, aquisição de pessoas e desenvolvimento do time.
- da comunicação do projeto: descreve os processos requeridos para assegurar a geração, coleta, divulgação, armazenamento e disposição apropriada e em tempo das informações do projeto. Consiste em plano de comunicação, distribuição da informação, relatório de performance e fechamento administrativo.
- de riscos do projeto: descreve os processos preocupados com a identificação, análise e resposta aos riscos do projeto. Consiste em planejamento do gerenciamento dos riscos, identificação dos riscos, análise qualitativa dos riscos, análise quantitativa dos riscos, plano de reposta aos riscos e controle e monitoramento dos riscos.
- da aquisição em projetos: descreve os processos requeridos para adquirir bens e serviços de empresas externas à organização que está executando o projeto. Consiste em planejamento da aquisição, planejamento da solicitação, solicitação, seleção de fornecedores, administração de contratos e fechamento de contratos.

## 2.5 Contexto da TI

Conforme Lientz e Rea (2001), de 30 a 45% dos projeto de tecnologia da informação (TI) falham antes de terminarem. Mais de metade dos projetos de sistemas ultrapassam o prazo e orçamento em mais de 200%. As falhas e problemas continuam, apesar das melhorias em ferramentas e tecnologia. Os dados também indicam que os projetos que falharam eram vistos como criticamente importantes pela gerência.

Pesquisas realizadas pelo Standish Group (1995) apontaram que somente os Estados Unidos, em 1994, estavam gastando mais de $250 bilhões de dólares por ano no desenvolvimento de novas aplicações de TI, divididas em aproximadamente 175.000 projetos. Segundo a pesquisa 31% destes projetos seriam cancelados antes de serem completados e cerca de 53% dos projetos iriam custar 189% além da estimativa original.

Baseado nestes dados, o Standish Group estimou que em 1995 as companhias americanas e as agências governamentais dos Estados Unidos iriam gastar cerca de $ 81 bilhões de dólares em projetos de *software* que seriam cancelados. Estas mesmas organizações iriam pagar um adicional de $59 bilhões de dólares em projetos de *software* que seriam completados, porém iriam exceder as estimativas originais de tempo e custo.

Em 1998, estes números foram estimados em $275 bilhões de dólares de investimentos em cerca de 200.000 projetos de desenvolvimento de *software* (STANDISH GROUP, 1999). Em 2000, foi estimado que cerca de 300.000 projetos seriam iniciados apenas naquele ano e mais de meio milhão seriam completados nos próximos 12 meses (STANDISH GROUP, 2001).

*Muitos destes projetos irão falhar, mas não por falta de dinheiro ou tecnologia; a maioria irá falhar por falta de habilidades em gerenciamento de projetos* (STANDISH GROUP, 1999)

A pesquisa dividiu os projetos em três categorias de resolução:

- Projetos de sucesso: projetos que foram completados dentro do prazo e orçamento previsto e com todos os recursos e funções especificados inicialmente
- Projetos desafiados: projetos que foram completados e estão operacionais, porém acima do orçamento, acima da estimativa de prazo e com menos recursos e funções que as originalmente especificadas

- Projetos fracassados: projetos que foram cancelados em algum ponto do ciclo de desenvolvimento

O Standish Group continua a repetir esta pesquisa a cada dois anos e os resultados tem mostrado uma melhoria contínua e significativa na taxa de sucesso dos projetos e também na diminuição dos estouros de orçamento. Porém, os números de insucesso ainda continuam bastante grandes. O gráfico 1 mostra a evolução das estatísticas de 1994 a 2002.

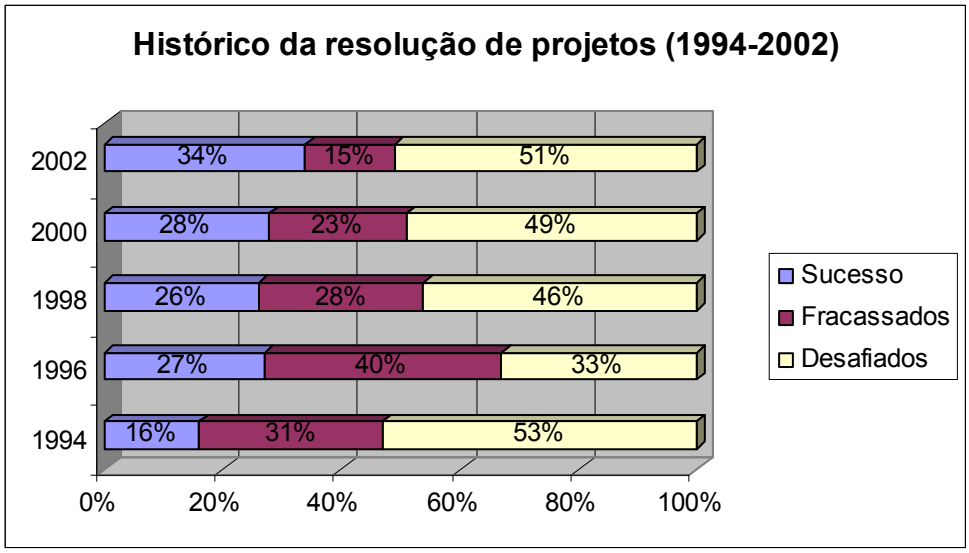


**Gráfico 1** – Resultados da pesquisa do Standish Group

**Fonte**: adaptado de Standish Group (2001)

Como se pode ver, a taxa de sucesso aumentou para 34%, representando um aumento de 100% em relação à taxa de 16% reportada em 1994. As falhas nos projetos caíram para 15% o que é menos da metade da taxa de 31% em 1994. Os projetos desafiados ficaram com os restantes 51%, praticamente a mesma taxa desde 1994.

Porém, ainda assim, os projetos tiveram um estouro de orçamento de 43% da estimativa inicial, um estouro de prazo de 82% e uma entrega dos recursos e funções previstas no requisito inicial de apenas 52%.

Porque muitos projetos de TI falham? Complexidade é parte da resposta. Além disto as pessoas são envolvidas pelo trabalho do dia a dia e a gerência e o negócio dependem cada vez mais da tecnologia. Novas tecnologias não devem ser apenas implementadas como devem

também estar integradas. Os níveis dos padrões e as expectativas foram elevados. (LIENTZ e REA, 2001).

Muitas são as oportunidades para a falha de projetos de software, de acordo com o Standish Group (1999), são:

- Esforços de desenvolvimento de software são conduzidos atualmente em ambientes de TI complexos e distribuídos.
- Os desenvolvimentos ocorrem em uma frágil matriz de aplicações, usuários, demandas dos consumidores, legislações, políticas internas, orçamento e dependências organizacionais que estão mudando constantemente.
- Os gerentes de projeto que não dominam ferramentas de planejamento, controle e monitoramento de multi-projetos que abrangem toda a organização, freqüentemente acham impossível compreender o sistema como um todo.
- Subestimar a complexidade do projeto e ignorar requisitos que estão mudando são razões básicas para a falha de projetos.

E mais ainda, o software hoje deve não apenas automatizar processos. Ele deve criar valor para o negócio ao melhorar o serviço ao cliente ou desenvolver vantagem competitiva. O retorno sobre o investimento (ROI) aumenta a expectativa no projeto: o software deve ter um resultado mensurável no resultado final da companhia.

*Por todas estas razões, a motivação para a adoção das disciplinas de gerenciamento de projeto para o desenvolvimento de software é óbvia. Simplesmente não há outra maneira, porém sua implementação é incerta.* (STANDISH GROUP, 1999)

O gerenciamento de projetos é um processo que abrange todo o ciclo de vida do projeto, desde a sua concepção até a finalização. Seus fundamentos são o planejamento, execução e controle de todos os recursos, tarefas e atividades necessárias para completar o projeto. O Gerenciamento de projeto não é uma atividade isolada, mas um esforço de time. No final, o gerenciamento de projetos é baseado em pessoas e processos – em como o trabalho está sendo conduzido.

De acordo com o relatório de 1999, o gerenciamento de projetos está ganhando corpo nas organizações de TI e os resultados são encorajadores. Taxas de falha estão caindo, os custos estão caindo e as taxas de sucesso estão subindo. Grandes companhias estão adotando uma abordagem de minimização no gerenciamento de projetos. Minimização significa diminuir as funções e recursos junto com a redução do escopo. Organizações de TI estão adotando metodologias padrão de projetos e desenvolvendo disciplinais formais de gerenciamento de projetos.

*Software nunca foi tão complexo e tão caro quando no ambiente atual* (BOEHM, 1991).

Lientz e Rea (2001) entende que falhas também acontecem porque as pessoas gerenciam projetos de TI da mesma forma que outros projetos. Ele defende que projetos de TI são diferentes:

- A duração do projeto pode ser longa. Durante este tempo a tecnologia evolui e pode afetar o projeto. Os requisitos do negócio podem alterar.
- Tipicamente projetos de tecnologia não saem do zero, eles devem se integrar com os sistemas e tecnologias atuais.
- Como parte do projeto, os membros do time do projeto podem ter que aprender a nova tecnologia ao mesmo tempo em que conduzem o projeto.

*Estas características são muito diferentes daquelas encontradas quando pontes são construídas, novos produtos são lançados e outros projetos mais comuns* (LIENTZ e REA, 2001).

Lientz e Rea (2001) lista as seguintes diferenças entre projetos tradicionais e os projetos de TI:

- Objetivo: os objetivos de um projeto de tecnologia e sistemas normalmente não são tão claramente definidos como os de engenharia ou outros projetos. Os objetivos podem não estar bem definidos no início do projeto de sistemas.
- Escopo: projetos de sistemas algumas vezes sofrem da falta de definição clara de suas fronteiras. O processo de negócio faz parte do projeto? Com quais sistemas deve o projeto fazer interface? E muito mais, o escopo pode mudar e expandir.

- Trabalho paralelo: enquanto o trabalho no novo sistema está sendo criado ou instalado, o trabalho continua no sistema atual, criando mudanças nos requisitos. Isto normalmente não ocorre em projetos tradicionais.
- Interface com outros projetos: projetos de sistemas normalmente têm uma gama complexa de interfaces.
- Dependência da tecnologia: parece que apenas em projetos de sistemas e de tecnologia as pessoas tentam utilizar novas tecnologias ou tecnologias com as quais elas têm nenhuma ou muito pouca experiência, aumentando assim o nível de risco do projeto.
- Expectativa da gerência: a alta administração participa de seminários e lê sobre a promessa de novas tecnologias. Suas expectativas podem impactar o projeto do sistema.
- Impacto cumulativo: um projeto pode afetar vários outros. O último projeto depende do resultado de vários anteriores e de alguns esforços correntes. É uma dependência cumulativa.
- Entendimento da tecnologia: apesar de que projetos "não TI" podem envolver tecnologia, sua aplicação é normalmente mais simples porque a tecnologia pode ser manejada separadamente. Em sistemas é o contrário. O único modo de sistemas modernos terem sucesso é integrando múltiplas tecnologias. Isto requer um conhecimento mais completo e profundo da tecnologia.
- Hiatos de tecnologia: projetos de sistemas e tecnologia podem também ser afetados por hiatos entre a mais nova tecnologia e as tecnologias antigas.

Também de acordo com Lientz e Rea (2001), novas tendências em negócios, tecnologia e sistemas influenciam as tendências em gerenciamento de projetos de TI. De uma maneira geral ele cita os seguintes impactos:

- Projetos envolvem mais aspectos de negócio devido à reengenharia, melhoria de processos e *e-business*
- Projetos são mais complexos em geral em termos de tecnologia
- Projetos são mais integrados uns com os outros e com o trabalho atual
- Membros da organização são designados para múltiplos projetos devido a limites de recursos e orçamento. Além disto, há disputa pelo tempo das pessoas entre os vários projetos e o trabalho operacional como suporte à rede, manutenção, melhorias e suporte à operação.

# 3 Gerenciamento de riscos

## 3.1 Definições e objetivos

A análise feita por Boehm (1991) do fracasso de vários projetos de *software* indicou que os seus **problemas poderiam ter sido evitados** ou fortemente reduzidos se tivesse havido uma **preocupação** clara **com a identificação antecipada e a resolução** de seus elementos de alto **risco**. Freqüentemente, estes projetos são envolvidos em um ambiente de otimismo durante suas fases iniciais que acabam ignorando alguns sinais claros de problemas de alto risco e que acabam provando ser a sua causa de fracasso no final.

*Um padrão que emergiu da análise de vários gerentes de projeto com diversas metodologias de desenvolvimento de software, foi que os* ***gerentes de projeto de sucesso*** *eram* ***bons gerentes de risco*** (BOEHM, 1991).

Apesar deles normalmente não utilizarem termos como identificação de riscos, análise de riscos, planejamento de riscos ou monitoração de riscos, eles utilizavam um conceito geral de exposição ao risco para guiar suas ações e prioridades. E seus projetos tendiam a evitar falhas e produzir bons produtos.

De acordo com Van Scoy (1992), se outras atividades da engenharia são capazes de controlar os seus riscos técnicos, então a engenharia de *software* também deveria ser capaz de fazer o mesmo.

A habilidade para gerenciar incertezas em projetos é uma necessidade para lidar com recursos escassos, avanços na tecnologia e com o aumento da demanda de sistemas complexos em um ambiente de rápidas mudanças. Para responder a essas necessidades, os métodos de gerenciamento de riscos foram adaptados para uso pelos gerentes e engenheiros de sistemas (HALL, 1998).

Van Scoy (1992) também apontou que a maioria dos gerentes de desenvolvimento de sistemas identifica o seu trabalho como gerenciamento de risco. Porém, raramente estes gerentes recebem as informações que eles precisam para gerenciar efetivamente os riscos nos projetos.

O resultado é um gerenciamento reativo a crises baseado fortemente nos indicadores de custo e tempo. Poucos gerentes identificam, analisam, planejam, acompanham e controlam sistematicamente os seus riscos. Quando o gerente endereça riscos técnicos, os problemas tendem a ser baseados apenas em sua própria experiência e os seus métodos são geralmente incompletos e não documentados. Para o gerenciamento de riscos de sistemas ser efetivo, deve-se evitar o método *ad-hoc*, "a experiência é o único professor" que domina o gerenciamento de risco de sistemas.

Assim, a **solução para a redução de eventos inesperados** no esforço de desenvolvimento de sistemas **é incorporar uma metodologia sistemática e disciplinada para gerenciar os riscos** no processo de desenvolvimento de sistemas.

*Não se pode escapar dos riscos, mas o gerenciamento dos riscos permite lidar mais efetivamente com o futuro, porque a única coisa mais difícil do que prever o futuro é alterar o passado* (VAN SCOY, 1992).

Esta também foi a conclusão a que chegou o Software Program Manager´s Network (HALL, 1998). Este grupo teve como missão identificar as melhores práticas que incrementam a performance e qualidade de sistemas, ao mesmo tempo em que reduzem custo e risco e identificou o **gerenciamento formal de riscos** como a **principal prática de sucesso**.

Higuera (1996) avaliou que todas as áreas no desenvolvimento de sistemas são potenciais fontes de risco ao sistema uma vez que envolvem tecnologia, *hardware*, *software*, pessoas, custos e prazos.

A necessidade de gerenciar riscos também aumenta com a complexidade do sistema. Assim, Higuera (1996) concluiu que há **um aumento da necessidade** de um **método sistemático e de ferramentas** para complementar o conhecimento individual, julgamento e experiência **à medida que a complexidade aumenta**, conforme mostra a figura 4.

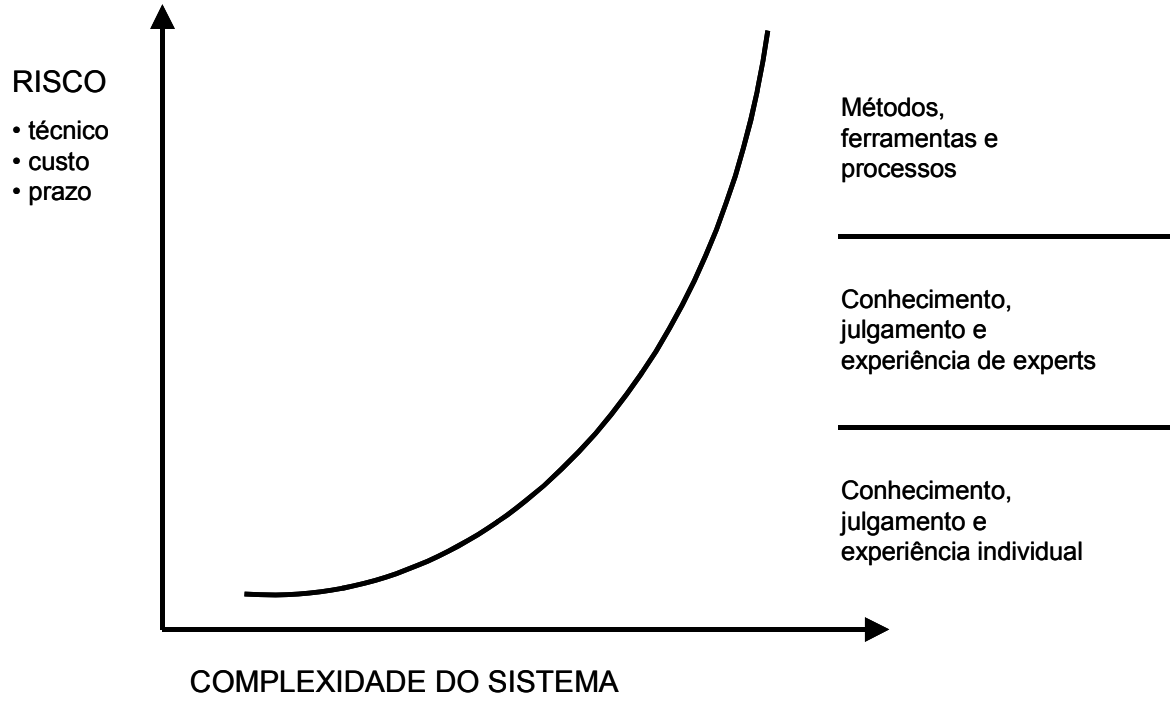


**Figura 4** – A necessidade de gerenciar riscos aumenta com a complexidade do sistema

**Fonte**: Higuera (1996)

O foco no gerenciamento de risco torna-se importante, pois técnicas estruturadas, mesmo técnicas simples, podem ser efetivas em identificar os riscos e existem abordagens, procedimentos e técnicas para reduzir o efeito do risco.

*O objetivo é tomar melhores decisões (identificando os riscos antes que se tornem problemas), comunicar os riscos de uma forma positiva e não ameaçadora e resolver os riscos de uma maneira "cost-effective"* (HIGUERA,1996).

O que é risco? Todos têm um entendimento intuitivo do que é risco; nós estamos envolvidos por riscos em nosso mundo. Mas como o entendimento do risco pode ajudar os projetos a ter mais sucesso? Primeiro, risco não deve ser entendido como problema, mas sim como algo que pode vir a ocorrer no futuro: uma possibilidade e não uma certeza.

Para ser mais preciso, **o risco é composto de dois fatores** (VAN SCOY, 1992):

a. **Probabilidade** ou possibilidade de que virá a ocorrer

b. **Perda resultante** de sua ocorrência

> O risco é parte de qualquer atividade e não pode nunca ser eliminado, tampouco se pode conhecer todos os riscos. O risco em si não é mau; o risco é essencial para o progresso e a falha é freqüentemente uma peça chave do aprendizado. Mas deve-se aprender a balancear as possíveis conseqüências negativas do risco com os potenciais benefícios da oportunidade associada a ele (VAN SCOY,1992).

Na literatura encontram-se várias definições e sugestões de classificação do risco apresentadas no quadro 2.

| Autor | Definição | Classificação |
|---|---|---|
| **FERREIRA, 1988** | Risco é definido como a possibilidade de perda ou de responsabilidade pelo dano | |
| **BOEHM, 1991** | Traduz a definição de risco no conceito fundamental de gerenciamento de risco: exposição ao risco, também chamado de impacto do risco ou fator de risco.<br>A exposição ao risco (RE) é definida pela relação: RE = P(UO) * L(UO), onde RE é a exposição ao risco, P(UO) é a probabilidade de um resultado insatisfatório e L(UO) é a perda ocasionada às partes pelo resultado insatisfatório. | O resultado insatisfatório para o campo do gerenciamento de projetos de software irá assumir várias dimensões:<br>- Para clientes e desenvolvedores<br>- Para usuários<br>- Para mantenedores |
| **VAN SCOY, 1992** | Risco técnico de software é a possibilidade de que a aplicação da teoria, princípios e técnicas de engenharia de software irão falhar na produção do produto correto de software.<br>O risco é composto de dois fatores:<br>a. Probabilidade ou possibilidade de que virá a ocorrer<br>b. Perda resultante de sua ocorrência | |
| **CARR, 1993** | | - Riscos conhecidos<br>- Riscos desconhecidos<br>- Riscos não identificáveis |
| **HALL, 1998** | Risco de sistemas é uma medida da possibilidade e da perda em um resultado insatisfatório que afete o projeto do sistema, o processo ou o produto. | - Riscos do projeto do sistema<br>- Riscos do processo do sistema<br>- Riscos do produto do sistema |
| **WILLIAMS, 1999** | O SEI define risco como a possibilidade de sofrer perdas. No desenvolvimento de um projeto, a perda descreve o impacto para o projeto que pode ser na forma de uma diminuição na qualidade do produto final, aumento de custos, atraso na entrega, perda de *market-share* ou falha. | |
| **PMI, 2000** | Risco é um evento ou condição incerta que, caso ocorra, terá um efeito positivo ou negativo em um objetivo do projeto. O conceito é ampliado para incluir tanto ameaças aos objetivos do projeto como também oportunidades de melhorar estes objetivos. | - Riscos técnicos, de qualidade ou performance<br>- Riscos de gerenciamento do projeto<br>- Riscos organizacionais<br>- Riscos externos |

**Quadro 2** – Definição e classificação dos riscos

Hall (1998) avalia que como o risco é definido como uma possibilidade de perda, trabalhos tradicionais tendem a dar uma conotação negativa ao risco. Por isto, o risco tem sido associado com o não atendimento de requisitos de confiabilidade e segurança. Apesar destes requisitos serem importantes aplicações dos conceitos de risco, isto não impede a utilização dos conceitos de gerenciamento de risco para satisfazer qualquer outro tipo de requisito, como rentabilidade, reusabilidade e qualidade.

O PMBoK (PMI, 2000) amplia o conceito de riscos do projeto para incluir tanto **ameaças aos objetivos** do projeto como também **oportunidades de melhorar estes objetivos**.

Boehm (1991) também procura ampliar o conceito de riscos através da análise do que é resultado insatisfatório para o campo do gerenciamento de projetos de software. Como os projetos envolvem vários participantes (cliente, desenvolvedor, usuário e mantenedor) cada um com diferentes e importantes critérios de satisfação, o resultado insatisfatório irá assumir várias dimensões:

- Para clientes e desenvolvedores: atrasos no cronograma e estouro de orçamentos.
- Para usuários: produtos com funcionalidades erradas, problemas com interfaces, performance ou confiabilidade.
- Para mantenedores: *software* de baixa qualidade técnica

Os vários autores concordam com a definição do PMBoK de que "Risco é um evento ou condição incerta que, caso ocorra, terá um efeito positivo ou negativo em um objetivo do projeto". Uma medida para se avaliar o risco é derivada do conceito de exposição ao risco apresentado por Boehm (1991).

Além disto, o conceito de risco é ampliado pelos autores para abranger ameaças e oportunidades aos objetivos do projeto (PMI, 2000), abranger resultados insatisfatórios para os vários participantes envolvidos (BOEHM, 1991) e abranger os objetivos do produto, processo e projeto (HALL, 1998).

Neste trabalho estaremos adotando a definição de risco do PMBoK, ampliando o conceito de resultados insatisfatórios para abranger além do produto, os objetivos do processo e do projeto. Este conceito será bastante utilizado na identificação de riscos, conforme será visto

no Capítulo 4.1. Também o conceito de exposição ao risco será utilizado na classificação dos riscos, conforme será visto no Capítulo 4.2.

Conforme exposto no quadro 2, Hall (1998), classifica os riscos de sistemas da seguinte forma:

1. Riscos do projeto do sistema. Esta categoria define parâmetros operacionais, organizacionais e contratuais no desenvolvimento do sistema. Riscos de projeto são primariamente uma responsabilidade da gerência. Riscos de projeto incluem restrições de recursos, interfaces externas, relacionamento com fornecedores ou restrições de contratos.
2. Riscos do processo de sistema. Esta categoria inclui procedimentos técnicos e gerenciais. Em procedimentos gerenciais, são encontrados riscos de processo em atividades como planejamento, alocação de recursos, acompanhamento, garantia da qualidade e gerenciamento de configurações. Em procedimentos técnicos, são encontrados riscos em atividades de análise de requisitos, desenho, codificação e testes.
3. Riscos do produto de sistema. Riscos de produto são basicamente uma responsabilidade do corpo técnico e são encontrados em requisitos de estabilidade, performance, complexidade do código e especificação dos testes. Como os requisitos do sistema são normalmente percebidos como flexíveis, o risco do produto é difícil de gerenciar.

O PMBoK (PMI, 2000) propõe uma classificação dos riscos conforme as seguintes categorias:

- Técnicos, de qualidade ou performance – como confiança em tecnologias complexas ou ainda não experimentadas, objetivos de performance irreais, mudanças na tecnologia usada ou nos padrões da industria durante o projeto.
- De gerenciamento do projeto – como baixa alocação de tempo e recursos, qualidade inadequada do plano do projeto, pouco uso das disciplinas de gerenciamento de projetos.
- Organizacionais – como objetivos de custo, tempo e escopo inconsistentes, falta de priorização dos projetos, orçamento inadequado ou cortado, conflito de recursos com outros projetos na organização.

- Externos – como mudanças na legislação, problemas trabalhistas, mudanças na prioridade dos proprietários, inundações, etc que geralmente necessitam de ações de *disaster recovery* ao invés de gerenciamento de riscos.

Outra forma de se caracterizar os riscos é apresentada por Carr (1993):

- Riscos conhecidos – aqueles em que uma ou mais pessoas do projeto estão cientes, se não explicitamente como riscos pelo menos como preocupações.
- Riscos desconhecidos – aqueles que podem ser levantados (tornado conhecidos) se for dada a oportunidade ao time do projeto.
- Riscos não-identificáveis – aqueles que ninguém pôde prever. Por isso, apesar de potencialmente críticos para o sucesso do projeto, eles estão além de qualquer método de identificação de risco.

A caracterização apresentada por Carr (1993) é a base do processo de identificação de riscos baseado em uma taxonomia e apresentada pelo SEI (CARR, 1993), veja Apêndice A, onde um processo estruturado de identificação de riscos é apresentado para documentar os riscos já conhecidos pelo time do projeto e, principalmente, para levantar os riscos desconhecidos.

As formas de caracterização dos riscos apresentadas por Hall (1998) e PMBoK (PMI, 2000) também se fazem presentes na taxonomia de riscos de software desenvolvida pelo SEI (CARR, 1993), veja Apêndice B, e serão utilizadas no processo de identificação de riscos.

## 3.2 Gerenciamento do Risco

Os vários autores propõem um objetivo e processos bastante próximos para o gerenciamento de risco.

*O objetivo da disciplina de gerenciamento de riscos de software deve ser identificar, endereçar, e eliminar itens de risco antes que eles se tornem ameaças à operação do software ou fontes importantes de retrabalho no software* (BOEHM,1991).

Van Scoy (1992) aponta três elementos chave para controlar os riscos em sistemas:

1. Identificar – os riscos devem ser identificados enquanto ainda existe tempo para tomar uma ação.
2. Comunicar – deve-se aceitar que o risco existe e comunicar o risco para aqueles que tem o poder de resolvê-los.
3. Resolver – deve-se tomar uma decisão consciente para agir sobre os riscos.

*Estes três elementos são chamados de "gerenciamento de riscos". A essência do gerenciamento de riscos é a tomada de decisões bem informada em um ambiente de incertezas. Gerência de riscos não lida com decisões futuras, mas com o futuro das decisões atuais* (VAN SCOY,1992).

*O Gerenciamento de riscos de sistemas é a prática de avaliar e controlar o risco que afeta o projeto, processo ou produto do sistema* (HALL, 1998).

Hall (1998) também diferencia o conceito de ação corretiva do gerenciamento de risco.

> Ação corretiva é diferente do gerenciamento de risco porque não há incerteza. A ação corretiva é um procedimento específico que é necessário para resolver um problema conhecido. O gerenciamento de risco é um procedimento genérico que é necessário para endereçar um risco. O gerenciamento de risco é a antítese da ação corretiva: onde um termina o outro se inicia (HALL, 1998).

Os autores também concordam que o gerenciamento de risco deve ser praticado regularmente durante todo o ciclo de desenvolvimento do sistema (HALL,1998; BOEHM,19991;VAN SCOY,1992). Além dos riscos serem dinâmicos, à medida que o projeto progride há um aumento nos recursos e um aumento na percepção dos problemas do projeto, levando a necessidade de uma prática rotineira do gerenciamento de risco.

Porém, apesar de sua importância ter ficado bastante clara, existem várias perspectivas que limitam o gerenciamento rotineiro do risco.

Hall (1998) apresenta duas perspectivas que limitam o gerenciamento rotineiro do risco:

1. Risco visto como uma atividade extra. Neste caso o gerenciamento de risco é visto com uma atividade adicional ao trabalho já designado. O perigo em identificar o risco como menos importante que as tarefas designadas é que o risco tende a ser ignorado quando a prioridade do trabalho é aumentada.

2. Risco visto como uma atividade externa. Neste caso o gerenciamento de risco é considerado de responsabilidade de um time externo. A armadilha neste caso é que quando a pessoa considerada responsável pelo risco não está por perto, o gerenciamento de risco é interrompido.

Assim ela propõe que quando o gerenciamento de risco é implementado como parte do processo de desenvolvimento, o projeto é suportado da melhor forma.

*Risco não é mais ou menos importante que as tarefas do projeto; ao invés disto ele é parte do esforço faltante para a conclusão do projeto* (HALL,1998).

Van Scoy (1992) vai além e afirma que as dificuldades de aplicar o gerenciamento de riscos não terminam com sua incorporação no gerenciamento do projeto. Pois mesmo quando os riscos são conhecidos, existem fortes barreiras culturais que previnem a comunicação do risco. Este problema acontece tanto em projetos onde todos pertencem à mesma corporação, como entre contratado e cliente.

*Para o gerenciamento de risco ser eficiente, a cultura que inibe as pessoas de admitir que possa haver um risco (ou problema) deve ser mudada* (VAN SCOY,1992).

Boehm (1997) também aponta a necessidade de uma mudança cultural para facilitar o gerenciamento de riscos:

> Nossa cultura evoluiu de tal forma que admitir a existência de riscos é freqüentemente confundido com derrotismo. Por isso, um gerente apresentado a um cronograma quase impossível de ser alcançado pode deliberadamente ignorar os riscos para projetar uma atitude confiante. (BOEHM, 1997).

Hall (1998) também introduz o paradoxo do pensamento negativo para auxiliar na eliminação da barreira cultural que previne a comunicação do risco.

*Resultados positivos mostram que o pensamento negativo é positivo. Este paradoxo destrói a ilusão de que os melhores resultados decorrem apenas através de pensamento positivo. Gerenciar riscos é o pensamento negativo que leva a resultados positivos* (HALL, 1998).

Vários outros autores desenvolvem especificamente este tema, como por exemplo Schmidt (1991) com o artigo "Disincentives for communicating risk: a risk paradox". Não é porém objetivo deste trabalho desenvolver o lado comportamental e de mudança cultural das organizações para facilitar a implantação da cultura do gerenciamento de risco. Fica aqui a sugestão para o desenvolvimento de outros trabalhos na área de Gerenciamento de Riscos de Projetos.

## 3.3 Fatores de riscos identificados na literatura

### 3.3.1 Boehm (1991)

Boehm (1991) identificou com gerentes de projeto experientes os principais riscos que poderiam comprometer o sucesso de um projeto de *software*:

- Falta de pessoal
- Prazos e orçamentos irreais
- Desenvolver as funções e propriedades erradas
- Desenvolver interface com usuário errônea
- Especificação contínua
- Fluxo contínuo de solicitação de mudanças
- Atrasos na entrega de componentes fornecidos externamente
- Atrasos em tarefas executadas externamente
- Falha na performance *real-time*
- Pressão nas capacidades da ciência da computação

### 3.3.2 Hall (1998)

Hall (1998) identificou os *Top-10* riscos na área governamental

1. Levantar capital
2. Papéis e responsabilidades
3. Experiência do time
4. Processo de desenvolvimento
5. Planejamento do projeto
6. Interfaces do projeto

7. Engenharia do sistema
8. Requisitos
9. Cronograma
10. Testes

E também os *Top-10* Riscos na área da indústria

1. Recursos: cronogramas agressivos com orçamento fixo quase certamente irão causar um atraso no cronograma e um aumento do custo
2. Requisitos: requisitos mau definidos pelos usuários quase certamente irão gerar especificações técnicas do sistema incompletas
3. Processo de desenvolvimento: processo de desenvolvimento fracamente concebido muito provavelmente irá causar problemas na implantação.
4. Interfaces do projeto: dependência da entrega de sistemas externos tem uma boa chance de causar atrasos no cronograma.
5. Processo de gerenciamento: um fraco planejamento muito provavelmente irá causar um aumento no risco de desenvolvimento.
6. Ferramenta de desenvolvimento: falta de experiência com a ferramenta de desenvolvimento irá provavelmente causar uma baixa produtividade no curto prazo.
7. *Design*: *design* não experimentado pode causar problemas de performance do sistema e incapacidade de atingir os requisitos de performance.
8. Métodos de gerenciamento: a falta de um controle do gerenciamento poderá causar um aumento do risco no projeto e uma queda na satisfação do cliente.
9. Ambiente de trabalho: a localização remota do time do projeto pode tornar o suporte organizacional difícil e causar perda de tempo.
10. Testes e integração: um cronograma de integração otimista tem uma boa probabilidade de aceitar um sistema não confiável.

### 3.3.3 SEI (HIGUERA, 1996)

Como resultado da condução de dúzias de levantamentos de risco e testes de campo, o SEI (HIGUERA, 1996) desenvolveu um banco de dados com os riscos associados ao desenvolvimento de *software*. As análises a seguir são baseadas nos resultados destes levantamentos feitos em diversos setores e comunidades, com diferentes domínios da

aplicação. A classificação dos riscos feita nestas análises utilizou a taxonomia de riscos de *software* desenvolvida pelo SEI (CARR, 1993), veja Apêndice B.

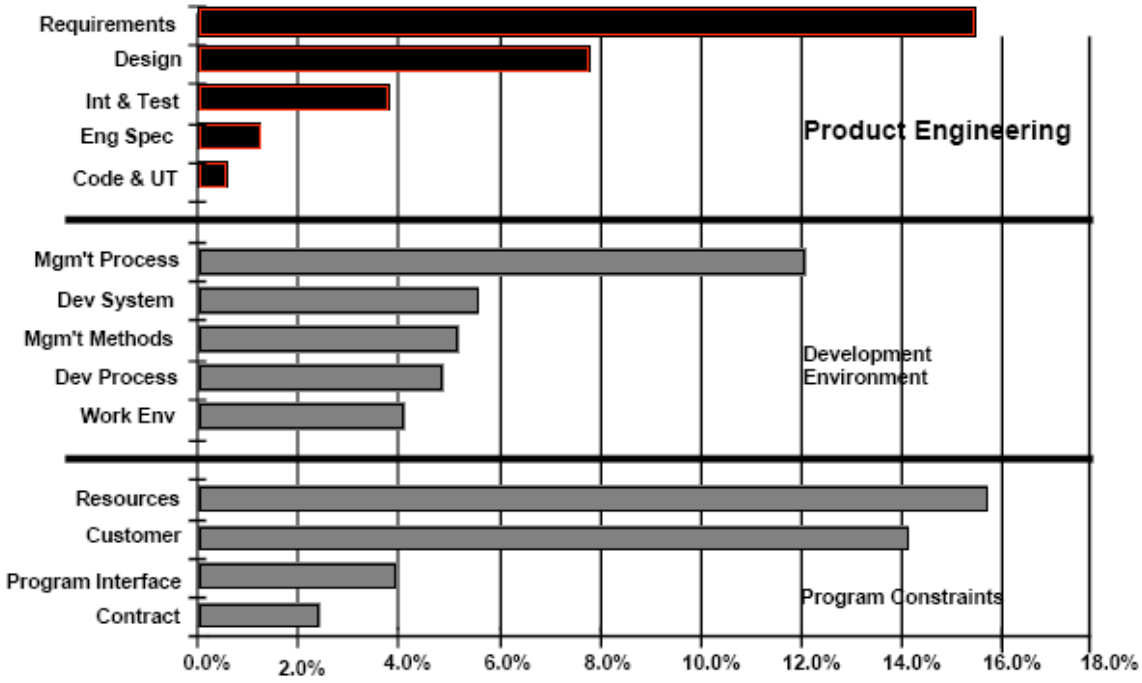


**Gráfico 2** – Níveis de risco identificados pelo SEI

**Fonte**: Higuera (1996)

Como mostra o gráfico 2, a distribuição dos riscos nas três classes da taxonomia apresentou uma distribuição surpreendentemente equilibrada:

- 30% engenharia do produto
- 33% ambiente de desenvolvimento
- 37% restrições do programa

Os dados estatísticos mostram que Requisitos, Processo de gerenciamento, Recursos e Cliente são as quatro fontes de risco mais críticas para o desenvolvimento de *software*. Necessidades e requisitos determinam o caminho pelo qual o desenvolvimento do *software* evoluirá. A dimensão humana (pessoas) forma as outras três fontes críticas de risco: processo de gerenciamento, recursos e cliente. (HIGUERA, 1996).

**Engenharia do Produto**

De acordo com o SEI, os resultados não são surpreendentes, pois confirmam a noção que dentro da engenharia de produto cerca de 80% de todo o risco é atribuído aos Requisitos e *Design*.

Requisitos: dentro da sub-categoria requisito, os atributos Totalidade e Estabilidade dominaram os demais atributos com 36% e 21% respectivamente.

**Ambiente de Desenvolvimento**

As estatísticas confirmam que o processo de gerenciamento é criticamente importante para atender os requisitos de desenvolvimento.

Processo de gerenciamento: dentro desta sub-categoria o requisito Planejamento dominou todos os demais atributos com 54%. Organização do projeto (24%) e Interfaces do programa vieram a seguir (20%).

**Restrições do Programa**

Duas categorias dominaram as fontes de risco nesta classe: Recursos com cerca de 43% e Cliente com cerca de 39%, ou seja, mais de 80% dos riscos são atribuídos a Recursos e Cliente.

Recursos: dentro desta sub-categoria o atributo *Staff* com 50% dominou os demais, seguido por Cronograma (21%) e Instalações (18%).

Cliente: dentro desta sub-categoria, o atributo Gerenciamento ficou com 25%, Atrasos com 21% e Interface do usuário com 19%.

### 3.3.4 Análise dos fatores

Hall (1998) analisou o resultado do levantamento feito por ela quando comparado com a lista dos *Top-10* riscos identificados por Boehm (1991) na década de 80. A análise mostra que progressos foram feitos e que ainda existem desafios para a indústria de *software*.

De acordo com Hall (1998), metade dos riscos apontados por Boehm (1991) são relativos a processos o que mostra a falta de foco no processo que tipificou o desenvolvimento de software na década de 80. Progresso foi feito nesta área e hoje se reconhece o risco no processo e se sabe como identificá-lo. Já recursos, requisitos, interfaces externas e *design*

continuam como áreas de maior risco e devem ser consideradas como oportunidades para melhoria.

A análise das estatísticas do SEI apontando Requisitos como um dos mais importantes fatores de risco, confirma a opinião dos vários autores:
*A maioria dos itens críticos de risco no checklist são relacionados com falhas no entendimento do domínio do projeto e em definir apropriadamente o escopo do trabalho a ser feito – áreas que são pouco enfatizadas na literatura e educação em ciência da computação. (BOEHM, 1991).*

Higuera (1996) também concorda quanto à criticidade na definição do escopo do trabalho.
*Um motivo para que a maioria das sementes de risco seja semeada durante a fase de requisitos e necessidades é que a engenharia de software ainda permanece mais como uma arte do que uma ciência, apesar de todo o desenvolvimento do conhecimento da área (HIGUERA, 1996).*

Portanto, de acordo com Higuera (1996), a mais importante função que o desenvolvedor de sistemas executa para seu cliente é a extração interativa e o refinamento dos requisitos do sistema. A verdade é que o cliente raramente sabe o que quer. O cliente normalmente não sabe que questões precisam ser respondidas e provavelmente não pensou suficientemente no problema para detalhar a especificação da forma necessária.

Jian (2002) discorre sobre a importância de riscos relacionados aos usuários e apresenta estratégias para reduzir estes riscos antes e durante a condução do projeto.

Não é objetivo deste trabalho aprofundar as estratégias para redução de risco de um fator específico, mas devido à sua importância e impacto fica aqui a sugestão para o desenvolvimento de futuros trabalhos na área de Gerenciamento de Risco.

## 3.4 Metodologias de gerenciamento de risco

Conforme foi apresentado, os vários autores recomendam a utilização de uma metodologia sistemática e estruturada para gerenciamento de riscos como a prática mais eficiente para obter sucesso em projetos de desenvolvimento de sistemas.

Neste tópico serão apresentadas e discutidas algumas das metodologias de gerenciamento de risco mais presentes na literatura científica:

- Disciplina do gerenciamento de risco do PMBoK
- Princípios e práticas do gerenciamento de riscos de Barry Boehm
- Metodologia do Software Engineering Institute (SEI) para o gerenciamento de risco
- Área de processo de gerenciamento de risco do Capability Maturity Model Integration (CMMi)

### 3.4.1 PMBoK

O PMI – Project Management Institute definiu o Gerenciamento de Risco como uma das nove disciplinas do Gerenciamento de projetos (PMI, 2000).

> O Gerenciamento de Riscos é um processo sistemático de identificar, analisar e responder aos riscos do projeto. Ele inclui a maximização da probabilidade e conseqüências de eventos positivos e a minimização da probabilidade e conseqüências de eventos negativos para os objetivos do projeto. (PMI, 2000)

O Gerenciamento de riscos foi divido em seis processos principais:

1. **Planejamento do gerenciamento de risco** – decidir como tratar e planejar as atividades de gerenciamento de risco no projeto.
2. **Identificação do risco** – determinar que riscos podem afetar o projeto e documentar suas características.
3. **Análise qualitativa do risco** – fazer uma análise qualitativa dos riscos e suas condições para priorizar seus efeitos nos objetivos do projeto.
4. **Análise quantitativa do risco** – medir a probabilidade e conseqüência do risco, estimando suas implicações nos objetivos do projeto.
5. **Planejamento da resposta ao risco** – desenvolver procedimentos e técnicas para aumentar as oportunidades e reduzir as ameaças aos objetivos do projeto.
6. **Controle e monitoramento do risco** – monitorar riscos residuais, identificar novos riscos, executar planos de redução de riscos e avaliar sua efetividade através de todo o ciclo de vida do projeto.

Cada um destes processos é composto de entradas (*inputs*), ferramentas e técnicas e saídas (*outputs*), que estão detalhadas na figura 5.

## Gerenciamento de risco do projeto

**11.1 Planejamento do gerenciamento de risco**

**.1 Inputs**
.1 Project charter
.2 Políticas de gerenciamento de risco da organização
.3 Definição de papéis e responsabilidades
.4 Tolerância ao risco do stakeholder
.5 Formulário para o plano de gerenciamento de risco da organização
.6 Work breakdown structure (WBS)
**.2 Técnicas e ferramentas**
.1 Reuniões de planejamento
**.3 Outputs**
.1 Plano de gerenciamento de risco

**11.2 Identificação do risco**

**.1 Inputs**
.1 Plano de gerenciamento do risco
.2 Outputs do planejamento do projeto
.3 Categorias de risco
**.2 Técnicas e ferramentas**
.1 Revisões da documentação
.2 Técnicas de coleta de informações
.3 Checklists
.4 Análise de premissas
.5 Técnicas de diagramação
.6 Reuniões de planejamento
**.3 Outputs**
.1 Riscos
.2 Gatilhos
.3 Inputs para outros processos

**11.3 Análise qualitativa do risco**

**.1 Inputs**
.1 Plano de gerenciamento do risco
.2 Riscos identificados
.3 Status do projeto
.4 Tipo do projeto
.5 Precisão dos dados
.6 Escalas de probabilidade e impacto
.7 Premissas
**.2 Técnicas e ferramentas**
.1 Impacto e probabilidade do risco
.2 Matriz de avaliação de probabilidade/impacto do risco
.3 Teste das premissas do projeto
.4 Avaliação da precisão do dado
**.3 Outputs**
.1 Classificação dos riscos do projeto
.2 Lista de riscos priorizados
.3 Lista de riscos para análise e gerenciamento adicional
.4 Tendências nos resultados da análise qualitativa do risco

**11.4 Análise quantitativa do risco**

**.1 Inputs**
.1 Plano de gerenciamento do risco
.2 Riscos identificados
.3 Lista de riscos priorizados
.4 Lista de riscos para análise e gerenciamento adicional
.5 Informação histórica
.6 Julgamento de experts
.7 Outros outputs de planejamento
**.2 Técnicas e ferramentas**
.1 Entrevistas
.2 Análise de sensibilidade
.3 Árvores de decisão
.4 Simulação
**.3 Outputs**
.1 Lista priorizada de riscos quantificados
.2 Análise probabilística do projeto
.3 Probabilidade de atingir os objetivos de custo e prazo
.4 Tendências nos resultados da análise quantitativa do risco

**11.5 Planejamento da resposta ao risco**

**.1 Inputs**
.1 Plano de gerenciamento do risco
.2 Lista de riscos priorizados
.3 Classificação dos riscos do projeto
.4 Lista priorizada de riscos quantificados
.5 Análise probabilística do projeto
.6 Probabilidade de atingir os objetivos de custo e prazo
.7 Lista de respostas potenciais
.8 Limites de risco
.9 Donos do risco
.10 Causas comuns de risco
.11 Tendências nos resultados das análises qualitativa e quantitativa de risco
**.2 Técnicas e ferramentas**
.1 Evitar
.2 Transferir
.3 Mitigar
.4 Aceitar
**.3 Outputs**
.1 Plano de resposta ao risco
.2 Riscos residuais
.3 Riscos secundários
.4 Acordos contratuais
.5 Necessidade de reserva de contingência
.6 Inputs para outros processos
.7 Inputs para revisão do plano do projeto

**11.6 Controle e monitoramento do risco**

**.1 Inputs**
.1 Plano de gerenciamento do risco
.2 Plano de resposta ao risco
.3 Comunicação do projeto
.4 Identificação e análise de risco adicional
.5 Mudanças do escopo
**.2 Técnicas e ferramentas**
.1 Auditoria da resposta aos riscos do projeto
.2 Revisões periódicas dos riscos do projeto
.3 Análise de valor agregado
.4 Medida da performance técnica
.5 Planejamento adicional da resposta ao risco
**.3 Outputs**
.1 Planos de contorno
.2 Ações corretivas
.3 Solicitações de alteração do projeto
.4 Atualizações do plane de resposta ao risco
.5 Banco de dados de riscos
.6 Atualizações nos checklists de identificação de riscos

**Figura 5** – Visão da gerência de risco de projetos

**Fonte**: PMBoK (2000, p.128)

### 3.4.2 Barry Boehm

Boehm (1991) em seu clássico artigo "Software Risk Management: Principles and Practices", apresentou a prática do gerenciamento de risco como sendo composta de dois passos primários, cada um dividido em três passos secundários.

O primeiro passo é a **avaliação do risco**, que envolve a identificação de riscos, a análise de riscos e a priorização de riscos:

- A identificação de riscos gera uma lista de itens de risco específicos para o projeto que podem afetar o seu sucesso. Técnicas usuais incluem *checklists*, exame de fatores de decisão, comparação com a experiência e decomposição.
- A análise de riscos avalia a probabilidade de perda e o tamanho da perda para cada item de risco identificado. Técnicas usuais incluem modelos de performance, modelos de custo, análise de rede, análise de decisão estatística e análise de fatores de qualidade (como confiabilidade, disponibilidade e segurança)
- A priorização de riscos gera uma lista ordenada dos itens de risco identificados e analisados. Técnicas usuais incluem análise de exposição ao risco, análise de redução de risco, e as técnicas Delphi ou de consenso de grupo.

O segundo passo, **controle do risco**, envolve o planejamento do gerenciamento de riscos, a resolução de riscos e a monitoração de riscos:

- O planejamento do gerenciamento de riscos auxilia o gerente de projeto a endereçar cada item de risco (através da aquisição de informações, evitando o risco, transferindo o risco ou reduzindo o risco). Técnicas usuais incluem *checklist* de técnicas de resolução de riscos, análise de custo-benefício e formulários padrões de plano de gerenciamento de riscos.
- A resolução de riscos produz uma situação onde os itens de risco são eliminados ou resolvidos de outra forma. Técnicas usuais incluem protótipos, simulações, *benchmarks*, acordos com pessoas chave.
- A monitoração de riscos envolve acompanhar o progresso do projeto em resolver os seus itens de risco e tomar as ações corretivas quando necessário. Técnicas usuais incluem o uso de listas dos principais itens de risco (T*op-10*) que são destacadas em cada reunião de revisão do projeto.

De acordo com Boehm (1991), o aspecto mais importante para um projeto é manter o foco nos seus fatores críticos de sucesso. Por muitas razões, projetos são focados em atividades que não são críticas para o seu sucesso. Muito freqüentemente estas atividades incluem escrever documentos, explorar problemas técnicos periféricos, atuar politicamente e tentar vender o "mais novo sistema". Neste processo, os fatores críticos de sucesso do projeto são negligenciados, o projeto falha e ninguém ganha.

*A principal contribuição do gerenciamento de risco de software é criar este foco nos fatores críticos de sucesso e fornecer técnicas que ajudem o projeto a lidar com eles*. (BOEHM, 1991).

### 3.4.3 SEI – Software Engineering Institute

O objetivo do Programa de Gerenciamento de Risco de Software do SEI é ajudar os programas de *software* a ter sucesso (VAN SCOY,19992). Van Scoy acredita que um caminho para atingir este objetivo é melhorar a prática do gerenciamento de risco para sistemas *software*-intensivos de forma que clientes e fornecedores utilizem rotineiramente um processo estruturado para gerenciar e comunicar riscos do programa através de todo o ciclo de vida do programa.

A estratégia do SEI pode ser sumarizada da seguinte forma:
*Até que se utilize uma forma disciplinada e sistemática para identificar e confrontar riscos técnicos, nunca seremos capazes de controlar a qualidade, custo ou prazo dos produtos de software* (VAN SCOY,1992).

**a. Estrutura da metodologia do SEI**

Conforme apresentado por Higuera (1996), a estrutura da metodologia desenvolvida pelo SEI para o gerenciamento de risco de *software* está representada na figura 6, e é composta de modelos, práticas e fundamentos conforme detalhado no quadro 3.

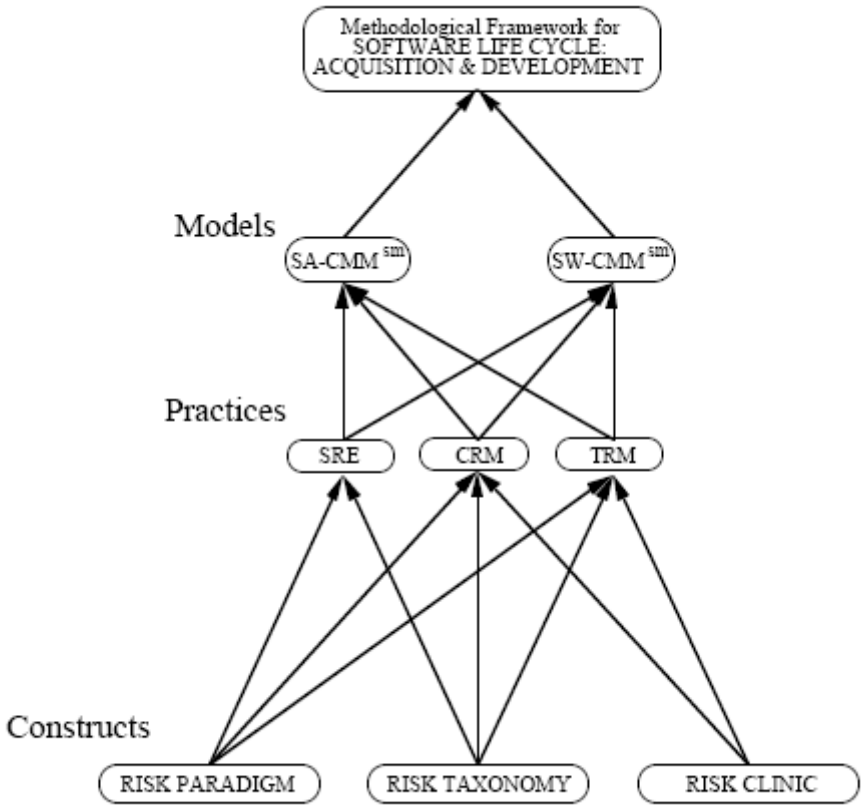


**Figura 6** – Estrutura da metodologia de gerenciamento de risco do SEI

**Fonte**: Higuera (1996)

| | |
|---|---|
| **Modelos** | |
| SW-CMM | Software Capability Maturity Model |
| SA-CMM | Software Acquisition Capability Maturity Model |
| **Práticas** | |
| SRE | Software Risk Evaluation |
| CRM | Continuous Risk Management |
| TRM | Team Risk Management |
| **Fundamentos** | |
| | Paradigma do gerenciamento de risco |
| | Taxonomia do risco |
| | Clínica de risco |

**Quadro 3 –** Estrutura da metodologia de gerenciamento de risco do SEI

**b. Fundamentos**

**Paradigma do gerenciamento de risco**

O paradigma, figura 7, é um modelo de como os diferentes elementos do gerenciamento de riscos de sistemas interagem e uma estrutura para descrever como o gerenciamento de risco pode ser implementado. O paradigma tem uma forma circular para destacar sua natureza contínua. As setas indicam o fluxo lógico da informação entre os elementos do paradigma. Comunicação é chave. É o meio através do qual toda a informação flui. (VAN SCOY,1992).

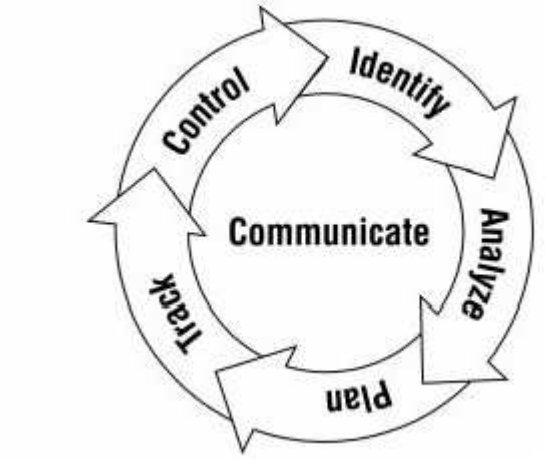


**Figura 7** – Paradigma do gerenciamento de risco do SEI

**Fonte**: Higuera (1996)

- Identificar – localizar o risco antes que ele se torne um problema e afete adversamente o projeto.
- Analisar – transformar o dado bruto do risco em informação para tomada de decisões
- Planejar – transformar a informação de risco em decisões e ações (presentes e futuras)
- Acompanhar – monitorar o status do risco e as ações tomadas contra os riscos
- Controlar – corrigir os desvios das ações planejadas.
- Comunicar – dar o *feedback* nas atividades em ação contra os riscos, riscos atuais, e riscos emergentes para os vários elementos do paradigma e dentro do projeto.

A comunicação do risco fica no centro do paradigma porque sem uma comunicação efetiva nenhuma estratégia de gerenciamento de risco pode ser viável. A comunicação é crítica porque facilita a interação entre os elementos do paradigma. Além disto, os riscos devem ser comunicados para os níveis adequados da organização de forma que os riscos possam ser

analisados e gerenciados efetivamente. Isto inclui os níveis dentro da organização de desenvolvimento, dentro da organização do cliente e, especialmente, através da ligação entre o desenvolvedor e o cliente. (VAN SCOY,1992; HIGUERA, 1996).

**Risk Taxonomy**

A Taxonomia de Risco segue o ciclo de vida do desenvolvimento de software e fornece uma estrutura para organizar dados e informações. O método de identificação baseado em uma taxonomia provê a organização de desenvolvimento de *software* de um processo de entrevistas sistemático para a identificação de riscos (HIGUERA, 1996).

A taxonomia de risco é um dos modelos mais amplos e simples para a identificação de riscos e maiores detalhes podem ser vistos nos Apêndices A, B, C e D.

**Risk Clinic**

A Clínica de risco é um *workshop* que inicia a instalação do CRM dentro da organização. Esta clínica pode ser usada para adaptar o CRM para atender as necessidades específicas do cliente e para implementá-lo em um ou mais projetos.

**c. Práticas**

**SRE – Software Risk Evaluation**

Conforme apresentado por Williams (1999), o SRE pode ser utilizado como um diagnóstico independente que auxilia a organização a determinar a melhor maneira de assegurar o sucesso de um de seus projetos como também pode ser uma base sólida para o programa de gerenciamento de riscos. O SRE levanta, analisa e define estratégias de mitigação para os blocos de dados do gerenciamento de riscos.

O SRE endereça os elementos de identificação, análise, planejamento e comunicação do paradigma do SEI. O SRE é uma ferramenta de diagnóstico e de tomada de decisões para o projeto. Ele é usado para identificar e categorizar especificações de risco do projeto derivadas do produto, processo e das restrições de recursos.

O SRE é mais efetivo como o início do processo de gerenciamento contínuo de risco (CRM) dentro do projeto e do gerenciamento de riscos de times (TRM) entre clientes e fornecedores. O SRE provê a fundação para o CRM e TRM ao fornecer a referência inicial de riscos.

**CRM – Continuous Risk Management**

CRM – é uma prática de engenharia de sistemas com processos, métodos e ferramentas para gerenciar riscos no projeto. Ele fornece um ambiente disciplinado para a tomada de decisões pró-ativas para:

- Avaliar continuamente o que pode dar errado (riscos)
- Determinar que riscos são importantes e que devem ser endereçados
- Implementar estratégias para lidar com estes riscos

Ao usar o CRM, os riscos são avaliados continuamente e usados para a tomada de decisões em todas as fases do projeto. Os riscos são acompanhados e lidados até que sejam resolvidos ou se tornem problemas e então sejam endereçados como tal (WILLIAMS, 1999).

**TRM – Team Risk Management**

TRM – o TRM é um novo paradigma para gerenciar projetos e desenvolver uma visão compartilhada do produto, focando em resultados e usando os princípios e ferramentas do gerenciamento de risco para gerenciar o risco e as oportunidades de uma forma cooperada.

O TRM, como mostra a figura 8, estabelece um ambiente baseado em um conjunto de processos, métodos e ferramentas que permitem ao cliente e fornecedor trabalhar de uma forma cooperada, gerenciando o risco de uma forma contínua através do ciclo de vida e desenvolvimento do sistema (WILLIAMS, 1999).


**Figura 8** – Team risk management

**Fonte**: Higuera (1996)

**d. Modelos**

O SW-CMM -- Software Capability Maturity Model fornece orientação em como obter controle sobre os processo de desenvolvimento e manutenção de *software* e como evoluir para uma cultura de excelência na engenharia de software

O SA-CMM – Software Acquisition Capability Maturity Model fornece orientação sobre os processos de aquisição de *software* e tem uma alta sinergia com o modelo SW-CMM

Os vários modelos do CMM foram posteriormente integrados no modelo CMMi – Capability Maturity Model Integration (CMMi, 2002), cujo objetivo é fornecer orientação para melhorar os processos da organização e sua habilidade em gerenciar o desenvolvimento, aquisição e manutenção de produtos ou serviços.

### 3.4.4 CMMi

A representação contínua do modelo CMMi apresenta quatro grupamentos de áreas chave de processo:

- Gerenciamento de processos
- Gerenciamento de projetos
- Engenharia
- Suporte

Cada área de processo é então suportada por objetivos específicos (SG – *specific goals*) e estes por práticas específicas (SP – *specific practices*), conforme mostra a figura 9.

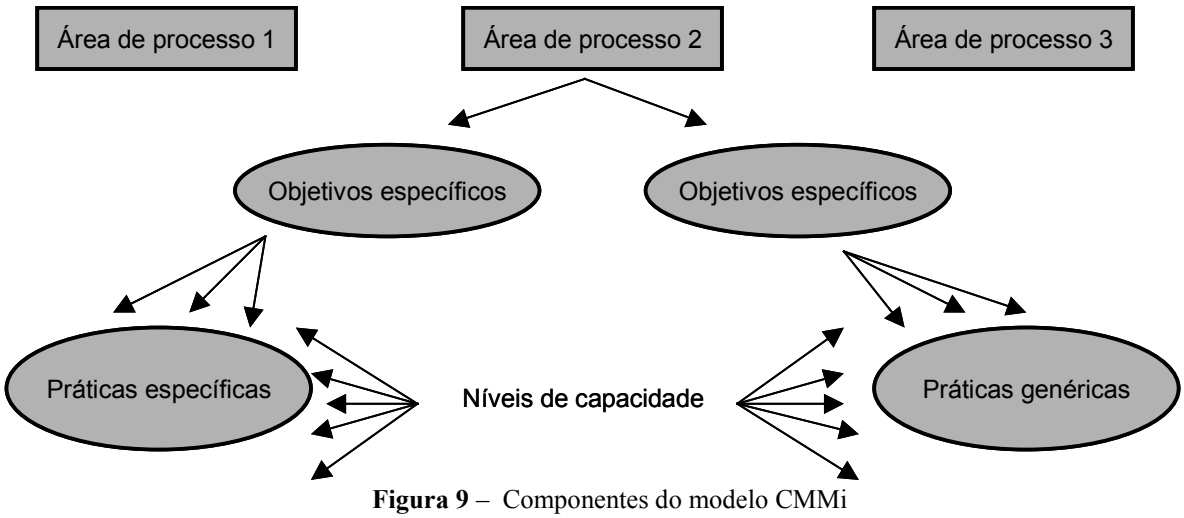


**Figura 9** – Componentes do modelo CMMi

**Fonte**: CMMi (2002)

Áreas de processo: são um agrupamento de práticas relacionadas em uma área de tal forma que, se executadas coletivamente, satisfazem um conjunto de objetivos considerados importantes para fazer uma melhoria significativa naquela área.

Objetivos específicos: aplicam-se a uma área de processo e endereçam as características singulares que descrevem o que deve ser implementado para satisfazer aquela área de processo.

Práticas específicas: são as atividades consideradas importantes para atingir o objetivo específico associado. As práticas específicas descrevem as atividades esperadas que resultam no atendimento dos objetivos específicos da área de processo.

Esta representação foca nas melhores práticas que a organização pode utilizar para melhorar seus processos nas áreas de processos que ela decidiu endereçar. E dentro deste contexto, o **Gerenciamento de Riscos** é classificado como uma **área de processo avançada** dentro da área de Gerenciamento de Projetos.

De acordo com o CMMi (CMMi, 2002) o objetivo do Gerenciamento de Risco é identificar problemas potenciais antes que eles ocorram, de modo que atividades de contenção de risco possam ser planejadas e utilizadas quando necessário ao longo da vida do produto ou projeto para diminuir impactos adversos à obtenção dos objetivos. Ele pode ser dividido em três partes:

1. Definir a estratégia de gerenciamento de risco
2. Identificar e analisar os riscos
3. Lidar com os riscos

As práticas do CMMi para o Gerenciamento de Riscos estão definidas como:

**SG 1 Preparação para o Gerenciamento do Risco**

SP 1.1-1 Determinar as fontes e categorias de risco

SP 1.2-1 Definir os parâmetros de risco

SP 1.3-1 Estabelecer a estratégia de gerenciamento de risco

**SG 2 Identificar e analisar os riscos**

SP 2.1-1 Identificar os riscos

SP 2.2-1 Avaliar, categorizar e priorizar os riscos

**SG 3 Mitigar os riscos**

SP 3.1-1 Desenvolver planos de redução de risco

SP 3.2-1 Implementar os planos de redução de risco

### 3.4.5 Análise comparativa

Utilizando como referência os processos descritos no PMBoK, o quadro 4 descreve a comparação entre os modelos de gerenciamento de risco.

| PMBoK | CMMi | Boehm | Paradigma do SEI |
|---|---|---|---|
| - Planejar a gerência de risco | - Determinar as fontes e as categorias de risco<br>- Definir parâmetros<br>- Estabelecer estratégia para gerência de riscos | | |
| - Identificar riscos | - Identificar riscos | - Identificar riscos | - Identificar riscos |
| - Analisar riscos qualitativamente<br>- Analisar riscos quantitativamente | - Avaliar, categorizar e priorizar os riscos | - Analisar riscos<br>- Priorizar riscos | - Analisar riscos |
| - Planejar a resposta aos riscos | - Desenvolver planos de redução de risco | - Planejar a gerência de risco | - Planejar riscos |
| - Controlar e monitorar riscos | - Implementar planos de redução de risco | - Resolver riscos<br>- Monitorar riscos | - Acompanhar<br>- Controlar |
| | | | - Comunicar |

**Quadro 4** – Comparação entre os modelos de gerenciamento de risco

**Fonte**: adaptado de Machado (2002)

Presente apenas nas metodologias propostas pelo PMBoK e CMMI, Machado (2002) comenta que a atividade **Planejar a Gerência de Risco** tem como objetivo estabelecer o escopo da gerência de risco, determinar a origem e as categorias e definir parâmetros, ajudando assim na definição dos recursos a serem empregados nesse processo.

A atividade **Identificar riscos** tem o mesmo escopo em todas as metodologias apresentadas.

As atividades **Analisar riscos qualitativamente** e **Analisar riscos quantitativamente** apresentadas pelo PMBoK como atividades distintas, são tratadas em conjunto nas outras metodologias. Porém, todas elas têm como objetivo analisar, classificar e gerar uma lista

priorizada de riscos, baseada no conceito de exposição ao risco. O próprio PMBoK cita que a análise quantitativa normalmente segue a análise qualitativa, porém os dois processos podem ser utilizados separadamente ou em conjunto.

A atividade de **Planejar a resposta ao risco** também aparece em todas as metodologias apresentadas, e consiste do desenvolvimento do plano de resposta ao risco.
A atividade de **Controlar e monitorar riscos** está presente em todas as metodologias, com o objetivo de desenvolver as ações definidas no plano de resposta ao risco e monitorar os seus efeitos para tomar ações corretivas quando necessário.

O Paradigma do SEI, apresenta as atividades de gerenciamento de risco de uma forma circular para evidenciar a sua natureza contínua e dinâmica. Além disto, ela introduz um outro elemento que é a atividade de **Comunicar** e a coloca no centro do processo, pois ela facilita a interação entre os elementos do paradigma e sem uma comunicação efetiva nenhuma estratégia de gerenciamento de risco pode ser viável (VAN SCOY, 1992; HIGUERA, 1996).

A comunicação no gerenciamento de riscos não é descrita no PMBoK por existir uma área de conhecimento específica que aborda a comunicação (gerência da comunicação no projeto). No processo de reportes de performance desta área de conhecimento, apesar de focar normalmente em informações de escopo, prazo, custo e qualidade, o PMBoK aponta que muitos projetos também requerem informações sobre riscos.

E na análise de variância, utilizada como ferramenta de reporte de performance, o PMBoK aponta:
*Variações em custo e prazo são as mais freqüentemente analisadas, porém variações do plano nas áreas de escopo, recursos, qualidade e risco são freqüentemente de igual ou maior importância (PMI, 2000).*

Apesar de apontar a necessidade da comunicação dos riscos nos relatórios de performance do projeto, o PMBoK não coloca a mesma ênfase na comunicação como o SEI.

Seleciona-se como modelo de gerenciamento de riscos o PMBoK acrescentando o elemento da comunicação.

# 4 Processos do Gerenciamento de Riscos

Neste capítulo serão apresentados os vários métodos, processos e ferramentas encontrados na literatura para cada uma das etapas do gerenciamento de risco. Para este fim, estaremos adotando a nomenclatura dos processos do PMBoK (PMI, 2000):

- Planejar o gerenciamento de riscos
- Identificar os riscos
- Analisar os riscos
- Planejar a resposta aos riscos
- Controlar e monitorar os riscos

## 4.1 Planejar o Gerenciamento de riscos

O PMBoK (PMI, 2000) apresenta o Planejamento do gerenciamento de riscos como o processo de decidir como abordar e planejar as atividades de gerenciamento de riscos do projeto.

O CMMi (CMMi, 2002) decompõe este processo nas seguintes práticas:

SP 1.1-1 Determinar as fontes e categorias de risco

Nesta etapa são identificas as fontes de risco para fornecer uma base para a análise de risco ao longo do projeto. A categorização destas fontes de risco é útil para coletar e organizar os riscos, servindo de base para consolidar as atividades no plano de resolução de riscos. O CMMi (CMMi, 2002) aponta que uma taxonomia de risco pode ser utilizada como uma referência para determinar as fontes e categorias de risco.

SP 1.2-1 Definir os parâmetros de risco

Nesta etapa são definidos parâmetros para avaliar, categorizar e priorizar os riscos, incluindo o seguinte:

- Probabilidade do risco
- Conseqüência do risco
- Limites para disparar atividades gerenciais

Estes parâmetros definidos previamente são úteis para prover um critério comum e consistente para comparar os vários riscos que serão gerenciados. Também serão úteis para padronizar e poder prover uma base a ser reutilizada em futuros projetos.

SP 1.3-1 Estabelecer a estratégia de gerenciamento de riscos

Nesta etapa é definida a estratégia que será utilizada para gerenciar o risco, endereçando itens como:

- O escopo do esforço de gerenciamento de riscos
- Métodos e ferramentas a serem usadas
- Fontes específicas de risco do projeto
- Como os riscos serão organizados
- Lista de parâmetros
- Técnicas que serão utilizadas para responder ao risco
- Definição das métricas de risco a serem utilizadas
- Freqüência da reavaliação ou monitoramento do risco

## 4.2 Identificar os riscos

Porque o gerenciamento de riscos é raramente executado: a maioria das pessoas não sabe por onde começar. O primeiro passo é saber como identificar os riscos. (HALL, 1998).

Apesar de não apresentar dados que comprovem esta afirmação, parece bastante plausível que uma vez identificados os riscos, torna-se quase que natural a necessidade do desenvolvimento de um plano para a sua resolução. Ao ser apresentado a uma lista de riscos o gerente de projeto pró-ativo é impelido a fazer a análise do risco e definir ações para minimizá-los.

De acordo com Hall (1998) os objetivos da identificação de riscos são:

- Encorajar o time a informar o risco percebido
- Identificar os riscos enquanto ainda há tempo para executar ações
- Descobrir riscos e fontes de risco
- Capturar os riscos em um formato legível
- Comunicar o risco para aqueles que podem resolvê-lo
- Prevenir surpresas no projeto

*O propósito da avaliação de risco é entender o componente de risco nas decisões. A avaliação de risco é um método de descoberta e informação de riscos. Está-se melhor preparado para tomar decisões baseadas no conhecimento do risco do que não o conhecendo.* (HALL,1998).

Para o CMMi (CMMi,2002) a identificação de riscos deve ser um processo organizado e persistente para procurar **identificar riscos reais e prováveis** à realização dos objetivos. Para ser efetivo, a identificação de risco **não deve ser** uma tentativa de **endereçar todo evento** possível **independentemente de quão improvável** ele seja.

**Os riscos identificados formam a base para iniciar as atividades de gerenciamento de riscos**. A lista de riscos deve ser revisada periodicamente para reexaminar possíveis fontes de risco e mudanças nas condições para revelar fontes e riscos não existentes ou desprezados na última revisão da estratégia.

Apesar de não haver limitações de quando se conduzir uma avaliação formal de riscos, ela é mais útil no início do projeto ou quando acontecerem mudanças importantes no projeto.

A atividade de identificação de riscos foca a identificação de riscos e não "colocar a culpa". O resultado da atividade de identificação de risco não deve ser usado pela gerência para avaliar a performance das pessoas.

Como vimos no Capitulo 3.1, o conceito de risco ao projeto é expandido pelos autores de forma a incluir tudo o que possa atrapalhar a realização do objetivo do projeto:

- Para o PMBok (PMI, 2000) o risco inclui ameaças aos objetivos do projeto como também oportunidades de melhorar estes objetivos.
- Para Hall (1998) riscos devem ser procurados em resultados insatisfatórios que afetem o projeto, processo ou produto do sistema.
- Para Boehm (1991) deve-se identificar o que pode ser resultado insatisfatório para o campo de gerenciamento de projetos de software.

Williams (1999) introduz o conceito de **visão de sucesso do projeto**. Para ele, antes de se discutir o tópico de exposição ao risco é importante preparar a fundação definindo a visão de

sucesso de projeto. Deve-se olhar para o futuro e imaginar que o projeto foi completado com sucesso. Três perguntas devem ser respondidas:

- Quando isto acontecerá?
- Como ele será?
- O que o torna um sucesso?

Ao final da discussão a visão de sucesso deve ser editada e reescrita até que se torne satisfatória.

Ao definir a **visão de sucesso do projeto**, forma-se uma expectativa que pode ser compartilhada com todos os envolvidos durante a fase de levantamento de riscos e que pode **identificar as condições em que o projeto coloca esta visão em risco**. Os riscos devem ser identificados em termos de um estágio final desejado. Assim a lista de riscos será completamente diferente dependendo do destino final almejado.

A visão que se tem do risco irá variar conforme o papel dos participantes do processo de levantamento de riscos e também de seu nível hierárquico.

Como avaliou Boehm (1991), como os projetos envolvem vários participantes (cliente, desenvolvedor, usuário e mantenedor) cada um com diferentes e importantes critérios de satisfação, é claro que o resultado insatisfatório irá assumir várias dimensões:

- Para clientes e desenvolvedores: atrasos no cronograma e estouro de orçamentos.
- Para usuários: produtos com funcionalidades erradas, problemas com interfaces, performance ou confiabilidade.
- Para mantenedores: software de baixa qualidade técnica

Para Hall (1998), enquanto que a gerência está focada no lucro do projeto, o corpo técnico tem responsabilidade direta no produto do sistema. Cada pessoa tem designada uma tarefa específica e os riscos que preocupam esta pessoa serão relativas ao critério de sucesso de sua tarefa específica. Consumidores, sub-contratados e usuários finais tem diferentes perspectivas que podem contribuir para uma figura mais completa do risco do sistema.

No processo de levantamento de riscos, devem ser envolvidas pessoas em todos os níveis, pois elas podem contribuir para o gerenciamento do risco identificando a incerteza com relação à suas tarefas designadas:

- Gerentes identificam riscos em custo e prazo ou relação com os clientes.
- Membros do time do projeto identificam riscos técnicos que normalmente são fontes de riscos de custo e prazo.
- O corpo técnico normalmente aponta as mudanças de requisitos com risco e as deficiências no processo de desenvolvimento.
- Os clientes identificam riscos no processo de gerenciamento, nos recursos escassos e já antecipam aumentos dos requisitos.
- Em níveis mais inferiores na hierarquia do projeto, indivíduos têm informações mais detalhadas que podem não ser visíveis para os níveis superiores.

Além disto, comunicar o risco a esse grupo expandido de membros do time (incluindo clientes, *sponsors*, usuários e a alta gerência) traz resultados importantes no curto e longo prazo e deve ser um movimento estratégico dos gerentes de projeto. Apesar de inicialmente provocar um choque que "o projeto pode não ser tão fácil como esperávamos", expandir o time irá permitir um gerenciamento cooperado do projeto, permitindo-se que sejam gerenciados riscos fora do alcance de atuação do time do projeto (HALL,1998).

### 4.2.1 Métodos

Existem uma série de métodos que podem ser utilizados na tarefa de identificação dos riscos.

Hall (1998) apresenta os métodos descritos no quadro 5.

| Método | Comentário |
|---|---|
| *Checklist* | Qualquer um pode identificar riscos usando um *checklist* como um lembrete das possíveis áreas de risco. |
| Entrevista | Riscos podem ser identificados através de perguntas em uma sessão de entrevistas em grupo. A discussão em grupo normalmente traz sinergia. |
| Reuniões | Reuniões periódicas de revisão do projeto são apropriadas para discutir sobre informações de riscos. É importante reservar um espaço na agenda para discutir sobre riscos. |
| Revisão | Riscos podem ser identificados através da revisão de planos, processos e produtos do trabalho. |
| Formulário | Um formulário padrão pode ser usado para registro rotineiro dos riscos identificados. |
| Pesquisa | Categorias selecionadas de pessoas podem identificar riscos rapidamente e sem uma preparação prévia através de um método de pesquisa. É importante registrar a categoria de trabalho da pessoa para entender a perspectiva da resposta da pesquisa. |

**Quadro 5** – Métodos para identificação de riscos
**Fonte**: Hall (1998)

O PMBoK (PMI, 2000) repete alguns métodos e sugere também outros:

- Revisão da documentação – revisão dos planos do projeto e premissas
- *Brainstorming*
- Técnica Delphi – obter consenso de *experts* em um assunto. O facilitador envia um questionário solicitando idéias sobre os riscos importantes do projeto. As respostas são posteriormente circuladas entre os *experts* para comentários adicionais.
- Entrevistas – a pessoa responsável pela identificação de riscos identifica as pessoas apropriadas, apresenta o projeto e fornece informações como o WBS (*Work Breakdown Structure*) e a lista de premissas. Os entrevistados identificam os riscos no projeto baseados em suas experiência.
- *Checklists* – *checklists* para identificação de riscos podem ser desenvolvidos baseados em informação histórica e no conhecimento acumulado de projetos similares. Uma

vantagem é que a identificação de riscos se torna rápida e simples. Uma desvantagem é que é impossível preparar um *checklist* com todos os riscos possíveis, e o usuário pode se limitar às categorias de risco listadas no *checklist*. Deve-se tomar cuidado para explorar itens não cobertos no *checklist* e que sejam relevantes para o projeto.

- Análise das premissas – todo projeto é concebido e desenvolvido baseado em um conjunto de hipóteses, cenários e premissas. A exploração da validade destas premissas pode identificar riscos ao projeto devido a falhas de consistência, exatidão ou abrangência.
- Técnicas de diagramação – como causa-e-efeito, fluxo do processo e diagramas de influência.

O CMMi (CMMI,2002) também apresenta sua relação de métodos para a identificação de riscos:

- Examinar cada elemento do WBS para revelar riscos
- Conduzir um levantamento de riscos usando uma taxonomia de riscos
- Entrevistar os *experts* nos assuntos
- Revisar os esforços de gerenciamento de riscos de produtos similares
- Examinar documentos e bancos de dados de experiências anteriores
- Examinar as especificações e os requisitos acordados

Alguns elementos são comuns a todos os autores, como o exame de todos os elementos do projeto (WBS e premissas), utilização de *checklists* e entrevistas em grupo.

### *a. Checklists*

Hall (1998) avalia que *Checklists* são fáceis de criar e proporcionam um método sistemático de identificar riscos. Pode-se revelar riscos desconhecidos no projeto revisando os fatores críticos de sucesso do projeto, listando todos os itens do caminho crítico do cronograma do projeto, revisando-se o WBS e relacionando as interfaces internas e externas do projeto.

Boehm (1991) também sugere a utilização de um *checklist* para ajudar a identificar e resolver os itens de risco mais importantes do projeto.

Carr (1993) desenvolve uma metodologia de identificação de riscos baseada em um *checklist*: TBQ – Taxonomy Based Questionnaire.
A taxonomia desenvolvida pelo SEI (CARR, 1993) mapeia as características do desenvolvimento de sistemas e conseqüentemente dos riscos ao desenvolvimento de sistemas.
O TBQ consiste de uma lista de questões para levantar problemas, preocupações (potenciais riscos) e riscos em cada grupo da taxonomia. A aplicação do processo é desenhada para assegurar que as questões são direcionadas para as pessoas certas e da maneira correta para produzir resultados ótimos.

Este método permite identificar os riscos sem justificativa e sem uma solução proposta, sendo assim o primeiro e necessário passo para estabelecer a comunicação dentro da organização. O método do TBQ pode ser descrito como uma forma estruturada de *brainstorming*.

Nos Apêndices A a D, explora-se em maiores detalhes a metodologia de identificação de riscos através do TBQ e apresentamos o questionário completo.

**b. Entrevistas**

Williams (1999) descreve em detalhes como deve ser o processo de entrevistas para a identificação e avaliação de riscos para o projeto.

As sessões de entrevistas são a forma básica de se coletar informações de risco do SRE. As entrevistas de riscos são entrevistas estruturadas de pessoas chave do projeto com o foco no seu conhecimento individual do risco do projeto. A atividade traz o conhecimento do participante de uma forma aberta e não ameaçadora. Esta técnica suporta o princípio do conhecimento individual – em sua maioria os riscos do projeto são de conhecimento das pessoas que participam do projeto. Em geral, **a entrevista de risco é o motor que cria a base fundamental do SRE: a especificação do risco.**

A utilização de um *briefing* do projeto é uma excelente oportunidade para apresentar ao time de identificação de riscos o contexto do projeto e seu histórico antes de se iniciar as sessões de entrevistas.

O *briefing* do projeto pode conter:

- Visão de futuro do projeto
- Descrição do produto/sistema
- Pessoal do projeto
- Onde o projeto será desenvolvido
- Cronograma do projeto
- Como o projeto será desenvolvido e quais os processos que serão utilizados
- Análise do orçamento do projeto

As sessões de entrevistas são divididas em duas etapas:

- Entrevista de risco (2,5 h) – onde são feitas perguntas aos membros do projeto utilizando-se o TBQ com o objetivo de levantar os riscos do projeto.
- Avaliação do risco (0,5 h) – esta etapa será discutida no próximo capítulo

Williams (1999) e também Pandelios (1999) discorrem sobre práticas e dicas de condução do processo de sessões de entrevistas.

### 4.2.2 Especificação do risco

Independente dos métodos utilizados na identificação de riscos, um processo chave para a continuação do gerenciamento de riscos é utilizar uma especificação de risco padronizada.

Conforme Williams (1999), a **especificação de risco é o centro de todo o processo de Gerenciamento de Risco**. Ele é uma descrição curta, baseada em fatos de uma preocupação levantada pelos membros do projeto. Esta especificação deve ser acompanhada de um contexto que irá preservar a intenção original da especificação do risco através dos próximos passos do processo de gerenciamento de riscos. Juntos, a especificação do risco e seu contexto formam um bloco de dados onde um programa sólido de gerenciamento de riscos será levantado.

O CMMi (CMMi, 2002) também identifica a documentação dos riscos identificados, incluindo o contexto, condições e conseqüências da ocorrência do risco como um produto típico da Prática padrão de identificar os riscos (SP 2.1–1).

Hall (1998) também discorre sobre a importância da especificação de risco:

> A maneira mais concisa de descrever um risco é escrever uma **especificação de risco** que contenha uma breve descrição do problema do risco, a probabilidade e a conseqüência em termos subjetivos. O uso de um formato padrão aumenta o entendimento do risco e o faz parecer familiar. **O valor de uma especificação estruturada de risco é a sua habilidade em simplificar a comunicação do risco**. A especificação do risco captura a sua essência e deve ser complementada com o contexto do risco, que responde as perguntas (5W 1H) do risco. Um banco de dados de riscos também pode ser utilizado como repositório dos riscos identificados. (HALL, 1998, friso nosso)

Ainda conforme Hall (1998), depois que um problema é identificado, pode-se concluir que ele é um risco ao se definir os principais atributos de risco: probabilidade e conseqüência.

A probabilidade pode ser descrita utilizando-se uma frase subjetiva ou mapeando-se a possibilidade percebida à uma probabilidade quantitativa.

A conseqüência deve ser definida em termos de um resultado insatisfatório que poderia ocorrer caso o risco se realizasse. Para se definir mais especificamente a conseqüência do risco, deve-se ter os objetivos claramente definidos e priorizados.

Williams (1999) sugere um formato padrão para a especificação do risco que também é sugerida de forma semelhante pelos outros autores. Esta especificação deve ser composta de:

- Uma condição: algo que verdadeiro ou aceito como verdadeiro
- Um separador: ponto-e-vírgula, seta, frase de ligação.
- Uma conseqüência: algo que pode decorrer como resultado da condição

Condição → Conseqüência

Exemplos:

Não foi feita nenhuma simulação da performance do sistema → os requisitos de performance do sistema podem não ser atingidos.

Os requisitos parecem estar mudando → não temos certeza se os *test cases* cobrem todos os requisitos.

Não existe um processo formal de controle de mudanças para coordenar todos os grupos afetados → os planos de teste não estão sendo atualizados com as mudanças.

## 4.3 Analisar e priorizar os riscos

Após o uso das técnicas de identificação de riscos, um risco que o projeto corre é de identificar uma quantidade tão grande de riscos que serão necessários vários anos apenas para investigá-los. É neste ponto que a priorização de riscos e as atividades de análise de risco se mostram essenciais (BOEHM, 1991).

O processo de análise de risco compreende as tarefas necessárias para transformar as especificações de risco em uma lista de riscos priorizados.

*As técnicas de análise de risco são ferramentas poderosas para ajudar no gerenciamento de incertezas. Pode-se utilizar a análise de risco para obter e manter suporte a decisões.* (HALL, 1998).

Para Boehm (1991), a **técnica mais efetiva para a priorização de riscos** envolve o **elemento de exposição de risco** descrito anteriormente. Ele permite classificar os itens de risco identificados e determinar quais são os mais importantes e que devem ser endereçados.

Ainda de acordo com Boehm (1991), a técnica mais simples para priorizar riscos é avaliar a probabilidade do risco e a perda em uma escala de 0 a 10 e calcular a correspondente exposição ao risco. Para apresentar esta técnica, Boehm apresenta um exemplo de análise de risco para um caso de sistema de um experimento de satélites (detalhados na tabela 1).

| Resultado insatisfatório | Probabilidade do resultado insatisfatório | Perda causada pelo resultado insatisfatório | Exposição ao risco |
|---|---|---|---|
| A Erros no software interrompem o experimento | 3 - 5 | 10 | 30 - 50 |
| B Erros no software levam a perda nos dados | 3 - 5 | 8 | 24 - 40 |
| C Funções tolerantes à falha causam performance inaceitável | 4 - 8 | 7 | 28 56 |
| D Software de monitoração reporta uma condição insegura como segura | 5 | 9 | 45 |
| E Software de monitoração reporta uma condição segura como insegura | 5 | 3 | 15 |
| F Atrasos no hardware causam atrasos no cronograma | 6 | 4 | 24 |
| G Erros no software de tratamento de dados causam retrabalho | 8 | 1 | 8 |
| H Interface com usuário deficiente causa operação inefeciente | 6 | 5 | 30 |
| I Memória do processador insuficiente | 1 | 7 | 7 |
| J Software de gerenciamento do banco de dados perde dados derivados | 2 | 2 | 4 |

**Tabela 1** – Fatores de exposição ao risco para um sistema de satélite

**Fonte**: Boehm (1991)

Esta técnica permite analisar os seguintes pontos:

- Projetos normalmente focam em fatores que apresentam ou um alto P(UO) ou um alto L(UO). Porém estes podem não ser os fatores chave com uma alta combinação de exposição ao risco. Um dos riscos com maior probabilidade é o item G (com 80%), porém devido ao fato destes erros poderem ser recuperados e não críticos para a missão, eles têm um baixo fator de perda, resultando em um RE de apenas 8. Por outro lado o fator de risco H torna-se um risco de prioridade relativamente alta devido à combinação de sua probabilidade e fator de perda moderados (6 e 5 respectivamente), resultando em um RE de 30.
- A medida do RE também pode ser usada para priorizar a verificação, validação e testes dando para cada classe de erros um peso relativo. Freqüentemente, erros são tratados com o mesmo peso, provocando um esforço muito alto de testes para identificar erros relativamente triviais.
- Existe um alto grau de incerteza na estimativa de probabilidade ou perda devido a um resultado insatisfatório. O tamanho desta incerteza é também uma fonte importante de risco que precisa ser reduzida o mais cedo possível no projeto. O fator de risco C é um ótimo exemplo. Se o P(UO) for avaliado como 4, este item tem apenas um RE moderado de 28. Porém, caso seja avaliado como 8, o RE tem uma altíssima prioridade como 56.

O PMBoK (PMI, 2000) também recomenda a priorização dos riscos de acordo com seu potencial efeito nos objetivos do projeto. E a análise dos riscos seria uma forma de determinar a importância de endereçar riscos específicos e guiar a resposta aos riscos e é feita através do processo de avaliação do impacto e a possibilidade de ocorrência dos riscos identificados.

As ferramentas e técnicas recomendadas são:

- Probabilidade e impacto do risco
- Matriz de classificação de probabilidade e impacto

Uma matriz pode ser construída para designar classificações de risco baseada na combinação de probabilidade e impacto. O objetivo das duas abordagens é apontar um valor relativo para

o impacto no objetivo do projeto caso o risco em questão aconteça. Escalas bem definidas, sejam ordinais (muito baixo, baixo, moderado, alto e muito alto) ou cardinais (0,1 – 0,3 – 0,5 – 0,7 – 0,9), podem ser desenvolvidas usando definições acordadas previamente na organização. Estas definições aumentam a qualidade dos dados e fazem o processo mais replicável.

Hall (1998) também recomenda a utilização de **critérios pré-definidos** para avaliar o risco para assegurar que todos os **riscos sejam julgados** de acordo **com o mesmo padrão**. Ela ainda sugere avaliar a **exposição ao risco contra o critério de tempo para agir**. Assim, os riscos devem ser ordenados conforme a exposição ao risco e ao tempo para agir. A classificação dos riscos deve ser utilizada para direcionar e focar os recursos efetivamente e eficientemente, criando uma lista de riscos priorizados.

As práticas do CMMi (CMMi,2002) para a avaliação e priorização de riscos também recomendam o mesmo processo

SP 2.2-1 Avaliar, categorizar e priorizar os riscos

- **Avaliar os riscos** identificados usando **parâmetros pré-definidos**
    - Uma escala de três a cinco valores pode ser usada para avaliar a probabilidade e a conseqüência
    - Probabilidade: remota, pouco provável, provável, muito provável, quase certamente
    - Conseqüências: baixa, media, alta, desprezível, marginal, significante, crítica, catastrófica
- Categorizar e agrupar os riscos de acordo com as categorias pré-definidas
- Priorizar os riscos

As sessões de entrevistas descritas no capítulo anterior e apresentadas por Williams (1999) também são utilizadas para fazer a avaliação do risco logo após a sessão de identificação de riscos. Nesta parte da entrevista os membros do projeto pontuam individualmente os riscos identificados coletivamente quanto à probabilidade e impacto (exposição ao risco baseadas na tabela padrão de avaliação apresentada ao time) e escolhem os *Top-5* riscos do projeto.

Novamente Williams (1999) e também Pandelios (1999) discorrem sobre práticas e dicas de condução do processo de avaliação e consolidação dos riscos.

### 4.3.1 Critérios de Classificação

Conforme discutido, para uma avaliação uniforme e para podermos fazer comparações entre projetos e acumular experiência em riscos de projetos, deve-se utilizar critérios de classificação e parâmetros pré-definidos na avaliação do risco.

Boehm (1989) recomenda a utilização de tabelas de probabilidade do risco para avaliar a possibilidade do item de risco do projeto, indicando as apresentadas no artigo "Software Risk Abatement", AFSC Pamphlet 800-45, US Department of Air Force, 1988 reproduzido em seu livro.

Baseada na experiência do SEI na utilização da metodologia do SRE, Williams (1999) apresenta os termos de probabilidade, impacto e exposição ao risco no nível de detalhe que o SEI considerou mais útil: quatro níveis de impacto e três de probabilidade, resultando em seis níveis de exposição ao risco.

**Impacto:** é o efeito de um risco em particular no projeto e que é determinado com base no efeito do risco na performance, suporte, custo e prazo de entrega do sistema. Os níveis de impacto são:

- 4 – catastrófico
- 3 – crítico
- 2 – marginal
- 1 – negligenciável

O gerente de projeto deve refinar as definições de impacto para as condições específicas do projeto. Uma tabela como a tabela 2 pode ser utilizada para estabelecer os parâmetros que vão avaliar a classificação do impacto do risco. É importante frisar que negligenciável não significa nenhum impacto, mas sim que o risco por si só não fará com que o projeto perca seus objetivos de performance, suporte, custo ou prazo.

| | Performance | Manutenção | Custo | Prazo |
|---|---|---|---|---|
| **Catastrófico** | não atingir performance técnica | software impossível de se fazer manutenções | grande aumento de custo (> 50% do budget) | Prazo inatingível |
| **Crítico** | degradação significativa da performance técnica | grandes atrasos na modificação do software | aumento importante do custo ( ~30%) | demora importante na entrega (> 30% atraso) |
| **Marginal** | alguma redução na performance técnica | pequenos atrasos na modificação do software | aumento do custo (~10%) | demora na entrega (>10% atraso) |
| **Negligenciável** | pequena à mínima redução da performance técnica, a um nível de detalhe | manutenção irritante e desconfortável | consumo de alguma contingência do budget | consumo de alguma folga -- fora do caminho crítico |

**Tabela 2** – Matriz de impacto do risco

**Fonte**: Willians (1999)

**Risco**: é a chance de que um impacto em particular irá ocorrer. O SEI entendeu que com três níveis de probabilidade torna-se relativamente fácil chegar a um acordo sobre o que a probabilidade significa:

- Provável – se acredita-se que a probabilidade do risco tem a mesma chance que sair cara ou coroa em uma jogada de moeda
- Muito provável – se acredita-se que a probabilidade do risco é significativamente maior do que sair cara ou coroa em uma jogada de moeda
- Improvável – se acredita-se que a probabilidade do risco é significativamente menor do que sair cara ou coroa em uma jogada de moeda

**Exposição ao risco**: é uma função da combinação da probabilidade do impacto e do risco, classificada em seis níveis conforme a tabela 3.

| Probabilidade → / Impacto ↓ | | Muito provável 3 | | Provável 2 | | Improvável 1 | |
|---|---|---|---|---|---|---|---|
| 4 | Catastrófico | 6 | Alto | 5 | Alto | 4 | Médio |
| 3 | Crítico | 5 | Alto | 4 | Médio | 3 | Médio |
| 2 | Marginal | 4 | Médio | 3 | Médio | 2 | Baixo |
| 1 | Negligenciável | 3 | Médio | 2 | Baixo | 1 | Baixo |

**Tabela 3** – Matriz de referência da exposição ao risco

**Fonte**: Willians (1999)

### 4.3.2 Lista de riscos priorizados

O resultado final da avaliação de riscos é uma lista de riscos com uma prioridade associada a cada um (CMMi, 2002).

Para o PMBoK (PMI, 2000) o resultado da análise deve fornecer

- Lista classificada dos riscos do projeto
- Lista dos riscos priorizados
- Lista dos riscos que necessitam de maior análise e gerenciamento
- Tendências nos resultados da análise

Além dos pontos apontados acima, Williams (1999) também comenta que como resultado adicional da análise de riscos um excelente material para a comunicação dos riscos do projeto foi desenvolvido.

## 4.4 Planejar a resposta ao risco

Para Hall (1998) o processo de planejamento de risco compreende as tarefas necessárias para transformar a lista de riscos priorizados em um plano de resolução de riscos.

Uma vez determinados os principais itens de risco do projeto e suas prioridades relativas, é preciso estabelecer um conjunto de funções de controle de risco para manter os itens de risco sob controle (BOEHM, 1991):

1. Desenvolver um conjunto de planos de gerenciamento de riscos que apontam as atividades necessárias para trazer os itens de risco sob controle
2. Desenvolver um plano de gerenciamento de riscos para cada item de risco
3. Integrar os vários planos de gerenciamento de riscos em um plano geral do projeto

De acordo com o PMBoK (PMI,2000), o planejamento da resposta ao risco é o processo de desenvolver opções e determinar ações para melhorar as oportunidades e reduzir as ameaças aos objetivos do projeto. Ele inclui a identificação e o apontamento de pessoas para tornar-se responsável por cada resposta ao risco acordada.

Este processo assegura que os riscos identificados são endereçados apropriadamente. A efetividade do planejamento de resposta irá determinar diretamente se o risco aumenta ou diminui para o projeto.

O planejamento da resposta ao risco deve ser apropriado para a severidade do risco, *cost effective* para atingir o desafio, em tempo para ter sucesso, realístico dentro do contexto do projeto, acordado por todas as partes envolvidas e direcionada para a responsabilidade de uma pessoa. Normalmente é requerida a escolha da melhor resposta dentro de uma série de opções.

O CMMi (CMMi,2002) através da prática SP 3.1-1 Desenvolver planos de Redução de riscos descreve que um componente crítico do plano de mitigação é desenvolver cursos de ação alternativos, *workarounds*, *fallback positions* com uma ação recomendada para cada risco crítico.

O plano de mitigação para um dado risco inclui técnicas e métodos usados para evitar, reduzir e controlar a probabilidade de ocorrência do risco, a extensão do dano caso o risco ocorra ou ambos. Os riscos devem ser monitorados e quando ultrapassam o limite definido devem disparar as ações do plano de mitigação para retornar o risco para um nível aceitável.

Caso o risco não possa ser diminuído o plano de contingência deve ser acionado. Tanto o plano de mitigação como o de contingência são criados apenas para os riscos selecionados onde as conseqüências do risco são identificadas como alta ou inaceitáveis; outros riscos podem ser apenas aceitos ou simplesmente monitorados.

Williams (1999) recomenda utilizar sessões de planejamento da estratégia de mitigação (MSP – *Mitigation Strategy Planning*) com o objetivo de identificar e documentar como as áreas de risco podem ser mitigadas. Nas sessões de MSP discussões profundas e estruturadas são conduzidas para cada área a ser mitigada.

Williams (1999) sugere o seguinte roteiro para estas sessões

- Revisão da visão de sucesso do projeto
- Discussão e identificação das possíveis causas de risco
- Discussão e identificação dos objetivos de contenção dos riscos

- Discussão e determinação das possíveis estratégias de mitigação
- Discussão e determinação das atividades de mitigação que podem suportar as estratégias sugeridas
- Início da identificação das medidas chave que serão usadas para acompanhar e controlar as atividades de mitigação
- Discussão dos possíveis recursos e restrições para as estratégias sugeridas
- Estimativa do esforço requerido

### 4.4.1 Estratégias e alternativas

Os vários autores apontam as seguintes alternativas como estratégias para a resolução do risco:

- Aceitar o risco: é a estratégia de conscientemente aceitar a opção de conviver com a conseqüência do risco. Deve ser usada quando se pode viver com a perda provocada pelo risco.
- Evitar o risco: é a estratégia de eliminar completamente o risco. Deve ser usada quando uma situação perde-perde se apresenta.
- Proteger o risco: é a estratégia de empregar redundância para reduzir a probabilidade ou conseqüência do risco.
- Reduzir o risco (mitigar): é a estratégia de reduzir o risco através da prevenção ou antecipação. O Risco é reduzido ao reduzir a probabilidade do risco ou a conseqüência do risco. Deve ser usada em uma situação onde a alavancagem do risco ocorra (o custo para reduzir o risco é menor do que a redução da exposição ao risco).
- Pesquisar o risco: é a estratégia de investir para obter maiores informações sobre o risco. Deve-se utilizar quando mais informações são necessárias. Prototipagem é uma técnica de levantar informações dos usuários para definir as interfaces do sistema.
- Adicionar reservas: é a estratégia de usar fundos de contingência e folgas no cronograma. Deve ser usado quanto existirem incertezas no custo ou tempo.
- Transferir o risco: é a estratégia de transferir o risco para outra pessoa, grupo ou organização. Deve ser usada quando outro grupo tem o controle.
- Controlar o risco: tomar ações preventivas para minimizar o risco
- Monitorar o risco: observar e reavaliar periodicamente o risco quanto à mudanças nos parâmetros do risco

De acordo com o CMMi (CMMi, 2002) em muitos casos, principalmente no caso de riscos severos, mais de uma estratégia para lidar com os riscos deve ser criada.

Em muitos outros casos, os riscos são aceitos ou apenas monitorados. **Aceitar o risco** é normalmente feito quando o risco é julgado muito baixo para um plano formal de mitigação ou quando não parece haver um meio viável de reduzir o risco. O **Risco é monitorado** quando existe uma forma objetiva, verificável e documentada para medir a performance, tempo ou exposição ao risco que irá disparar o plano de mitigação ou de contingência quando necessário.

**Planos de mitigação** de risco são desenvolvidos e implementados quando necessário para reduzir pró-ativamente os riscos antes que eles se tornem problemas. Apesar dos melhores esforços, alguns riscos podem ser inevitáveis e irão se tornar problemas que impactam o projeto.

**Planos de contingência** devem ser desenvolvidos para riscos críticos para descrever as ações que o projeto deve tomar para lidar com a ocorrência deste impacto.

*O objetivo é definir um plano pró-ativo para lidar com o risco, seja para reduzir o risco (mitigação) ou responder ao risco (contingência), mas em ambos os caso para gerenciar o risco (CMMi, 2002)*

### 4.4.2 Plano de resposta aos riscos

Para o PMBoK (PMI, 2000) o plano de resposta aos riscos deve ser escrito no nível de detalhe necessário para as ações que serão tomadas. Deve incluir alguns ou todos os itens abaixo

- Riscos identificados, sua descrição, áreas do projeto afetadas, causas e como ele pode afetar os objetivos do projeto
- Donos/Responsáveis pelo risco e suas responsabilidades
- Resultados do processo de análise quantitativa e qualitativa
- Respostas acordadas para cada risco no plano de respostas
- O nível do risco residual esperado após a implementação da estratégia
- Ações específicas para implementar a estratégia de resposta definida
- Tempo e orçamento para as respostas

- Planos de contingência e *fallback*

Produtos típicos desta fase são (CMMi, 2002)

- Documento com as alternativas para lidar com cada risco identificado
- Plano de mitigação de riscos
- Planos de contingência
- Lista dos responsáveis por acompanhar e endereçar cada risco

E eles devem conter

- Os níveis e limites que definem quando o risco se torna inaceitável e disparam a execução de planos de mitigação ou contingência
- A pessoa ou grupo responsável por endereçar cada risco
- A relação custo-benefício da implementação do plano de mitigação para cada risco
- Um plano completo de mitigação do risco para coordenar a implementação dos planos individuais de mitigação
- Planos de contingência para riscos selecionados

### 4.4.3 Técnicas para resolver risco

Boehm (1991) apresenta na tabela 4 as técnicas de resolução de risco com maior sucesso na resolução ou eliminação da fonte de risco.

| Risco | Técnica de gerenciamento do risco |
|---|---|
| Falta de pessoal | Escalar pessoas de alto nível, construção de times, acordo com pessoas chave. |
| Prazos e orçamentos irreais | Estimativa detalhada de custos e prazo, design para o custo, desenvolvimento incremental, reutilização de software, detalhamento dos requisitos |
| Desenvolver as funções e propriedades erradas | Análise da organização, análise da missão, formulação do conceito da operaçao, pequisa com usuários e participação dos usuários, prototipagem |
| Desenvolver interface com usuário errônea | Prototipagem, cenários, análise de tarefas, participação do usuário. |
| Especificação contínua | Detalhamento de requisitos, prototipagem, análise custo-benefício, design para o custo |
| Fluxo contínuo de solicitação de mudanças | Desenvolvimento incremental (postergando mudanças para próximas implementações) |
| Atrasos na entrega de componentes fornecidos externamente | Benchmarking, inspeções, conferência de referências, análise de compatibilidade |
| Atrasos em tarefas executadas externamente | Conferência de referências, auditorias pré-contrato, concorrência com protótipos ou design, construção de times. |
| Falha na performance real-time | Simulação, benchmarking, modelagem, prototipagem, instrumentação, ajustes finos. |
| Pressão nas capacidades da ciência da computação | Análise técnica, análise de custo-benefício, prototipagem, conferência de referências |

**Tabela 4** – Técnicas para resolver riscos

**Fonte**: Adaptado de Boehm (1991)

Hall (1998) aponta a **Prototipagem** como uma técnica para reduzir riscos através da aquisição de informação. Conhecimento é obtido através da criação de um modelo físico sem adicionar detalhes de implementação. O protótipo valida os modelos mentais e provê meios efetivos de comunicar com a comunidade usuária.

A **Simulação** é um modelo analítico do comportamento do sistema usado para determinar a capacidade de performance e suas limitações. Simulações auxiliam na avaliação de sistemas grandes e complexos que requerem um grande número de otimizações.

Bartié (2002) recomenda que além dos normalmente já empregados testes de validação do software (aplicados sobre um componente de *software* já desenvolvido), sejam também empregados **testes de verificação** dos documentos gerados durante todas as fases do processo de desenvolvimento de *software*. A principal característica destes testes é o fato de não envolverem o processamento de *software*. Na verdade, essas atividades antecedem a criação

do aplicativo, exatamente para garantir que todas as decisões e definições estabelecidas foram adequadamente entendidas e aceitas pelos diversos grupos que integrarão o processo de desenvolvimento.

O Standish Group (2001) apresenta uma receita para o sucesso do projeto baseada em 10 itens (*The CHAOS ten*) e nos três pilares para o sucesso do projeto: tamanho do projeto, duração e tamanho do time. Ele recomenda que projetos devam durar menos de quatro meses, envolver no máximo quatro pessoas e ter um custo inferior a $ 500.000.

Vários autores como Piney (2002), que discorre sobre como definir a melhor estratégia de resposta aos riscos, e Filiatrault (2000), que apresenta recomendações de como se reduzir o risco ao prazo de entrega, discorrem sobre a melhor forma de se responder a cada risco. Lientz e Rea (2001) apresentam uma extensa lista de problemas de projeto, como eles ocorrem, qual é o seu impacto potencial e como prevení-lo ou endereçá-lo.

Não é, porém, objetivo deste trabalho desenvolver as estratégias de resposta para os riscos, ficando aqui a sugestão para o desenvolvimento de outros trabalhos na área de Gerenciamento de Riscos de Projetos.

## 4.5 Controlar e monitorar o risco

Para o PMBoK (PMI, 2000) o gerenciamento e controle do risco é o processo de monitorar os riscos identificados, acompanhar os riscos residuais e identificar novos riscos, assegurando a execução do plano de riscos e avaliando a efetividade de sua redução no risco. É um processo contínuo durante o ciclo de vida do projeto, uma vez que os riscos mudam com a maturação do projeto, novos riscos emergem ou riscos antecipados desaparecem.

Um bom processo de gerenciamento e controle do risco fornece informações para a tomada de decisões efetivas antecipando a ocorrência de riscos. A comunicação com todas as pessoas chave do projeto é necessária para avaliar periodicamente o aceite dos níveis de risco do projeto.

Conforme o PMBoK (PMI, 2000), o objetivo do processo é monitorar se:

- As respostas ao risco foram implementadas como planejado
- As ações de resposta ao risco foram efetivas como esperado ou se novas respostas são necessárias
- As premissas do projeto ainda são válidas
- A exposição ao risco mudou do seu estado inicial conforme a análise de tendências pode apontar
- Um disparo de eventos de risco aconteceu
- As políticas e procedimentos apropriados foram seguidos
- Riscos aconteceram ou surgiram e que não haviam sido originalmente identificados

O CMMi (CMMi,2002) apresenta a prática SP 3.2-1 para implementar o plano de redução de risco:

SP 3.2-1 Implementar os planos de redução de risco

Para controlar e gerenciar os riscos efetivamente durante o projeto, é preciso seguir um programa pró-ativo para monitorar regularmente os riscos e o status e resultados das ações para lidar com os riscos. (CMMi, 2002)

As técnicas para lidar com o risco são desenvolvidas para evitar, reduzir e controlar o impacto adverso nos objetivos do projeto e para trazer resultados satisfatórios em face dos impactos dos problemas. Ações geradas para lidar com o risco requerem uma carga própria de recursos e tempo dentro dos planos e cronograma base do projeto. Um esforço de replanejamento é necessário para considerar os efeitos em atividades dependentes ou adjacentes.

Produtos típicos desta etapa são:

- Lista atualizada do status dos riscos
- Avaliação atualizada da probabilidade, conseqüência e limites
- Lista atualizada das alternativas para lidar com os riscos
- Lista atualizada das ações tomadas para lidar com os riscos
- Planos de mitigação de riscos

E suas práticas são:

- Monitorar o status do risco
- Acompanhar ações para lidar com os riscos até o seu fechamento
- Chamar as opções para lidar com o risco caso o risco monitorado exceda os limites definidos
- Estabelecer um cronograma para cada ação de mitigação do risco
- Prover recursos para a execução das atividades de mitigação do risco
- Coletar medidas de performance das atividades de mitigação do risco

### 4.5.1 Acompanhamento da Lista dos Riscos Priorizados

Para Boehm (1991), uma vez estabelecidos os planos de gerenciamento dos riscos, o processo de resolução dos riscos consiste em implementar as ações de redução de riscos definidas e o monitoramento do risco garante que este é um processo de ciclo fechado, acompanhando o progresso da redução dos riscos e aplicando ações corretivas sempre que necessário.

Boehm (1991) nomeia a lista dos principais riscos de *Top-10*. Apesar deste nome, a lista não precisa ter necessariamente dez itens e pode variar conforme cada projeto. Nas sessões de identificação de riscos, Williams (1999) sugere eleger os cinco principais riscos.

O **acompanhamento dos *Top-10* itens de risco** é uma **técnica muito efetiva** para os gerentes manterem os **projetos sob controle**. Esta técnica concentra a atenção da gerência para os itens de alto risco e para os fatores críticos de sucesso ao invés de atolar as revisões gerenciais com pilhas de detalhes de baixa prioridade. (BOEHM, 1991, grifo nosso).

*Este tipo de revisão orientada aos itens de risco poupa muito tempo, reduz surpresas para a gerência e mantém o foco nos itens de maior relevância onde o gerente pode fazer a diferença. (BOEHM, 1991).*

O acompanhamento dos *Top-10* itens de risco envolve as etapas:

1. Classificar os itens de risco mais importantes para o projeto.
2. Estabelecer uma agenda periódica de revisões gerenciais do progresso do projeto.
3. Iniciar a reunião de revisão do projeto com um sumário do progresso nos itens de risco *Top-10*. Este sumário deve incluir a classificação atual, a classificação anterior, por

quanto tempo está na lista *Top-10*e um sumário do progresso feito para resolver o risco desde a última revisão.

4. Focar a revisão do projeto em endereçar qualquer problema na resolução dos itens de risco.

Hall (1998) descreve que o risco é encontrado no trabalho que falta para completar a tarefa.

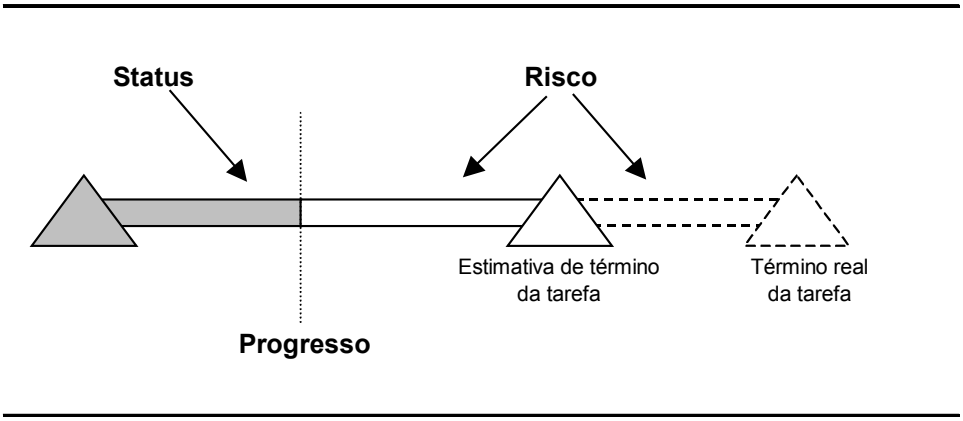


**Figura 10** – Risco e a estimativa de término da tarefa
**Fonte**: Hall (1998)

Como mostra a figura 10, o risco em sistemas existe na incerteza do que é preciso para completar uma certa tarefa. Para qualquer tarefa, o risco está cercado pelo progresso já realizado para completar a tarefa e pelo término real da tarefa. O progresso é um movimento na direção de completar a tarefa. Risco, como o status, é relativo a uma meta específica (neste caso, completar a tarefa).

Enquanto que o **status é uma medida do progresso** em direção à meta, o **risco é uma medida da probabilidade e da conseqüência de não atingir a meta** (um resultado insatisfatório). Risco não é mais nem menos importante que o trabalho, mas antes é um dos obstáculos ainda existentes para atingir a meta. Nenhum progresso é feito sem conquistar riscos. Isto significa que se o risco não foi resolvido no trabalho anterior, o progresso reportado pode estar sob risco.

Goldratt (1998) também entende que os relatórios de status de projetos devem ser focados no trabalho faltante a ser executado e seus riscos e não no trabalho que já foi executado.

*Relatórios de progresso geralmente indicam o problema muito tarde, pois um relatório de progresso pode indicar que 90% do projeto está concluído em um ano e, então, os 10% restantes levam outro ano. (GOLDRATT, 1998)*

As análises de Hall e Goldratt reforçam a posição de Boehm de que a utilização da **lista *Top-10*** é uma forma **muito efetiva de focar** a atenção da alta gerência nos **fatores críticos de sucesso do projeto**. Ela também usa o tempo da gerência de uma forma muito efetiva não desperdiçando a maior parte do tempo com coisas que a alta gerência não pode fazer nada. E caso a alta gerência aborde uma preocupação adicional, ela pode ser facilmente adicionada à lista para ser discutida em futuras revisões (BOEHM, 1991).

## 4.6 Estratégia de implantação

A implementação do gerenciamento de risco envolve a inserção dos princípios de gerenciamento de risco e suas práticas dentro das práticas gerenciais existentes na organização.

Boehm (1991) propõe uma estratégia de implantação gradual, que permite à organização ajustar gradualmente sua cultura às práticas de gerenciamento orientadas ao risco e a modelos de processo orientados ao risco.

Esta estratégia compreende os seguintes passos:

1. Estabelecer um processo de acompanhamento dos *Top-10* itens de risco. Este processo é sem custos e fácil de implantar, provê rápidas melhorias e inicia a familiarização com outras técnicas e princípios de gerenciamento de risco.
2. Identificar um projeto adequado para implementar um plano de alto nível para o gerenciamento de risco.
3. Com base na experiência desenvolvida no projeto inicial, passos sucessivos podem ser tomados para sofisticar as técnicas de gerenciamento de risco e aumentar o campo de sua aplicação.

Hall (1998) recomenda que o processo padrão seja adaptado:
*Enquanto que o processo padrão provê um entendimento comum do processo, papéis e responsabilidades, a adaptação fornece a vantagem da flexibilidade e adaptação para projetos diferentes (HALL,1998).*

O objetivo de adaptar o processo padrão é definir um processo *cost–effective* de gerenciamento de riscos específico para o projeto, onde aspectos particulares do projeto como tamanho, orçamento e estrutura organizacional são analisados. A utilização de um conjunto básico de processos padrão permite a sua reutilização em outros projetos, também fornecendo uma base para que lições aprendidas em cada projeto possam ser disseminadas dentro da organização e para que métricas possam ser identificadas e registradas para aumentar o aprendizado da organização.

Hall (1998) recomenda padronizar os seguintes itens do processo de gerenciamento de riscos:

- Especificação do risco
- Lista de riscos
- Plano de ação de riscos
- Métricas de risco
- Banco de dados de risco

Para a adaptação do processo padrão, Hall (1998) recomenda considerar os seguintes fatores:

- Tamanho do projeto: o risco é menor se o tamanho do projeto é pequeno (menos de 10 pessoas envolvidas)
- Orçamento: o risco é maior se o orçamento é apertado ou se existe um contrato a preço fixo
- Estrutura: o risco aumenta em função do número de interfaces dentro da estrutura organizacional.
- Modelo do ciclo de vida
- Processo de desenvolvimento do sistema: o risco é alto se o projeto não seguir um processo documentado ou se o processo é novo
- Nível de automação

# 5 Pesquisa-ação

Para o desenvolvimento da pesquisa-ação deste trabalho, foi definida a utilização da Metodologia SSM (*Soft Systems Methodology*), pois esta é uma técnica que permite planejar e implementar mudanças, embora também seja usada para desenvolver novos sistemas.

## 5.1 Metodologia SSM

Conforme apresentado por Lunardi e Henrique (2002), o SSM (figura 11) é um método sistêmico que visa a identificação e estruturação de situações problemáticas, caracterizadas por diferentes perspectivas de definição. Ele permite estruturar o problema de uma forma encadeada, analisando-o sob dois pontos de vista: uma relacionada ao mundo real e outra ao pensamento sistêmico.

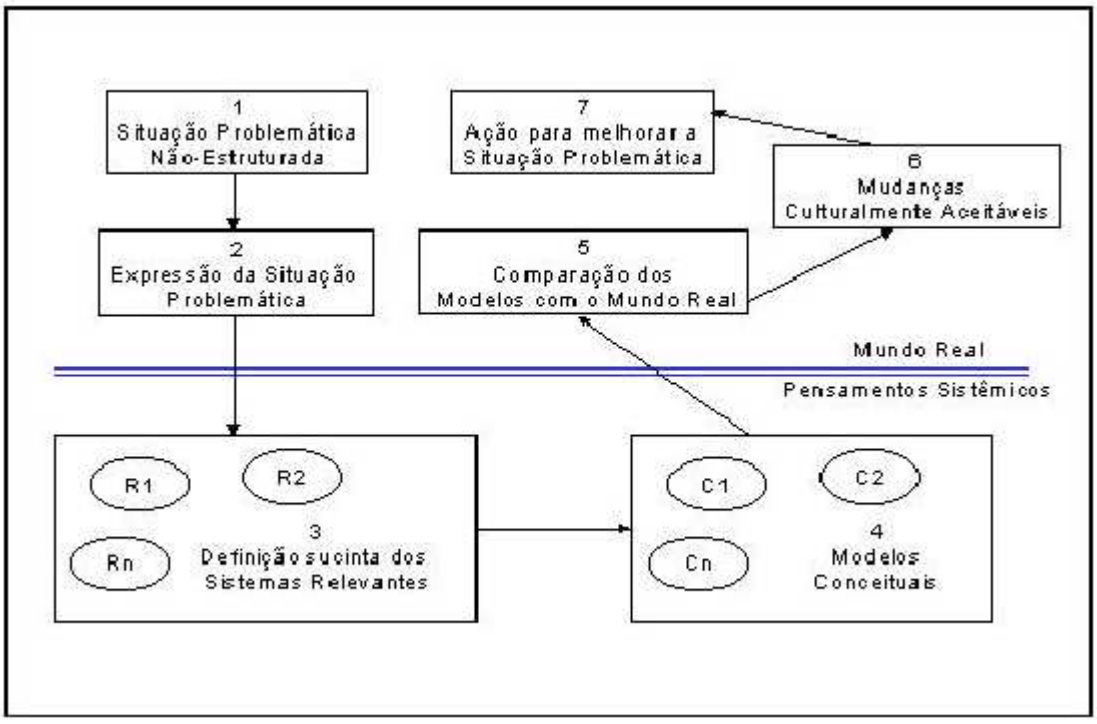


**Figura 11** – Soft Systems Methodology

**Fonte**: Lunardi e Henrique (2002)

O método é composto de sete etapas:

- Etapa 1: Situação Problemática Não-Estruturada – nesta etapa as equipes envolvidas deverão identificar a situação problemática a ser estruturada e resolvida. Podem ser utilizadas técnicas como: figuras ricas (*rich picture*) , 5W1H e *brainstorming*.
- Etapa 2: Expressão da Situação Problemática – de acordo com as técnicas utilizadas na etapa anterior, os problemas do mundo real são identificados pelo grupo e representados de uma forma ilustrativa. A representação final deverá capturar todos os elementos do sistema e os seus relacionamentos.
- Etapa 3: Definição Sucinta dos Sistemas Relevantes – nesta etapa, o grupo busca identificar a definição-chave, analisando e definindo os sistemas mais relevantes. As definições devem ser interpretadas segundo as percepções e valores dos membros das equipes envolvidas, sobre como o sistema deveria ser para desempenhar suas funções.
- Etapa 4: Modelos Conceituais – nesta etapa o grupo deve modelar os sistemas relevantes, identificados na etapa anterior, estritamente de acordo com as respectivas definições-chave.
- Etapa 5: Comparação dos Modelos Conceituais com o Mundo Real – nesta etapa há a confrontação dos Modelos Sistêmicos (etapa 4) com os sistemas desenvolvidos nas etapas 1 e 2, representados pelas figuras ilustrativas. Consiste, essencialmente, na utilização do Modelo Conceitual para questionar a situação do mundo real.
- Etapa 6: Mudanças Culturalmente Aceitáveis – nesta etapa, o grupo, de acordo com os resultados obtidos nas etapas anteriores, faz os seus diagnósticos e apresenta possíveis sugestões para aproximar a situação problemática do pensamento sistêmico.
- Etapa 7: Ação para Melhorar a Situação Problemática – é a etapa final, onde as sugestões propostas são implementadas.

Ao se trabalhar com a SSM, desenvolve-se uma aprendizagem a cada etapa, tanto na percepção do mundo real, nas suas cinco etapas, quanto nas duas etapas do pensamento sistêmico. A vantagem de se trabalhar com a SSM é que não há rigor no cumprimento encadeado das fases, podendo-se, a cada instante, fazer *feedback* de etapas já, aparentemente, cumpridas (LUNARDI e HENRIQUE, 2002).

Os dados utilizados no SSM foram coletados através de observações do pesquisador (que é executivo da empresa no setor de TI), além de entrevistas com os participantes do SSM ao longo de seu desenvolvimento.

## 5.2 Aplicação da Metodologia SSM ao caso Ford

A área de Tecnologia da Informação da Ford Motor Company Brasil tem como objetivo garantir o retorno do investimento de TI, aplicando a tecnologia mais adequada para o suporte às necessidades das diversas áreas de negócio da companhia, que são:

- Marketing e Vendas
- Peças e Serviços
- Finanças
- Recursos Humanos
- Desenvolvimento de Produtos
- Compras
- Manufatura

Contando atualmente com cerca de 150 pessoas, entre empregados e contratados, este suporte compreende:

- Fornecer infra-estrutura tecnológica, com o suprimento de microcomputadores, servidores, telefonia, redes de conexão local (LAN – *Local Area Network*), redes de conexão global com as demais plantas e escritórios da Ford no mundo (WAN – *Wide Area Network*), acesso a Internet, salas de videoconferência, etc.
- Oferecer suporte à utilização das ferramentas e serviços de tecnologia através de um *Help Desk*.
- Desenvolver e manter aplicações de *software* (sistemas) que suportam as atividades das áreas de negócio.
- Disponibilizar suporte técnico para a própria área de TI, com a área de Recursos Técnicos que fornece a infra-estrutura de desenvolvimento de sistemas, área de Banco de Dados que gerencia a criação dos bancos de dados e área de Controle Interno que fornece orientação e suporte para o desenvolvimento de controles e segurança nos sistemas.

A metodologia SSM será aplicada neste trabalho para o contexto do desenvolvimento de novos projetos de sistemas para as áreas de negócio.

O desenvolvimento de novos projetos para cada uma das áreas de negócio é orientado por um plano denominado *IT Business Plan*. Este plano é estabelecido no quarto trimestre do ano com vistas para o ano seguinte e é posteriormente revisado trimestralmente com a Diretoria de cada área de negócio. O desenvolvimento deste plano é direcionado pelos seguintes objetivos estratégicos e prioridades:

- Da companhia
- De cada área de negócio
- Globais e locais da área de TI

Assim que estes objetivos são estabelecidos, as etapas a seguir são conduzidas pelos Gerentes e Supervisores de TI, em conjunto com cada área de negócio, de forma a culminar na aprovação do *IT Business Plan*. As etapas conduzidas pelos Gerentes e Supervisores de TI são:

- Reuniões internas de TI para identificação de potenciais projetos.
- Reuniões com Gerentes da área de negócio para levantamento das, gerando lista com os potenciais projetos.
- Levantamento das prováveis necessidades de recursos de TI para cada projeto
- Criação de lista inicial com os projetos que podem ser contidos com os recursos atuais de TI.
- Revisão com Gerência da área de Planejamento de TI para uniformidade dos critérios dos projetos que farão parte do *IT Business Plan*.
- Revisão dos projetos com as Diretorias de TI dos demais países da América do Sul (Argentina, Chile e Venezuela) para identificação de possíveis sinergias nos projetos.
- Reunião final com Diretoria da área de negócio para revisão das prioridades e dos projetos que serão efetivamente desenvolvidos no próximo ano.

Este processo garante que os recursos de TI sejam alocados de forma alinhada com a estratégia da companhia e, ao final deste processo, é criada a lista dos projetos que serão efetivamente desenvolvidos, identificando para cada projeto:

- Responsável na área de negócio (*Business owner*)
- Prioridade do projeto
- Estimativa de recursos de TI

- Data estimada de entrega do projeto

Para o ano de 2005, o *IT Business Plan* conta com 69 projetos num total de 53.000 horas de desenvolvimento. Além das horas de desenvolvimento de novos projetos, também é verificado no processo de análise de recursos a necessidade de suporte e de manutenções e melhorias nas aplicações existentes.

Cada um destes projetos é gerenciado pelo analista de sistemas responsável em conjunto com o seu Supervisor. O desenvolvimento do plano de ação é baseado no modelo de desenvolvimento em cascata, abrangendo as seguintes etapas:

- Identificação e Levantamento
- Análise
- Desenho
- Codificação e testes
- Implantação
- Manutenção

Reuniões periódicas são estabelecidas para acompanhamento de cada projeto, onde são discutidos e avaliados o status atual do projeto e das tarefas planejadas e os problemas e dificuldades encontrados. Nesta reunião é revisado o cronograma do projeto e também podem ser definidas novas ações. Ao final do projeto é conduzida uma avaliação formal por parte da área de negócio quanto ao projeto entregue.

Para o ano de 2005 está programada a implantação de uma metodologia formal para o desenvolvimento de sistemas, adaptada aos padrões culturais da Ford pela área de TI nos Estados Unidos, chamada de SDM – *Solution Delivery Methodology*.

### 5.2.1 Expressão da situação problemática

Para condução desta etapa, foi realizada uma reunião com uma hora e meia de duração e que envolveu a participação de quatro analistas de sistemas Sênior, experientes na condução de projetos (com uma média de 16 anos de experiência na área de IT e 9 anos na coordenação de projetos), além de contar com a participação do próprio autor (Supervisor de desenvolvimento

de sistemas para a área de Marketing e Vendas, com experiência de 6 anos no desenvolvimento de projetos de TI).

Nesta reunião, utilizou-se a técnica de *brainstorm* para levantar as principais dificuldades que podem ser encontradas no desenvolvimento de projetos de TI na Ford. O autor adaptou uma figura de Higuera (1996) para representar graficamente a situação problemática (figura 12).

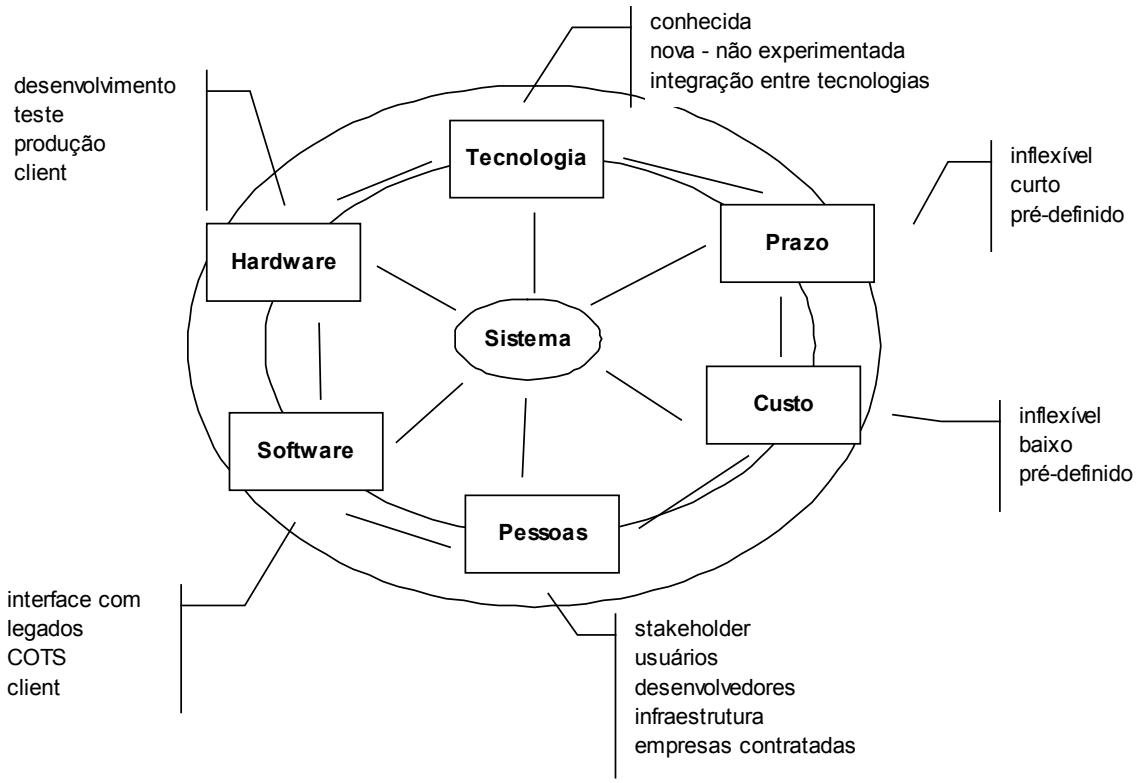


**Figura 12** – Principais preocupações identificadas pela equipe de TI da Ford.

A relação detalhada das preocupações e das situações que podem ser encontradas no ambiente de projetos de TI da Ford e que precisam ser gerenciadas, pode ser vista no quadro 6.

| Ator | | Potenciais problemas ou preocupações |
|---|---|---|
| **Tecnologia** | | desconhecida, não-experimentada |
| | | manter consistência na estratégia de linguagem de desenvolvimento |
| | | integração de diferentes plataformas |
| **Prazo** | | incluir procedimentos e metodologias para qualidade do processo de TI |
| | | necessidade de entrega do produto em prazo curto ou com data pré-definida |
| | | dificuldade de estimar precisamente e de aprovar revisões posteriores |
| **Custo** | | pressão por baixo custo |
| | | dificuldade de gerar estimativa precisa e completa |
| **Hardware** | | disponibilidade no tempo correto |
| | | procedimentos desconhecidos para novas tecnologias |
| | desenvolvimento | diferenças entre os ambientes de desenvolvimento e produção |
| | teste | massa de dados para teste não disponíveis |
| | produção | configuração requerida desconhecida |
| | client (usuário) | configuração de PC e rede |
| **Software** | legados | interface com software e processos já existentes |
| | | dependências de prioridade de outras áreas e testes complexos |
| | | interface com COTS |
| | | interface com outras tecnologias |
| | | adquirir licenças de uso de ferramentas de desenvolvimento |
| | client (usuário) | configuração do software (browser, sistema operacional) |
| **Pessoas** | stakeholder | falta de envolvimento |
| | | falta de participação ou disponibilidade |
| | | falta de comprometimento |
| | usuário | não sabe o que quer ou não consegue definir |
| | | falta de participação ou disponibilidade |
| | | velocidade nas mudanças das regras de negócio |
| | | falta de domínio do processo atual |
| | | resistência à mudança nos processos de trabalho atuais |
| | infraestrutura de IT | envolvido em múltiplas tarefas |
| | | não disponível no momento necessário |
| | | gargalo no processo de desenvolvimento |
| | desenvolvedor | multi-tarefa, envolvido no dia-a-dia e em vários projetos simultaneamente |
| | | sem experiência anterior na tecnologia |
| | | falta de conhecimento dos processos/sistemas atuais |
| | | falta de experiência na metodologia de desenvolvimento |
| | | pouca experiência no gerenciamento de projetos |
| | empresa contratada | desconhecimento dos processos de negócio |
| | | dificuldade de estimar recursos e prazo com precisão |
| | | sem experiência na tecnologia |
| | | mudanças de pessoas |
| | | ambientes de HW e SW diferentes do ambiente real |
| | | deficiência de massa de dados para teste |
| | | dificuldade de comunicação e interação |

**Quadro 6** – Detalhamento das principais preocupações identificadas pela equipe de TI da Ford

### 5.2.2 Modelo conceitual

O modelo conceitual utilizado foi baseado na revisão teórica desenvolvida por este autor, resultando na seguinte definição sucinta:

*O objetivo do gerenciamento de riscos de software deve ser identificar, endereçar e eliminar itens de risco antes que eles se tornem ameaças à operação do software ou importantes fontes de retrabalho no software (BOEHM, 1991).*

Sendo detalhado nas etapas:

- Identificar os riscos: localizar o risco antes que ele se torne um problema e afete adversamente o projeto.
- Analisar e priorizar os riscos: transformar o dado bruto do risco em informação para tomada de decisões.
- Planejar a resposta aos riscos: transformar a informação de risco em decisões e ações (presentes e futuras).
- Controlar e monitorar os riscos: monitorar o status do risco e as ações tomadas contra os riscos e corrigir os desvios das ações planejadas.

### 5.2.3 Comparação do modelo com o mundo real

Nesta etapa foi realizada uma nova reunião, com duas horas de duração, com quatro analistas Sênior da área de TI, onde, após uma apresentação dos modelos conceituais de gerenciamento de riscos, discutiu-se a utilização das etapas de gerenciamento de riscos no modelo atual de gerenciamento de projetos.

O resultado da análise pode ser visto no quadro 7.

| Atividade conceitual | Presença Real | Avaliação (boa, média, ruim) | Responsável | Comentário |
|---|---|---|---|---|
| **Identificar riscos** | sim | média | Analista Supervisor | Sim em algumas áreas, mas de forma não estruturada. Em outras áreas, não. Depende das pessoas envolvidas. Alguns riscos são identificados mas não são comunicados (vistos de forma negativa). |
| **Analisar risco** | sim | ruim | Analista Supervisor | Acontece quando algum risco é identificado, mas sem um formato padrão e sem documentação. |
| **Planejar resposta ao risco** | sim | ruim | Analista Supervisor | Nem sempre são desenvolvidas ações para responder ao risco e os riscos podem se tornar problemas. Conseqüência dos riscos não terem sido documentados. |
| **Acompanhar e controlar** | nao | --- | Analista Supervisor | --- |

**Quadro 7** – Comparação dos modelos conceitual e real

A principal conclusão a que o grupo chegou foi a de que no processo atual o gerenciamento de riscos é conduzido de forma não estruturada e de que depende totalmente da experiência das pessoas envolvidas.

De acordo com o grupo, alguns gerentes de projeto procuram identificar riscos durante as reuniões de revisão de status do projeto, analisando as etapas do cronograma e através do questionamento aos analistas líderes do projeto das principais dificuldades a serem enfrentadas. Porém, esta não seria uma prática usual de todos os gerentes de projeto.

Alguns riscos, principalmente riscos que podem levar a atrasos no cronograma, podem não ser levantados pelo analista por receio de ser entendido de uma forma negativa.

Além disto, por não haver um processo estruturado de documentação, algumas preocupações levantadas pelos envolvidos podem não ser registradas como riscos nas reuniões de revisão de status e não ter ações desenvolvidas para mitiga-los. Neste caso, por vezes, riscos que poderiam ser mitigados, acabam por se tornar problemas reais.

### 5.2.4 Mudanças culturalmente aceitáveis

Ao final desta mesma reunião, o grupo de trabalho procurou identificar ações que poderiam ser desenvolvidas no gerenciamento dos projetos para cada uma das etapas do modelo conceitual.

- Identificar riscos
    - Conduzir reuniões com os envolvidos (usuários, desenvolvedores, áreas de suporte de TI)
    - Utilizar *checklist* adaptado conforme a categoria do projeto
        - Cat. 0 e 1 – questionário resumido do TBQ
        - Cat. 2 e 3 – questionário completo do TBQ
- Analisar e priorizar riscos
    - Classificar os riscos conforme parâmetros pré-definidos (matriz de referência da exposição ao risco do SEI, tabela 3)
    - Identificar lista com os *Top-10* riscos
- Planejar resposta aos riscos
    - Desenvolver ação, responsável e data e incluir no plano maior do projeto
- Acompanhar e controlar os riscos
    - Incorporar a revisão dos *Top-10* riscos nas reuniões de revisão de status do projeto.

A classificação nas categorias de 0 a 3 segue um padrão da metodologia SDM, e que calcula a Categoria do projeto conforme os critérios:

- Quantidade de horas do projeto
- Quantidade de pessoas gerenciadas pelo time do projeto
- Quantidade de interfaces com áreas externas ao time do projeto
- Quantidade de interfaces do sistema
- Número de plataformas tecnológicas envolvidas

Apesar de não ter sido desenvolvida nesta etapa a análise e proposta do Planejamento do gerenciamento de riscos, apresentado pelo PMBoK e pelo CMMi, como resultado da utilização da técnica SSM para desenvolver a proposta de gerenciamento de riscos, foram definidos todos os elementos esperados do planejamento de riscos:

- Determinação das fontes e categorias do risco: proposta a utilização do TBQ

- Definição dos parâmetros de risco: proposta a utilização da matriz de referência da exposição ao risco do SEI, tabela 3
- Estabelecimento da estratégia de gerenciamento de riscos: modelo proposto

Além das recomendações acima, o grupo identificou que implementar o levantamento de riscos exigiria uma mudança cultural, porém conclui-se que ao se utilizar um processo estruturado de identificação de riscos, esta barreira praticamente não existiria.

O grupo também lembrou a importância de incluir as horas e recursos necessários para as tarefas de risco no planejamento do projeto e que havia uma ótima oportunidade de se implantar o gerenciamento de riscos juntamente com a metodologia SDM, ajudando a inibir a reação de que seria mais uma atividade extra para os analistas.

Um dos analistas comentou que: "O mais difícil e importante seria dar o pontapé inicial que é levantar os riscos". Assim, a conclusão do grupo é de que seria totalmente viável a implantação do processo de gerenciamento de risco na área de TI da Ford, e o grupo acredita que devido à forte cultura da empresa de desenvolver e controlar planos de ação, uma vez levantados e documentados os principais riscos do projeto, as demais etapas de gerenciamento do risco acontecerão naturalmente.

### 5.2.5 Ação para melhorar a situação problemática

Após o desenvolvimento da comparação com o modelo real e da definição das mudanças culturalmente aceitáveis, o autor preparou uma apresentação para toda a Gerência e Diretoria de TI da Ford com uma revisão da teoria de gerenciamento de riscos e as sugestões do grupo. Com duas horas de duração, esta apresentação teve o propósito de discutir a viabilidade da real implantação das sugestões e também colher as recomendações deste grupo.

Participaram desta apresentação o Diretor de TI da Ford, o Gerente de desenvolvimento de sistemas de Vendas e Marketing e Peças e Serviços, o Gerente de desenvolvimento de sistemas de Recursos Humanos e Finanças, o Gerente de desenvolvimento de sistemas de Compras, Manufatura e Logística, o Supervisor de desenvolvimento de sistemas de Peças e Serviços, o Supervisor de Bando de Dados e o Supervisor de Infra-estrutura, com experiências entre 17 e 31 anos na área de TI, sendo de 3 a 20 anos como executivo.

Este grupo também entendeu que atualmente o gerenciamento de risco é feito, porém de uma forma não estruturada e totalmente dependente da experiência das pessoas. Porém, ele entendeu que a avaliação do processo atual deveria ser melhor do que a que foi apontada pelos analistas. Esta diferença de percepção foi entendida devido à participação destas pessoas em projetos de maior relevância, que normalmente têm um acompanhamento mais detalhado e também devido à sua grande experiência anterior permitir mais facilmente identificar riscos ao projeto.

O grupo também concordou com as recomendações do time de analistas e a opinião do Diretor de TI é de que o processo deva ser aplicado a todos os projetos de TI.

Uma percepção importante do grupo foi de que através do processo estruturado de identificação de riscos, pessoas menos experientes poderão levantar riscos que normalmente apenas os mais experientes identificariam.

Uma recomendação do Diretor de TI foi a de que se devem incorporar os objetivos corporativos da área de TI à visão de sucesso do projeto, para que riscos a estes objetivos também sejam identificados e gerenciados.

Também foi discutido neste grupo como o processo apresentado estaria alinhado com as práticas de gerenciamento de projetos presentes na metodologia SDM. O autor demonstrou que o processo de gerenciamento de riscos está totalmente alinhado com as práticas do SDM, uma vez que esta metodologia é baseada nas práticas recomendadas pelo PMI.

Assim, a proposta do modelo conceitual a ser implementado ficou definida com as seguintes etapas:

- Na fase de Identificação e Levantamento da metodologia SDM:
    - Identificar riscos
        - Conduzir reuniões com os envolvidos (usuários, desenvolvedores, áreas de suporte de TI, gerente do projeto, supervisão e gerência de TI).
        - Discutir a visão de sucesso do projeto, incluindo os objetivos de TI.
        - Utilizar como ferramenta o *checklist* do TBQ, adaptado conforme a categoria do projeto:
            - Cat. 0 e 1 – questionário resumido do TBQ

        - Cat. 2 e 3 – questionário completo do TBQ
    - Analisar e priorizar riscos
        - Utilizar a matriz de referência da exposição ao risco do SEI, tabela 3, para classificar os riscos.
        - Identificar a lista com os *Top-10* riscos, registrando-os no documento padrão da metodologia SDM (*Risk Management Log*).
    - Planejar a resposta aos riscos
        - Desenvolver ação, responsável e data para os riscos *Top-10*.
        - Incluir estas ações no plano maior do projeto.
- Durante as demais fases da metodologia do SDM:
    - Acompanhar e controlar os riscos
        - Incorporar a revisão dos *Top-10* riscos nas reuniões periódicas de revisão de status do projeto.
        - Revisar o documento *Risk Management Log*, atualizando a relação de riscos e os planos de resposta ao risco.
    - Revisar a identificação de riscos quando de uma mudança importante no projeto

# 6 Conclusões

A presente pesquisa teve como objetivo propor um processo estruturado para planejamento e controle preventivo de problemas através do gerenciamento dos riscos inerentes aos projetos de tecnologia da informação para diminuir o stress e esforço necessário para os times de projeto concluírem os trabalhos no tempo, custo e com a qualidade esperada.

Através da pesquisa-ação, realizada no setor de TI da Ford Motor Company Brasil, foram obtidos dados através da observação direta do pesquisador e da utilização de entrevistas não-estruturadas. A partir da análise dos dados coletados e da utilização da metodologia SSM, foram obtidas as conclusões em relação aos objetivos propostos e que são apresentadas a seguir.

## 6.1 Descrição e análise dos métodos de gerenciamento de riscos de projetos

Foram descritos os métodos de gerenciamento de projetos apresentados pelos seguintes autores:

- PMI – que apresenta o gerenciamento de riscos como uma das nove disciplinas de gerenciamento de projetos.
- CMMi – que apresenta o gerenciamento de riscos como uma área de processo avançada dentro da área de gerenciamento de projetos
- Barry Boehm – autor que apresentou que gerentes de projeto de sucesso eram bons gerentes de risco
- SEI – que apresenta o seu paradigma de gerenciamento de risco como um dos fundamentos para o desenvolvimento de sistemas

A análise comparativa feita entre os vários métodos mostrou que todos eles são muito parecidos e utilizam basicamente as mesmas etapas:

- Identificar os riscos
- Avaliar e priorizar os riscos
- Planejar a resposta aos riscos
- Controlar e monitorar os riscos

E as principais diferenças se resumiram a:

- PMI e CMMi apresentaram uma etapa adicional de planejamento para o gerenciamento de risco
- SEI apresentou em seu paradigma a comunicação como um elo central de todo o processo de gerenciamento de risco

Também foram abordadas as principais dificuldades apontadas na literatura para a implementação do gerenciamento de risco:

- Risco visto como uma atividade extra
- Risco visto como uma atividade externa
- Barreira cultural que inibe a comunicação do risco

## 6.2 Identificação dos métodos que são mais aplicáveis à área de projetos de Tecnologia da Informação

Dos métodos discutidos, apenas o PMI tem o propósito de ser um método aplicável a qualquer tipo de projeto, sendo que os demais métodos são orientados para o desenvolvimento de sistemas. Devido à semelhança entre todos eles e a grande aceitação dos conceitos do PMI, foi adotada durante o trabalho a terminologia do PMI.

Porém, durante a análise das ferramentas e técnicas recomendadas pelos autores, foram selecionadas pela sua simplicidade de utilização e aplicabilidade à área de TI várias ferramentas e técnicas que não estavam presentes no material do PMI. Os relatórios técnicos do SEI, por seu foco na área de TI e também por terem sua utilização testada e experimentada na prática, forneceram as principais ferramentas recomendadas pelo autor. Entre elas:

- TBQ – *Taxonomy Based Questionnaire*
- Especificação padrão do risco
- Matriz de exposição ao risco

Uma técnica simples apresentada por Barry Boehm, mas de resultado importante que é recomendada pelo autor, é o acompanhamento da lista dos riscos priorizados (*Top-10* itens de risco). Pois, de acordo com Boehm (1991), este tipo de revisão orientada aos itens de risco poupa muito tempo, reduz surpresas para a gerência e mantém o foco nos itens de maior relevância onde o gerente pode fazer a diferença.

A representação gráfica do paradigma do SEI também foi utilizada pelo autor para apresentar as etapas do gerenciamento de riscos, devido a sua:

- Facilidade de interpretação pela representação gráfica
- Apresentar a natureza cíclica do gerenciamento de riscos
- Similaridade com o ciclo do PDCA (*Plan – Do – Check – Act*) já bastante difundido pelas metodologias de qualidade total
- Foco na comunicação como elo central do processo de gerenciamento de riscos

## 6.3 Análise do ambiente de projetos de Tecnologia da Informação na Ford e os elementos de risco mais freqüentes

Para desenvolver a análise do ambiente de projetos de tecnologia da informação na Ford e identificar os elementos de risco mais freqüentemente enfrentados pelos times de projeto, foram utilizadas as etapas 1 e 2 da metodologia SSM, contando com a participação de vários analistas experientes na condução de projetos.

O ambiente que foi representado pela figura 12 – Principais preocupações identificadas pela equipe de TI da Ford e pelo quadro 6 – Detalhamento das principais preocupações identificadas pela equipe de TI da Ford está inserido nas descrições apresentadas na literatura, especialmente se compararmos com os principais fatores de risco discutidos no tópico 3.3.4 e com a própria taxonomia do SEI.

## 6.4 Proposta e avaliação de um processo estruturado para gerenciamento destes riscos na Ford

A condução das demais etapas da metodologia SSM culminou com o desenvolvimento de uma proposta de processo estruturado para o gerenciamento de riscos na Ford.

Para poder ter sua implantação discutida, foi importante comparar a proposta desenvolvida a partir dos conceitos expostos neste trabalho com a metodologia de desenvolvimento de sistemas que está sendo implantada na Ford (SDM – *Solution Delivery Methodology*). Porém, como esta metodologia baseia-se nos conceitos do PMI, a proposta está totalmente alinhada e incorporou documentos padronizados da própria metodologia da Ford para o registro e acompanhamento dos riscos.

Esta proposta foi apresentada à Diretoria e Gerência de TI da Ford para avaliação de sua real implantação. A avaliação da proposta foi bastante positiva, sendo a sua utilização

recomendada para todos os projetos do *IT Business Plan* juntamente com a implantação da metodologia SDM, com a seguinte estrutura:

- Na fase de Identificação e Levantamento da metodologia SDM:
  - Identificar riscos
    - Conduzir reuniões com os envolvidos (usuários, desenvolvedores, áreas de suporte de TI, gerente do projeto, supervisão e gerência de TI).
    - Discutir a visão de sucesso do projeto, incluindo os objetivos de TI.
    - Utilizar como ferramenta o *checklist* do TBQ, adaptado conforme a categoria do projeto:
      - Cat. 0 e 1 – questionário resumido do TBQ
      - Cat. 2 e 3 – questionário completo do TBQ
  - Analisar e priorizar riscos
    - Utilizar a matriz de referência da exposição ao risco do SEI, tabela 3, para classificar os riscos.
    - Identificar a lista com os *Top-10* riscos, registrando-os no documento padrão da metodologia SDM (*Risk Management Log*).
  - Planejar a resposta aos riscos
    - Desenvolver ação, responsável e data para os riscos *Top-10*.
    - Incluir estas ações no plano maior do projeto.
- Durante as demais fases da metodologia do SDM:
  - Acompanhar e controlar os riscos
    - Incorporar a revisão dos *Top-10* riscos nas reuniões periódicas de revisão de status do projeto.
    - Revisar o documento *Risk Management Log*, atualizando a relação de riscos e os planos de resposta ao risco.
  - Revisar a identificação de riscos quando de uma mudança importante no projeto

## 6.5 Conclusões gerais

O Gerenciamento de Riscos é uma necessidade para os projetos de desenvolvimento de sistemas devido à complexidade do ambiente em que estes projetos são desenvolvidos. A prática do gerenciamento de riscos deve reduzir as surpresas durante o processo de desenvolvimento de sistemas e ajudar no aumento da taxa de sucesso nos projetos.

O Gerenciamento de Riscos utiliza algumas técnicas bastante simples para identificar os riscos, requerendo apenas tempo e dedicação do time do projeto para esta tarefa e que, após a identificação, o planejamento das repostas aos riscos e seu acompanhamento devem acontecer de forma quase natural. O acompanhamento dos riscos nas reuniões de revisão de status do projeto também leva a um foco nos itens mais importantes do projeto e onde a gerência pode efetivamente auxiliar.

A condução da etapa de Planejamento do gerenciamento de riscos é importante para se estabelecer uma estratégia padrão para a condução do gerenciamento de riscos e também para se definir o conjunto de padrões que serão utilizados neste gerenciamento: fontes de risco e parâmetros de avaliação de probabilidade e impacto. Este conjunto de padrões é a base para prover conhecimentos a serem reutilizados em futuros projetos.

Além das ações apontadas anteriormente para a implantação do Gerenciamento de Risco, recomenda-se utilizar a Taxonomia do SEI para a classificação dos riscos identificados e se criar um banco de dados dos riscos de cada projeto e as estratégias e ações utilizadas para mitigar estes riscos. Outra fonte de dados pode ser as análises realizadas ao longo do projeto e no seu encerramento, onde lições aprendidas podem avaliar a efetividade do gerenciamento de riscos.

Este banco de dados seria uma importante ferramenta de gerenciamento do conhecimento de riscos da empresa e poderia ser utilizado a cada novo projeto. Assim, a experiência de projetos do passado poderia ser transmitida e utilizada por pessoas menos experientes na condução de novos projetos similares, o conhecimento antes tácito passa a ser explicito.

Além desta função, o banco de dados poderia ser utilizado para se identificar as situações de riscos mais freqüentes nos vários projetos e se desenvolver ações para eliminar a origem destes riscos recorrentes. Assim, de uma forma pró-ativa, se buscaria a causa raiz da origem destes riscos e ações seriam tomadas para eliminar a causa, evitando que o risco precisasse ser gerenciado em projetos futuros.

## 6.6 Propostas para trabalhos futuros

Para continuar desenvolvendo o conhecimento do tema discorrido nesta dissertação, sugere-se os seguintes temas cuja discussão foi limitada neste trabalho:

- Avaliar os resultados obtidos pela implantação do gerenciamento de risco e a metodologia SDM na Ford
- Desenvolver estudo sobre estratégias eficientes e mais utilizadas para resposta aos riscos mais freqüentes no ambiente de TI.
- Desenvolver estudo sobre o lado comportamental e de mudança cultural das organizações para facilitar a implantação da cultura do gerenciamento de risco.
- Desenvolver estudo para aprofundar as estratégias para a redução de riscos relacionados à definição dos Requisitos de sistemas e dos riscos ligados aos usuários.
- Desenvolver estudo para identificar os riscos mais freqüentes e identificar ações que poderiam eliminar a causa raiz da origem destes riscos.

# Apêndice A – Identificação de riscos baseados em uma taxonomia

Este Apêndice é baseado na metodologia de identificação de riscos baseada em uma taxonomia, apresentada por Carr (1993) e que é base da estrutura de gerenciamento de risco do SEI.

Carr (1993), contextualiza os riscos em projetos de desenvolvimento de sistema da seguinte forma:

1. Risco é parte inerente do desenvolvimento de sistemas
2. Práticas atuais de gerenciamento de risco são *ad-hoc*
3. Comunicação é pobre e inibida
4. É necessário um método sistemático e disciplinado de gerenciamento de risco para controlar a qualidade, custo e cronograma do projeto
5. O paradigma do SEI é a estrutura para o gerenciamento de risco
6. A identificação do risco é a base para o gerenciamento de risco

A maior parte dos riscos do projeto é normalmente conhecida pelas pessoas do projeto (apesar de em muitos casos o termo risco não ser usado para descrevê-los) e como conseqüência podem ser levantados e gerenciados. Normalmente o time do projeto identifica o risco como um sentimento desconfortável, uma preocupação ou uma dúvida sobre algum aspecto do sistema que eles estão desenvolvendo. Porém devido aos problemas para a comunicação estes riscos não são levados à atenção da gerência do projeto.

Identificação de risco é a base para o gerenciamento de riscos, pois sem um método efetivo e repetível de levantamento de riscos, o verdadeiro gerenciamento de riscos torna-se impossível, uma vez que não se pode gerenciar o que não se conhece. Ao manter esta abordagem, o método de identificação descrito também começa a endereçar o problema de comunicação que é central ao gerenciamento de riscos.

## A.1 – A metodologia de identificação de riscos

O foco do processo de identificação de riscos é nos riscos conhecidos (que podem ou não ter sido comunicados à gerência do projeto) e nos riscos desconhecidos. O método de

identificação de riscos é baseado na interdependência da ferramenta do TBQ e o seu processo de aplicação.

O método de identificação de risco do SEI é baseado nas seguintes premissas:

1. Os riscos ao desenvolvimento de sistemas são normalmente conhecidos do time técnico do projeto, mas são comunicados pobremente.
2. É preciso um método estruturado e repetível de identificação de riscos para gerenciar os riscos consistentemente
3. A identificação efetiva de riscos deve cobrir todas as áreas chave do desenvolvimento e suporte do projeto
4. O processo de identificação de riscos deve criar e manter um ambiente adequado de levantamento de riscos
5. Não se pode fazer um julgamento sobre o sucesso ou falha do projeto baseado apenas no número e natureza dos riscos levantados.

A taxonomia desenvolvida pelo SEI mapeia as características do desenvolvimento de sistemas e conseqüentemente dos riscos ao desenvolvimento de sistemas. O TBQ consiste de uma lista de questões para levantar problemas, preocupações (potenciais riscos) e riscos em cada grupo da taxonomia. A aplicação do processo é desenhada para assegurar que as questões são direcionadas para as pessoas certas e da maneira correta para produzir resultados ótimos.

Este método permite identificar os riscos sem justificativa e sem uma solução proposta, sendo assim o primeiro e necessário passo para estabelecer a comunicação dentro da organização. O método do TBQ pode ser descrito como uma forma estruturada de *brainstorming*.

O método de identificação de riscos levanta e clarifica as incertezas e preocupações do pessoal técnico e gerencial do projeto. Este time está perto dos problemas em seu nível e tem a experiência e o conhecimento para reconhecer potenciais problemas técnicos, de procedimento e contratuais.

## A.2 – A Taxonomia

A taxonomia fornece uma estrutura para organizar e estudar a abrangência dos problemas de desenvolvimento de sistemas. Ela serve como base para levantar e organizar todo o escopo de

riscos ao desenvolvimento de sistemas, sejam eles técnicos ou não-técnicos. A taxonomia também fornece uma estrutura consistente para o desenvolvimento de outros métodos e atividades de gerenciamento de riscos.

A taxonomia é organizada em três classes principais:

1. Engenharia do produto – os aspectos técnicos do trabalho que está sendo desenvolvido.
2. Ambiente de desenvolvimento – os métodos, procedimentos e ferramentas usados para desenvolver o produto.
3. Restrições ao projeto – fatores contratuais, organizacionais e operacionais dentro do contexto em que o sistema é desenvolvido, porém que são geralmente fora do controle direto da gerência local.

Estas classes são novamente dividas em elementos e cada elemento é caracterizado por seus atributos. Esta estrutura pode ser vista na figura 13.

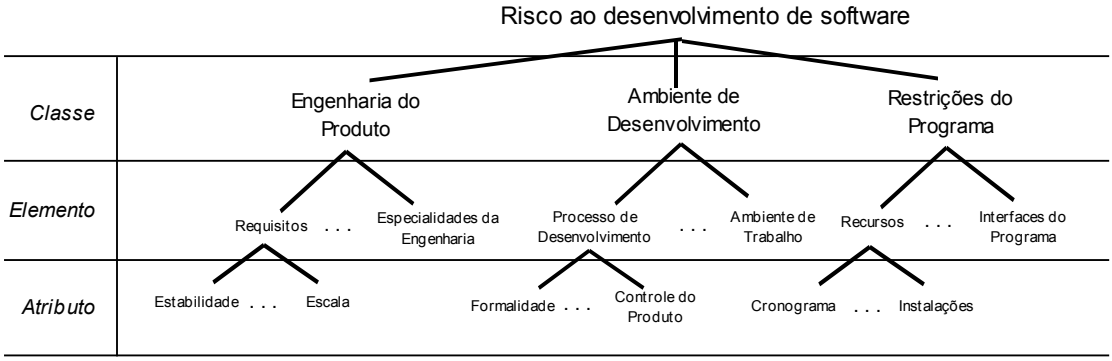


**Figura 13** – Estrutura da taxonomia de risco do SEI

**Fonte**: Carr (1993)

As classes e seus elementos apresentados a seguir são explicados em detalhe no Apêndice B.

## A. Engenharia do Produto

**1. Requisitos**

a) Estabilidade
b) Totalidade
c) Clareza
d) Validade
e) Viabilidade
f) Precedência
g) Escala

**2. Design**

a) Funcionalidade
b) Dificuldade
c) Interfaces
d) Performance
e) Testabilidade
f) Restrições de hardware
g) Non-developmental Software

**3. Codificação e teste unitário**

a) Viabilidade
b) Teste unitário
c) Codificação/Implementação

**4. Teste e Integração**

a) Ambiente
b) Produto
c) Sistema

**5. Especialidades de engenharia**

a) *Maintainability*
b) Confiabilidade
c) Segurança
d) Proteção
e) Fatores Humanos
f) Especificações

## B. Ambiente de desenvolvimento

**1. Processo de desenvolvimento**

a) Formalidade
b) Adequação
c) Controle do Processo
d) Familiaridade
e) Controle do produto

**2. Sistema de desenvolvimento**

a) Capacidade
b) Adequação
c) Usabilidade
d) Familiaridade
e) Confiabilidade
f) Suporte ao sistema
g) Capacidade de entrega

**3. Processo de gerenciamento**

a) Planejamento
b) Organização do projeto
c) Experiência dos gerentes
d) Interfaces do programa

**4. Métodos de gerenciamento**

a) Monitoração
b) Gerenciamento do pessoal
c) Garantia da qualidade
d) Gerenciamento de configurações

**5. Ambiente de trabalho**

a) Atitude com a qualidade
b) Cooperação
c) Comunicação
d) Moral

## C.Restrições do programa

**1. Recursos**

a) Cronograma
b) Staff
c) Orçamento
d) Instalações

**2. Contrato**

a) Tipo do contrato
b) Restrições
c) Dependências

**3. Interfaces do programa**

a) Cliente
b) Contratantes associados
c) Sub-contratados
d) *Prime contractor*
e) *Corporate Management*
f) Fornecedores
g) Política

## A.3 – O Questionário – TBQ

O TBQ consiste de questões ao nível do atributo em conjunto com dicas específicas e questões investigativas. Como o TBQ é abrangente, ele contém questões que podem não ser relevantes para todos os estágios do ciclo de desenvolvimento de sistemas, para alguns domínios específicos, ou para organizações específicas. Tipicamente, o questionário deve ser adaptado para cada projeto em particular e para o estágio no ciclo de desenvolvimento de sistemas excluindo questões não relevantes.

O Questionário completo é apresentado no Apêndice C e um questionário resumido, apresentado por Pandelios (1999), está desenvolvido no Apêndice D.

### A.3.1 – Método

O objetivo do método é encontrar um balanço entre a necessidade de uma completa identificação dos riscos contra os custos em termos da demanda de tempo do time do projeto.

As atividades para condução do método podem ser vistas na figura 14.

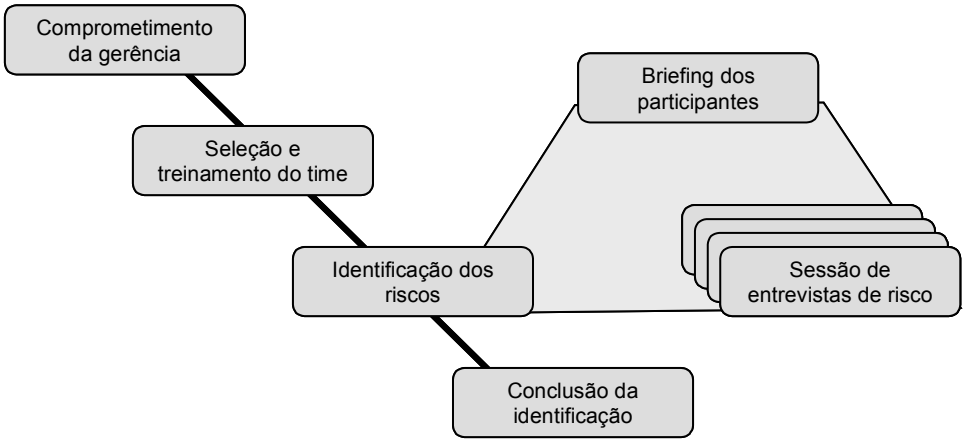


**Figura 14** – Processo de identificação de riscos

**Fonte**: Carr (1993)

### A.3.2 – Comprometimento da Gerência

Carr (1993) recomenda que três etapas sejam seguidas antes de se iniciar a identificação do risco: comprometimento dos executivos, seleção do projeto e seleção dos participantes.

Comprometimento dos executivos: a experiência com os testes de campo mostrou que a falta de uma aceitação séria dos benefícios da identificação dos riscos pelos gerentes do projeto e por seus superiores imediatos levou a atrasos significativos e até a cancelamentos da identificação. Por isso, o primeiro passo é apresentar um *briefing* executivo para obter o compromisso dos executivos. Este *briefing* deve fornecer uma visão geral do processo com detalhes do pessoal envolvido e do custo para o projeto, além de apresentar os benefícios de conduzir a identificação de riscos.

Seleção do projeto: após a obtenção do comprometimento dos executivos, o próximo passo é selecionar o projeto. Dois critérios devem ser seguidos: o gerente de projeto deve enxergar os benefícios de fazer a identificação de riscos e o projeto deve ter conteúdo significativo de *software*.

Seleção dos participantes das sessões de entrevistas: a experiência do SEI recomenda a seguinte formatação dos grupos de entrevistas:

- Cada grupo de entrevista deve ser constituído apenas de pares: o protocolo das entrevistas de identificação de risco requer um ambiente onde incertezas, preocupações, problemas e riscos podem ser levantados sem medo da atribuição ou da necessidade de uma solução.
- Divisão em quatro grupos: técnico, desenvolvedores, funções de suporte (qualidade, teste, integração, etc) e gerência do projeto. Para a abrangência da identificação de riscos é importante selecionar participantes que forneçam cobertura vertical e horizontal adequadas.
- Os grupos devem ser limitados a cinco participantes: por diversas razões (dinâmica de grupo, follow-up, etc).

### A.3.3 – Seleção do time e treinamento

Todo o time deve ser treinado antes das sessões de identificação de riscos. O treinamento deve cobrir o TBQ, papéis, processos e o protocolo das entrevistas. Um aspecto chave do treinamento é fornecer conhecimento específico sobre o projeto em questão, o que pode ser desenvolvido a partir de um *briefing* dado pelo pessoal do projeto.

### A.3.4 – Identificação do risco

O processo de identificação do risco consiste de uma série de entrevistas com os grupos de pessoas do projeto selecionadas. Cada sessão de entrevista tem duas partes:

1. Perguntas e respostas: este segmento envolve o uso do TBQ e perguntas investigativas para levantar problemas, preocupações ou riscos que podem ameaçar o sucesso do projeto.
2. Explicação dos problemas: este segmento envolve a explicação das palavras e do significado dos problemas identificados no segmento de Perguntas e Respostas, através da classificação consensual dos riscos nos grupos da taxonomia no nível de elemento das classes. Após a classificação dos problemas, os participantes avaliam os que são essencialmente equivalentes para agrupá-los em uma única descrição.

### A.3.5 – Conclusão da identificação

Ao concluir a identificação do risco, é necessário dar um retorno a todos os participantes. Isto é feito apresentando um sumário dos resultados, contento a lista de todos os problemas identificados e algumas sugestões de como gerenciá-los.

Enquanto que o passo do relatório conclui o processo de identificação de risco, ela representa apenas o início da estratégia de gerenciamento de risco do projeto. Uma cópia dos resultados deve ser deixada exclusivamente com o gerente do projeto, como forma de manter com ele o controle da situação, pois a comunicação prematura desta informação pode expor o projeto a uma atenção negativa injustificada causada pela noção de que qualquer projeto que tenha uma lista de riscos está em problemas e precisa de ajuda urgente.

O processo descrito acima envolveu mais de 15 testes de campo, cada um trazendo refinamentos significativos para o processo. **O SEI não recomenda o processo como sendo o único método capaz de identificar riscos efetivamente, porém ele identificou o processo como sendo eficiente e efetivo.**

## A.4 – Recomendações

1. O processo é importante – as recomendações de tamanho do grupo de entrevistas e de somente serem constituídos de pares deve ser seguida. Caso o tempo da sessão seja curto, recomenda-se definir o início do questionário baseado no grupo que será

entrevistado (desenvolvedores – engenharia de produto, engenheiros especialistas – ambiente de desenvolvimento, gerente de projeto – restrições do programa).

2. Riscos mudam com o tempo (probabilidade, impacto e período de tempo) – o projeto deve repetir o processo de identificação de risco e os processos de follow-up periodicamente ao longo do ciclo de desenvolvimento do sistema.
3. Para organizações que estão adquirindo sistemas – uma identificação formal de riscos deve ser feita durante a fase de definição do conceito da aquisição para determinar os potenciais riscos ao projeto. Este conhecimento pode ser usado para reduzir os riscos ou para avaliar as propostas dos fornecedores para lidar com estes riscos.
4. Para organizações desenvolvedoras – uma identificação formal de riscos deve ser feita durante a fase de proposta do projeto e posteriormente nos principais *milestones*. Um processo informal de identificação de riscos deve ser previsto e executado mais freqüentemente. Em todos os casos o fornecedor deve reportar ao cliente os riscos que podem trazer um perigo significativo ao projeto.

# Apêndice B – Definições dos grupos da taxonomia

Este Apêndice é uma tradução feita pelo autor das definições de grupos da taxonomia apresentada por Carr (1993). Alguns termos foram mantidos pelo autor em inglês devido à sua larga utilização no domínio da tecnologia da informação e da dificuldade de tradução para um termo semelhante em português.

## A. Engenharia do Produto

A engenharia do produto refere-se às atividades de engenharia de sistemas e software envolvidos na criação de um sistema que satisfaça os requisitos e expectativas do cliente. Estas atividades incluem análise e especificação de sistemas, desenho e implantação do *software*, integração dos componentes de *software* e *hardware* e testes do sistema.

Os elementos desta classe cobrem as atividades tradicionais da engenharia de *software*. Ela compreende os fatores técnicos associados à entrega do produto em si, independentemente dos processos e ferramentas usados para produzi-lo ou das restrições impostas pelo número finito de recursos ou fatores externos fora do controle do projeto.

Os riscos da engenharia do produto geralmente resultam de:

- Requisitos que são tecnicamente difíceis ou impossíveis de serem atingidos, freqüentemente em combinação com a incapacidade de negociar requisitos mais relaxados ou revisar cronogramas e orçamentos
- Análise dos requisitos ou desenho da especificação inadequados
- Qualidade pobre no desenho ou codificação das especificações

### 1. Requisitos

Os atributos do elemento requisitos cobrem tanto a qualidade da especificação dos requisitos como também a dificuldade de implementar um sistema que satisfaça os requisitos.

**a) Estabilidade**

O atributo da estabilidade refere-se ao grau com que os requisitos estão mudando e o possível efeito que as mudanças nos requisitos e interfaces externas podem ter na qualidade, funcionalidade, cronograma, design, integração e testes do produto que está sendo construído.

O atributo também abrange os problemas que surgem da incapacidade de se controlar requisitos que estão mudando rapidamente. Por exemplo, análises de impacto podem se tornar imprecisas porque é impossível se definir a referência base contra a qual as mudanças serão implementadas.

**b) Totalidade**

Requerimentos especificados de maneira incompleta podem aparecer de muitas formas

- como um documento de requisitos com muitas funções ou parâmetros "a definir"
- requisitos que não são especificados adequadamente para desenvolver um critério de aceite ou requisitos omitidos inadvertidamente

Quando informações faltantes não são fornecidas em tempo hábil, a implementação pode ser baseada em suposições do fornecedor que podem diferir das expectativas do consumidor.
Quando as expectativas do consumidor não são documentadas na especificação, elas não são orçadas quanto a custo e cronograma.

**c) Clareza**

Este atributo se refere a requisitos escritos de maneira ambígua ou imprecisa e que não são esclarecidos até muito tarde na fase de desenvolvimento. Esta falta de entendimento mútuo entre o fornecedor e o cliente pode requerer re-trabalho para atender a intenção do requisito do cliente.

**d) Validade**

Este atributo se refere a se os requisitos como um todo refletem as intenções do cliente para o produto. Pode ser afetado por falhas no entendimento dos requisitos escritos pelo fornecedor ou cliente, requisitos ou expectativas não descritas ou uma especificação onde o usuário final não forneceu *inputs*.
Este atributo é afetado pelos atributos de clareza e totalidade das especificações dos requisitos, porem se refere à questão mais ampla se o sistema como um todo atende as intenções do cliente.

**e) Viabilidade**

O atributo de viabilidade se refere à dificuldade de implementar um requisito técnico ou operacional específico, ou de simultaneamente atender requisitos conflitantes. Algumas vezes dois requisitos isoladamente são viáveis, porém em conjunto não; eles não podem existir ao mesmo tempo no mesmo produto.

Também está incluída aqui a habilidade de determinar um método adequado de qualificação para demonstrar que o sistema satisfaz os requisitos.

**f) Precedência**

O atributo da precedência é relativo a capacidades que ainda não foram implementadas com sucesso em nenhum sistema existente ou estão além da experiência do pessoal ou da própria companhia. O grau de risco depende

- da alocação de verba e tempo adicional para determinar a possibilidade de sua implementação
- de planos de contingência no caso dos requisitos não serem viáveis
- da flexibilidade no contrato para alocar verbas e tempo de implementação baseados no resultado do estudo de viabilidade

Mesmo que requisitos sem precedência anterior sejam viáveis, ainda existe o risco de subestimar a dificuldade de implantação e do comprometimento com um cronograma e orçamento inadequados.

**g) Escala**

Este atributo cobre tanto os desafios técnicos como de gerenciamento apresentados pelo desenvolvimento de sistemas amplos e complexos.

Desafios técnicos incluem a satisfação com os requisitos de tempo de resposta, comunicação entre processadores, complexidade de integração entre sistemas, análise das dependências entre componentes e impacto devido às mudanças nos requisitos.

O gerenciamento de um grande número de tarefas e pessoas introduz a complexidade em áreas como organização do projeto, delegação de responsabilidades, comunicação entre gerentes e pares e gerenciamento de configuração.

## 2.*Design*

Os atributos do elemento *design* cobrem o *design* e a viabilidade de requisitos de algoritmos, funções e performance, e as interfaces internas e externas do produto. A dificuldade nos testes pode começar aqui, falhando ao obter requisitos que sejam testáveis ou ao incluir características de teste no *design*.

**a) Funcionalidade**

Este atributo cobre os requisitos funcionais que podem não gerar um *design* viável, ou o uso de algoritmos específicos ou *designs* com um alto grau de incerteza quanto a poder satisfazer

seus requisitos originais. Estudos de *design* e algoritmos podem não ter usado técnicas apropriadas de investigação ou podem ter uma viabilidade muito pequena.

**b) Dificuldade**

O atributo da dificuldade se refere a requisitos funcionais ou de *design* que podem ser extremamente difíceis de realizar. A engenharia de sistemas pode desenhar uma arquitetura difícil de implementar ou a análise dos requisitos pode ter sido baseada em suposições otimistas do *design*.

O atributo da dificuldade difere da viabilidade do design uma vez que ele não procede de algoritmos e designs pré-dispostos.

**c) Interfaces**

Este atributo cobre todas as interfaces de *hardware* e *software* que estão no escopo do programa de desenvolvimento, incluindo as interfaces entre itens de configuração e as técnicas para definir e gerenciar as interfaces. Uma atenção especial é dada ao *non-developmental software* (NDS) (software não desenvolvido pelo projeto) e a interfaces de *hardware* que serão desenvolvidas.

**d) Performance**

O atributo da performance se refere à performance crítica para o tempo

- requisitos de uso e tempo de resposta *real-time*
- requisitos de *throughput*
- análises de performance
- modelagem da performance ao longo do ciclo de desenvolvimento

**e) Testabilidade**

O atributo da testabilidade cobre a receptividade do *design* aos testes, o desenho de características para facilitar os testes e a inclusão no processo de *design* de pessoas que irão desenhar e conduzir os testes do produto.

**f) Restrições de *hardware***

Este atributo cobre o *hardware* em relação à arquitetura do sistema e do processador, e da dependência do *hardware* para atingir os requisitos de performance do software e do sistema. Estas restrições podem incluir a capacidade de velocidade da memória, capacidade de repostas *real-time*, limitações de acesso ou capacidade do banco de dados, confiabilidade insuficiente ou insuficiência no tamanho do *hardware* especificado.

**g) *Non-developmental Software***

Como o *non-developmental software* não é desenhado de acordo com os requisitos do sistema, mas selecionados como a melhor escolha, ele pode não suportar exatamente os requisitos de performance, operação e suporte.

O cliente pode não aceitar os dados de confiabilidade e de teste gerados pelo vendedor ou desenvolvedor para demonstrar a satisfação dos requisitos alocados ao NDS. Neste caso torna-se difícil de produzir estes dados para satisfazer o critério de aceite e dentro do orçamento estimado para teste do NDS.

Mudanças nos requisitos podem necessitar de uma re-engenharia ou confiança nos vendedores para *upgrades* por motivos especiais.

## 3. Codificação e teste unitário

Os atributos deste elemento estão associados com a qualidade e estabilidade das especificações do *software* ou das interfaces e com as restrições que podem apresentar dificuldades de teste ou implementação.

**a) Viabilidade**

O atributo da viabilidade do elemento de codificação e teste unitário endereça possíveis dificuldades que podem surgir de um desenho ou especificação de desenho pobre ou de necessidades de difícil implementação.

Por exemplo:

- O *design* pode não ter atributos de qualidade como coesão dos módulos ou minimização das interfaces
- O tamanho dos módulos pode contribuir para a complexidade
- O design pode não ter sido especificado com os detalhes suficientes, requerendo que o programador faça suposições ou tome decisões durante a codificação
- As especificações do *design* ou das interfaces podem mudar, talvez sem a aprovação de uma referência base detalhada do design
- O uso de um *developmental hardware* pode trazer uma contribuição adicional para especificações inadequadas ou instáveis de interfaces
- A natureza do sistema pode agravar a dificuldade e complexidade da tarefa de codificação

**b) Teste unitário**

Fatores que afetam o teste unitário incluem o planejamento e a preparação e também os recursos e tempo alocados para o teste.

Componentes destes fatores são:

- Iniciar os testes unitários com a qualidade do código obtida de procedimentos formais ou informais de inspeção e verificação do código
- *Test cases* pré-planejados que foram confrontados com os requisitos de teste unitário
- Bancada de testes composta dos *hardware*, emuladores, *software* e simuladores necessários
- Dados de teste que satisfazem os testes planejados
- Tempo alocado suficiente para planejar e executar o plano de teste

**c) Codificação/Implementação**

Este atributo endereça as implicações de restrições de implementação. Alguns deste são

- *Hardware* que é inadequado com respeito à velocidade, arquitetura, tamanho da memória ou capacidade de armazenamento externo
- Métodos ou linguagens de implementação requeridos
- Diferenças entre o *hardware* final e o de desenvolvimento

## 4.Teste e Integração

Este elemento cobre o planejamento, execução e instalações dos testes e integração tanto para o produto contratado como para a integração do produto no sistema ou ambiente.

**a) Ambiente**

O ambiente de teste e integração inclui as instalações de *hardware* e *software* e os *test cases* adequados refletindo cenários operacionais reais e dados e condições de teste reais.

Este atributo endereça a adequação deste ambiente para permitir a integração em um ambiente realístico ou para testar completamente todos os requisitos funcionais e de performance.

**b) Produto**

O atributo da integração do produto se refere à integração dos componentes de *software* uns com os outros e com o *hardware* final, e os testes do produto entregue contratualmente. Fatores que podem afetar são:

- Especificações internas das interfaces para *hardware* ou *software*
- Testabilidade dos requisitos
- Negociação de um acordo sobre o critério de teste com o cliente

- Adequação das especificações de teste
- Tempo suficiente para integração e teste

**c) Sistema**

O atributo da integração do sistema se refere à integração do produto contratual com sistemas relacionados. Fatores associados com este atributo são:

- Especificações externas de interface
- Habilidade de reproduzir condições da interface do sistema antes da integração do sistema
- Acesso ao sistema que enviará/receberá a interface
- Adequação de tempo para testes

## 5. Especialidades de engenharia

Os requisitos das especialidades de engenharia são tratados separadamente do elemento dos requisitos gerais principalmente porque eles são normalmente endereçados por especialistas que podem não estar dedicados todo o tempo para o projeto. Esta separação taxonômica é uma forma de assegurar que estes especialistas sejam chamados para analisar os riscos associados com as suas áreas de expertise.

**a) *Maintainability***

A *maintainability* pode ser prejudicada por uma arquitetura de software, desenho, codificação ou documentação pobres resultantes de padrões não documentados ou não reforçados ou da negligência em analisar o sistema de um ponto de vista de manutenção.

**b) Confiabilidade**

Os requisitos de confiabilidade ou de disponibilidade do sistema podem ser afetados por um *hardware* que não atinge suas especificações de confiabilidade ou complexidade do sistema que agrava as dificuldades em atingir os tempos de recuperação. Requisitos de confiabilidade ou disponibilidade alocados para o *software* podem ter sido definidos em termos absolutos, ao invés de separados do *hardware* e testados independentemente.

**c) Segurança**

Este atributo endereça a dificuldade de implementação de requisitos de segurança e também a potencial dificuldade de demonstrar satisfatoriamente os requisitos através da simulação de condições inseguras e de ações corretivas. Uma demonstração completa pode não ser possível até que o sistema esteja instalado e operacional.

**d) Proteção**

Este atributo endereça a falta de experiência na implementação dos níveis requeridos de proteção ao sistema que podem resultar em subestimação do esforço necessário para métodos rigorosos de verificação, certificação e aprovação, e para logísticas seguras do processo de desenvolvimento; dependência na entrega de *hardware* ou *software* certificados.

**e) Fatores Humanos**

Atender os requisitos de fatores humanos depende do entendimento do ambiente operacional do sistema instalado e do acordo com vários clientes e grupos de usuários com um entendimento mútuo das expectativas incorporadas nos requisitos dos fatores humanos. É muito difícil transformar este entendimento em uma especificação escrita. O acordo mútuo na interface humana pode requerer uma prototipagem contínua e a demonstração para vários grupos de clientes.

**f) Especificações**

Este atributo endereça as especificações para o sistema, *hardware*, *software*, interface ou requisitos de *design* ou teste em qualquer nível com respeito à viabilidade da implementação e os atributos de estabilidade, totalidade, clareza e verificabilidade.

# B. Ambiente de desenvolvimento

A classe do ambiente de desenvolvimento endereça o ambiente de projeto e o processo usado na engenharia do sistema. Este ambiente inclui o processo e sistema de desenvolvimento, métodos de gerenciamento e ambiente de trabalho. Estes elementos de ambiente são caracterizados abaixo pelos seus atributos.

## 1. Processo de desenvolvimento

O elemento do processo de desenvolvimento se refere ao processo pelo qual o fornecedor se propõe a satisfazer os requisitos do cliente. O processo é uma seqüência de passos – os *inputs*, *outputs*, ações, critérios de validação e atividades de monitoração – indo da especificação inicial dos requisitos até a entrega do produto final. O processo de desenvolvimento inclui fases como análise dos requisitos, definição do produto, testes e entrega. Inclui tanto os processos genéricos de gerenciamento como levantamento de custos, acompanhamento do cronograma e indicação do pessoal, como processos específicos do projeto como estudos de viabilidade, revisões do *design* e testes de regressão.

Este elemento agrupa riscos que resultam de um processo de desenvolvimento que é mal planejado, definido e documentado; que não é apropriado para as atividades necessárias para atingir os objetivos do projeto; e que é pobremente comunicado para o *staff* e que não tem sua aplicação reforçada.

**a) Formalidade**

A formalidade do processo de desenvolvimento é uma função do grau com que um processo consistente é definido, documentado e comunicado em todos os aspectos e fases do desenvolvimento.

**b) Adequação**

Este atributo se refere à quão adequado o modelo de desenvolvimento, processo, métodos e ferramentas selecionados suportam o escopo e tipo de atividades requeridas para o programa específico.

**c) Controle do Processo**

O controle do processo se refere não somente a assegurar a utilização do processo definido, mas também à medição e melhorias no processo baseadas na observação com respeito aos objetivos de qualidade e produtividade. O controle pode se tornar complicado devido ao uso de locais de desenvolvimento distribuídos.

**d) Familiaridade**

A familiaridade com o processo de desenvolvimento cobre o conhecimento, experiência e conforto na utilização do processo prescrito.

**e) Controle do produto**

O controle do produto depende da rastreabilidade dos requisitos a partir da especificação inicial até a implantação de forma que o teste do produto possa demonstrar os requisitos iniciais. O processo de controle de mudanças faz uso do mecanismo de rastreabilidade em análises de impacto e reflete toda modificação resultante nas documentações, incluindo documentação das interfaces e de teste.

## 2. Sistema de desenvolvimento

O elemento do sistema de desenvolvimento endereça o hardware, as ferramentas de software e os equipamentos de suporte utilizados no desenvolvimento do produto. Ele inclui ferramentas *computer aided software engineering*, simuladores, compiladores, equipamentos de teste e sistemas de hospedagem de computadores.

**a) Capacidade**

Riscos associados com a capacidade do sistema de desenvolvimento podem resultar de uma quantidade pequena de estações de trabalho, capacidade de processamento ou de banco de dados insuficiente, ou outras inadequações nos equipamentos para suportar atividades paralelas para o desenvolvimento, teste e atividades de suporte.

**b) Adequação**

A adequação do sistema de desenvolvimento é associada ao grau em que ela suporta os modelos específicos de desenvolvimento, processos, métodos, procedimentos e atividades requeridas e selecionadas para o programa. Ela inclui os processos de desenvolvimento, gerenciamento, documentação e gerenciamento de configurações.

**c) Usabilidade**

A usabilidade se refere à documentação do sistema de desenvolvimento, sua acessibilidade e facilidade de uso.

**d) Familiaridade**

A familiaridade com o sistema de desenvolvimento depende de sua utilização anterior pela companhia e pelo pessoal do projeto assim com da adequação do treinamento para os novos usuários.

**e) Confiabilidade**

A confiabilidade do sistema de desenvolvimento é uma medida de quando os componentes do sistema de desenvolvimento estão disponíveis e funcionando corretamente sempre que requeridos por qualquer pessoa do programa.

**f) Suporte ao sistema**

Suporte ao sistema de desenvolvimento envolve o treinamento no uso do sistema, acesso a usuários *experts* ou consultores e reparo ou resolução de problemas por fornecedores.

**g) Capacidade de entrega**

Alguns contratos requerem a entrega do sistema de desenvolvimento. Riscos podem resultar da negligência na cotação e alocação de recursos para assegurar que o sistema de desenvolvimento atenda todos os requisitos de entrega.

# 3. Processo de gerenciamento

O elemento do processo de gerenciamento pertence aos riscos associados com o planejamento, monitoração e controle de orçamento e cronograma; com o controle dos fatores envolvidos na definição, implementação e testes do produto; com o gerenciamento do pessoal

do projeto; e com lidar com organizações externas, incluindo o cliente, gerência sênior, matriz de gerenciamento e outros contratados.

**a) Planejamento**

O atributo planejamento endereça riscos associados com o desenvolvimento de um plano bem definido que responde a contingências, assim como a objetivos de longo prazo e que foi formulado com o input e concordância daqueles afetados por ele. Também endereça o gerenciamento de acordo com o plano e a modificação formal do plano quando alterações são necessárias.

**b) Organização do projeto**

Este atributo endereça a efetividade da organização do programa, a efetividade da definição dos papéis e responsabilidades, e a garantia de que estes papéis e linhas de autoridade são compreendidas pelo pessoal do projeto.

**c) Experiência dos gerentes**

Este atributo se refere à experiência de todos os níveis de gerentes com respeito ao gerenciamento, gerenciamento do desenvolvimento de *software*, domínio da aplicação, escala e complexidade do sistema e programa, do processo de desenvolvimento selecionado e a codificação de *software*.

**d) Interfaces do programa**

Este atributo se refere às interações dos gerentes em todos os níveis com o pessoal do programa, e com pessoal externo como cliente, gerência sênior e gerentes pares.

# 4.Métodos de gerenciamento

Este elemento se refere à métodos para gerenciar tanto o desenvolvimento do produto como o pessoal do programa. Ele inclui garantia da qualidade, gerenciamento de configurações, desenvolvimento do *staff* com respeito às necessidades do programa e a manutenção da comunicação sobre o status do programa e suas necessidades.

**a) Monitoração**

A monitoração inclui atividades para obter relatórios de status e agir sobre eles, alocar informações de status para as organizações apropriadas do programa e manter e usar medidas de progresso.

**b) Gerenciamento do pessoal**

O gerenciamento do pessoal se refere à seleção e treinamento dos membros do programa e assegurando que eles: tomem parte no planejamento e na interação com os clientes em suas

áreas de responsabilidade; trabalhem de acordo com o plano; e recebam o suporte que eles precisem ou solicitem para cumprir com suas responsabilidades.

**c) Garantia da qualidade**

O atributo da garantia da qualidade se refere aos procedimentos instituídos para assegurar que os processos contratuais e padrões sejam implementados adequadamente para todas as atividades do programa, e também que a função de garantia de qualidade tenha o recurso de pessoal necessário para cumprir suas funções.

**d) Gerenciamento de configurações**

O atributo do gerenciamento de configuração (CM) endereça as ferramentas e pessoal para a função de CM e também a complexidade do processo de CM requerido com respeito a fatores como desenvolvimento múltiplo e instalação de *sites* e coordenação do produto com sistemas existentes e possivelmente em mudanças.

# 5. Ambiente de trabalho

O elemento do ambiente de trabalho se refere a aspectos subjetivos do ambiente como a atenção dada para assegurar que as pessoas se mantenham informadas sobre os objetivos do projeto, como a forma com que as pessoas trabalham juntas, a resposta aos inputs do *staff* e a atitude e o moral do pessoal do programa.

**a) Atitude com a qualidade**

Este atributo se refere a tendência do pessoal do programa de produzir um trabalho de qualidade em geral e para atender padrões de qualidade específicos para o programa e produto.

**b) Cooperação**

O atributo da cooperação endereça a falta de espírito de equipe entre o *staff* de desenvolvimento tanto entre como através de times de trabalho e também a falha em todos os níveis da gerência em demonstrar que os melhores esforços estão sendo feitos para remover barreiras à execução do trabalho.

**c) Comunicação**

Riscos que resultam de uma comunicação falha são resultantes da falta de conhecimento da missão do sistema, seus requisitos, métodos e objetivos do *design* ou falta de informação sobre a importância dos objetivos do programa para a companhia e para o projeto.

**d) Moral**

Riscos que resultam de uma baixa moral derivam de baixos níveis de entusiasmo e conseqüentemente de baixa performance, produtividade ou criatividade; irritação que pode resultar em dano intencional ao projeto ou produto; êxodo em massa do *staff* do projeto; e uma reputação da companhia que torne difícil o recrutamento.

# C. Restrições do programa

As restrições do programa se referem a elementos externos ao projeto. Estes são fatores que podem ser fora do controle do projeto, mas que ainda assim podem ter efeitos importantes em seu sucesso ou constituir fontes substanciais de risco.

## 1. Recursos

Este elemento endereça os recursos de que o programa depende, mas cuja obtenção e manutenção estão fora do controle do programa. Estes incluem prazos, *staff*, orçamento e instalações.

**a) Cronograma**

Este atributo se refere à estabilidade do cronograma com respeito aos eventos ou dependências interna e externa e da viabilidade das estimativas e planos para todas as fases e aspectos do programa.

**b) *Staff***

Este atributo se refere à estabilidade e adequação do *staff* em termos de quantidade e nível de competência, conhecimento em termos das áreas técnicas requeridas e domínio da aplicação e de sua disponibilidade quando preciso.

**c) Orçamento**

Este atributo se refere à estabilidade do orçamento com respeito a eventos ou dependências internas ou externas e da viabilidade das estimativas e planos para todas as fases e aspectos do programa.

**d) Instalações**

Este atributo se refere à adequação das instalações do programa para o desenvolvimento, integração e testes do produto.

## 2. Contrato

Riscos associados com o contrato do programa estão classificados de acordo com o tipo do contrato, suas restrições e dependências.

**a) Tipo do contrato**

Este atributo cobre as condições de pagamento e os requisitos do contrato associados com itens como o *statement of work*, lista de requisitos de dados do contrato, e o tamanho e condições do envolvimento do cliente.

**b) Restrições**

Restrições e limitações do contrato se referem a diretivas contratuais como, por exemplo, uso de métodos específicos de desenvolvimento ou equipamento e as complicações resultantes com a aquisição de direito de uso de *non-developmental software*.

**c) Dependências**

Este atributo se refere a possíveis dependências contratuais com fornecedores externos, equipamento ou software fornecido pelo cliente ou outros produtos e serviços externos.

## 3. Interfaces do programa

Este elemento consiste de várias interfaces com as entidades e organizações fora do programa de desenvolvimento em si.

**a) Cliente**

O atributo do cliente se refere ao nível de experiência e competência do cliente com o domínio técnico ou da aplicação, assim como com a dificuldade das relações de trabalho ou de mecanismos pobres para obter concordância e aprovação do cliente, não ter acesso a certos grupos de clientes ou não ser capaz de comunicar com o cliente de uma maneira direta.

**b) *Associate contractors* (Contratantes associados)**

A presença de contratantes associados pode introduzir riscos devido a agendas políticas conflitantes, problemas de interface entre os sistemas sendo desenvolvidos pelas organizações externas ou a falta de cooperação na coordenação de cronogramas e mudanças de configurações.

**c) Sub-contratados**

A presença de sub-contratados pode introduzir riscos devido à inadequação das tarefas definidas e dos mecanismos de gerenciamento do sub-contratado, ou na não transferência de tecnologia e conhecimento do sub-contratado para o programa ou corporação.

**d) *Prime contractor***

Quando o programa é sub-contratado de um programa maior, riscos podem surgir de tarefas mal definidas, combinações complexas de reporte ou dependências em informações técnicas ou de programa.

**e) *Corporate Management***

Riscos na área de *corporate management* incluem comunicação e direcionamento da gerência sênior pobres assim como níveis de suporte baixos.

**f) Fornecedores**

Riscos de fornecedores podem se apresentar na forma de dependências de entrega e suporte para componentes críticos do sistema.

**g) Política**

Riscos políticos podem surgir de relacionamentos entre companhias, clientes, contratantes associados ou sub-contratados e podem afetar as decisões técnicas.

# Apêndice C – TBQ - Questionário completo

Este Apêndice é uma tradução feita pelo autor do questionário completo do TBQ apresentado por Carr (1993), com o objetivo de ser utilizado pelos leitores como base para o desenvolvimento de um questionário adaptado para cada empresa ou projeto. Novamente, alguns termos foram mantidos pelo autor em inglês devido à sua larga utilização no domínio da tecnologia da informação e da dificuldade de tradução para um termo semelhante em português.

## A. Engenharia do produto

### 1. Requisitos

**a) Estabilidade**

*[os requisitos estão mudando mesmo com o produto sendo produzido?]*

[1] Os requisitos estão estáveis?

(não) (1.a) Qual o efeito no sistema?

- qualidade
- funcionalidade
- cronograma
- integração
- *design*
- testes

[2] As interfaces externas estão se alterando?

**b) Totalidade**

*[Existem requisitos faltando ou especificados de forma incompleta?]*

[3] Existe algum "a definir" nas especificações?

[4] Você conhece algum requisito que deveria estar nas especificações, mas que não está?

(sim) (4.a) Você será capaz de colocar estes requisitos no sistema?

[5] O cliente tem expectativas/requisitos que não estão escritos?

(sim) (5.a) Existe alguma forma de capturar estes requisitos?

[6] As interfaces externas estão completamente definidas?

**c) Clareza**

*[Existem requisitos que não estão claros ou que dependam de interpretação?]*

[7] Você consegue entender os requisitos como eles estão escritos?

(não) (7.a) As ambigüidades estão sendo resolvidas de forma satisfatória?

(sim) (7.b) Não existe nenhum problema de ambigüidade ou de interpretação?

**d) Validade**

*[Os requisitos levarão ao produto que o cliente tem em mente?]*

[8] Existe algum requisito que pode não estar especificando o que o cliente realmente quer?

(sim) (8.a) Como você está resolvendo isto?

[9] Você e o cliente entendem os requisitos da mesma forma?

(sim) (9.a) Existe algum processo para determinar este entendimento?

[10] Como você valida os requisitos?

- Prototipagem
- Análise
- Simulação

**e) Viabilidade**

*[Os requisitos são inviáveis de um ponto de vista analítico?]*

[11] Existe algum requisito que seja tecnicamente difícil de implementar?

(sim) (11.a) Quais são?

(sim) (11.b) Por que são difíceis de implementar?

(não) (11.c) Foram feitos estudos de viabilidade para estes requisitos?

(sim) (11.c.1) Quão confiante você está nas premissas feitas nestes estudos?

**f) Precedência**

*[Os requisitos especificam algo que nunca foi feito antes ou algo que a companhia nunca tenha feito antes?]*

[12] Existe algum requisito *state-of-the-art*?

- Tecnologias
- Métodos
- Linguagens

- *Hardware*

(não) (12.a) Algum deles é novo para você?

(sim) (12.b) O programa tem conhecimento suficiente nestas áreas?

(não) (12.b.1) Existe um plano para adquirir conhecimento nesta área?

**g) Escala**

*[Os requisitos especificam um produto mais amplo, mais complexo ou que requer uma organização maior do que a experiência anterior da companhia?]*

[13] O tamanho e complexidade do sistema são uma preocupação?

(não) (13.a) Você já fez algo deste tamanho ou complexidade antes?

[14] O tamanho do sistema requer uma organização maior do que a usual para a companhia?

## 2. *Design*

**a) Funcionalidade**

*[Existe algum problema potencial para atender os requisitos de funcionalidade?]*

[15] Existe algum algoritmo especificado que possa não satisfazer os requisitos?

(não) (15.a) Algum dos algoritmos ou *design* atinge de forma marginal os requisitos?

[16] Como você determina a viabilidade de algoritmos e *designs*?

- Prototipagem
- Modelagem
- Análise
- Simulação

**b) Dificuldade**

*[Será difícil de atingir o design e/ou a implementação?]*

[17] Algum *design* depende de premissas otimistas ou irreais?

[18] Algum dos requisitos ou funções é difícil de desenhar?

(não) (18.a) Você tem soluções para todos os requisitos?

(sim) (18.b) Quais são estes requisitos?

- Por que eles são difíceis?

**c) Interfaces**

*[As interfaces internas (hardware e software) estão bem definidas e controladas?]*

[19] As interfaces internas estão bem definidas?

- De *software* para *software*
- De *software* para *hardware*

[20] Existe um processo para definir interfaces internas?

(sim) (20.a) Existe um processo de controle de mudança para interfaces internas?

[21] O hardware está sendo desenvolvido em paralelo com o *software*?

(sim) (21.a) As especificações do *hardware* estão sendo alteradas?

(sim) (21.b) Todas as interfaces com o *software* já forma definidas?

(sim) (21.c) Existirão modelos de desenho de engenharia que poderão ser usados para testar o *software*?

**d) Performance**

*[Existem requisitos estritos de tempo de resposta ou de throughput?]*

[22] Existe algum problema com a performance?

- *Throughput*
- Agendamento assíncrono de eventos real-time
- Tempo de resposta real-time
- Tempos de recuperação
- Tempo de resposta
- Acesso, contenção ou resposta do banco de dados

[23] Foi feita uma análise de performance?

(sim) (23.a) Qual o seu nível de confiança na análise de performance?

(sim) (23.b) Você tem um modelo para monitorar a performance ao longo do *design* e implementação?

**e) Testabilidade**

*[O produto é difícil ou impossível de testar?]*

[24] O *software* será fácil de testar?

[25] O *design* inclui recursos para facilitar os testes?

[26] Os testadores foram envolvidos na análise dos requisitos?

**f) Restrições de *hardware***

*[Existem restrições limitadas no hardware planejado?]*

[27] O *hardware* limita sua capacidade de atingir os requisitos?

- Arquitetura
- Capacidade de memória
- *Throughput*
- Resposta *real-time*
- Tempo de resposta
- Tempos de recuperação
- Performance do banco de dados
- Funcionalidades
- Confiabilidade
- Disponibilidade

**g) *Non-developmental software***

*[Existem problemas com o software usado no programa mas que não é desenvolvido pelo programa?]*

- **Caso exista software re-utilizado ou *reengineered***

[28] Você está reusando ou reengenheirando algum *software* não desenvolvido pelo programa?

(sim) (28.a) Você prevê algum problema?

- Documentação
- Performance
- Funcionalidade
- Entrega em tempo
- Adaptação

- **Caso esteja sendo utilizado algum software COTS (*commercial off-the-shelf*)**

[29] Existe algum problema na utilização de software COTS?

- Documentação insuficiente para determinar as interfaces, tamanho ou performance
- Performance ruim
- Necessita de uma quantidade grande de memória ou de banco de dados
- É difícil de fazer uma interface com o software da aplicação
- Não foi completamente testado

- Não é isento de *bugs*
- Não é mantido adequadamente
- Tempo de resposta do fornecedor ruim

[30] Você prevê algum problema com a integração de revisões e *updates* do *software* COTS?

## 3.Codificação e teste unitário

**a) Viabilidade**

*[A implementação do design é difícil ou impossível?]*

[31] Alguma parte da implementação do produto não está completamente definida pela especificação do *design*?

[32] Os *design* e algoritmos selecionados são fáceis de implementar?

**b) Testes**

*[O tempo e nível de teste especificados são adequados para os testes unitários?]*

[33] Você inicia os testes unitários antes de verificar o código com respeito ao *design*?

[34] Foram especificados testes unitários suficientes?

[35] Existe tempo suficiente para ser executado todo o teste unitário que você acredita que deva ser feito?

[36] O tempo de teste será comprometido em caso de problemas com o cronograma?

**c) Codificação/implementação**

*[Existe algum problema com a codificação e implementação?]*

[37] Existem detalhes suficientes na especificação do *design* para escrever o código?

[38] O *design* está mudando enquanto o código está sendo escrito?

[39] Existem restrições de sistema que tornem o código difícil de escrever?

- Tempo
- Memória
- Armazenamento externo

[40] A linguagem é adequada para produzir o *software* deste programa?

[41] Existem múltiplas linguagens sendo utilizadas no programa?

(sim) (41.a) Existe compatibilidade nas interfaces entre os códigos produzidos pelos diferentes compiladores?

[42] O computador de desenvolvimento é o mesmo do computador de produção?

(não) (42.a) Existem diferenças de compilador entre os dois?

- **Se *developmental hardware* estiver sendo usado**

[43] As especificações do *hardware* são adequadas para a codificação do *software*?

[44] As especificações do *hardware* estão mudando enquanto o código está sendo desenvolvido?

## 4.Teste e integração

**a) Ambiente**

*[Os ambientes de teste e integração são adequados?]*

[45] Estará disponível *hardware* suficiente para conduzir os testes e a integração adequadamente?

[46] Existe algum problema no desenvolvimento de cenários realísticos e dados de teste para demonstrar os requisitos?

- Tráfico de dados especificados
- Resposta *real-time*
- Gerenciamento de eventos assíncronos
- Interação de múltiplos usuários

[47] Você está apto a verificar a performance na suas instalações?

[48] A instrumentação do *hardware* e *software* facilita os testes?

(sim) (48.a) Ela é suficiente para todo o teste?

**b) Produto**

*[A definição das interfaces é inadequada, as instalações são inadequadas, o tempo é insuficiente?]*

[49] O *hardware* de produção estará disponível quando necessário?

[50] Foram acordados critérios de aceite para todos os requisitos?

(sim) (50.a) Existe um acordo formal?

[51] As interfaces externas foram definidas, documentadas e referenciadas?

[52] Existe algum requisito que será difícil de testar?

[53] Foi especificado o suficiente em termos de integração do produto?

[54] Foi alocado tempo suficiente para teste e integração do produto?

- **Caso COTS esteja envolvido**

[55] Dados do fornecedor serão aceitos na verificação dos requisitos alocados a produtos COTS?

(sim) (55.a) O contrato está claro quanto a este ponto?

**c) Sistema**

*[Falta de coordenação na integração do sistema, má definição das interfaces, instalações inadequadas?]*

[56] Foi especificada integração suficiente do sistema?

[57] Foi alocado tempo adequado para integração do sistema e testes?

[58] Todos os contratados fazem parte do time de integração?

[59] O produto será integrado a um sistema existente?

(sim) (59.a) Existe um tempo de paralelo com o sistema atual?

(não) (59.a.1) Como você garantirá que o produto irá funcionar corretamente quando integrado?

[60] A integração do sistema será feita nas instalações do cliente?

## 5.Especialidades de engenharia

**a) *Maintainability***

*[A implementação será difícil de entender ou manter?]*

[61] A arquitetura, *design* ou codificação criam alguma dificuldade na manutenção?

[62] O time de manutenção foi envolvido no início do *design*?

[63] A documentação do produto é adequada para a manutenção por uma organização externa?

**b) Confiabilidade**

*[Os requisitos de confiabilidade e disponibilidade são difíceis de atingir?]*

[64] Existem requisitos de confiabilidade alocados ao *software*?

[65] Existem requisitos de disponibilidade alocados ao *software*?

(sim) (65.a) Os tempos de recuperação são problema?

**c) Segurança**

*[Os requisitos de segurança são inviáveis ou impossíveis de demonstrar?]*

[66] Existem requisitos de segurança alocados ao *software*?

(sim) (66.a) Você vê alguma dificuldade em implementar estes requisitos?

[67} Será difícil de verificar a satisfação dos requisitos de segurança?

**d) Proteção**

*[Os requisitos de proteção são mais restritivos do que a prática atual ou do que a experiência do programa?]*

[68] Existem requisitos de proteção *state-of-the-art* ou sem precedência?

[69] É um sistema *orange book* ?

[70] Você já implementou este nível de proteção antes?

**e) Fatores humanos**

*[O sistema será difícil de usar devido a má definição da interface humana?]*

[71] Você prevê alguma dificuldade na implementação dos requisitos de Fatores Humanos?

(não) (71.a) Como você está se assegurando de atingir estes requisitos?

- **Caso haja prototipagem**

(sim) (71.a.1) É um protótipo que será jogado fora?

(não) (71.a.1a) Você está fazendo um desenvolvimento evolucionário?

(sim) (71.a.1.1a.1) Você tem experiência neste tipo de desenvolvimento?

(sim) (71.a.1.1a.2) São entregues versões parciais ?

(sim) (71.a.1.1a.3) Isto complica o controle de mudanças?

**f) Especificações**

*[A documentação é adequada para desenhar, implementar e testar o sistema?]*

[72] As especificações dos requisitos de *software* são adequadas para desenhar o sistema?

[73] As especificações do *hardware* são adequadas para desenhar e implementar o *software*?

[74] Os requisitos de interface externa estão bem especificados?

[75] As especificações dos testes estão adequadas para testar totalmente o sistema?

- **Caso esteja ou tenha passado pela fase de implementação**

[76] As especificações do desenho são suficientes para implementar o sistema?

- Interfaces internas

# B. Ambiente de desenvolvimento

## 1. Processo de desenvolvimento

### a) Formalidade

*[A implementação será difícil de entender ou manter?]*

[77] Está sendo utilizado mais de um modelo de desenvolvimento?

- Espiral
- Cascata
- Incremental

(sim) (77.a) A coordenação entre eles é um problema?

[78] Existe um plano formal e controlado para todas as atividades do desenvolvimento?

- Análise dos requisitos
- *Design*
- Codificação
- Integração e testes
- Garantia da qualidade
- Gerenciamento de configurações

(sim) (78.a) Os planos especificam bem o processo?

(sim) (78.b) Os desenvolvedores estão familiarizados com o plano?

### b) Adequação

*[O processo está adequado ao modelo de desenvolvimento, isto é, espiral, prototipagem?]*

[79] O processo de desenvolvimento é adequado para este produto?

[80] O processo de desenvolvimento é suportado por um conjunto compatível de procedimentos, métodos e ferramentas?

### c) Controle do Processo

*[O processo de desenvolvimento do software é reforçado, monitorado e controlado usando métricas? Os ambientes de desenvolvimento distribuídos são coordenados?]*

[81] Todos seguem o processo de desenvolvimento?

(sim) (81.a) Como isto é assegurado?

[82] Você pode medir se o processo de desenvolvimento está atingindo suas metas de produtividade e qualidade?

- **Caso existam ambientes de desenvolvimento distribuídos**

[83] Existe uma coordenação adequada entre os ambientes de desenvolvimento distribuídos?

**d) Familiaridade**

*[Os membros do projeto tem experiência no uso do processo? O processo é entendido por todos os membros do staff?]*

[84] As pessoas estão confortáveis com o processo de desenvolvimento?

**e) Controle do produto**

*[Existem mecanismos para controlar mudanças no produto?]*

[85] Existe um mecanismo de acompanhamento dos requisitos que monitora os requisitos desde a especificação inicial até os *test cases*?

[86] O mecanismo de acompanhamento é utilizado na análise de impacto de solicitações de mudança?

[87] Existe um processo formal de controle de mudanças?

(sim) (87.a) Ele cobre todas as mudanças de requisitos, *design*, codificação e documentação?

[88] Todas as mudanças em qualquer nível são mapeadas para cima até o nível do sistema e para baixo até o nível de testes?

[89] É feita uma análise adequada quando novos requisitos são adicionados ao sistema?

[90] Você tem um modo de monitorar as interfaces?

[91] Os planos e procedimentos de teste são atualizados como parte do processo de mudanças?

## 2.Sistema de desenvolvimento

**a) Capacidade**

*[Existe capacidade suficiente de processamento, memória e armazenamento nas estações de trabalho?]*

[92] Existem estações de trabalho e capacidade de processamento suficiente para todo o *staff*?

[93] Existe capacidade suficiente para fases em paralelo, como codificação, integração e teste?

**b) Adequação**

*[O sistema de desenvolvimento suporta todas as fases, atividades e funções?]*

[94] O sistema de desenvolvimento suporta todos os aspectos do programa?

- Análise de requisitos
- Análise de performance
- *Design*
- Codificação
- Testes
- Documentação
- Gerenciamento de configurações
- Acompanhamento gerencial
- Monitoramento dos requisitos

**c) Usabilidade**

*[O sistema de desenvolvimento é fácil de usar?]*

[95] As pessoas acham o sistema de desenvolvimento fácil de usar?

[96] Existe uma boa documentação sobre o sistema de desenvolvimento?

**d) Familiaridade**

*[Os membros do projeto ou a companhia tem alguma experiência anterior com o sistema de desenvolvimento?]*

[97] As pessoas já usaram estas ferramentas e métodos antes?

**e) Confiabilidade**

*[O sistema sofre de bugs, down-time, falta de backup?]*

[98] O sistema é considerado confiável?

- Compilador
- Ferramentas de desenvolvimento
- *Hardware*

**f) Suporte ao sistema**

*[O sistema tem suporte em tempo?]*

[99] As pessoas estão treinadas no uso das ferramentas de desenvolvimento?

[100] Você tem acesso a pessoas *experts* no uso do sistema?

[101] Os fornecedores respondem a problemas rapidamente?

**g) Capacidade de entrega**

*[Os requisitos de aceite e definição para a entrega do sistema de desenvolvimento para o cliente foram orçados? DICA: Se os participantes ficarem confusos sobre isto, provavelmente ele não é um problema de um ponto de vista de riscos.]*

[102] Você está entregando o sistema de desenvolvimento para o cliente?

(sim) (102.a) Foram alocados custos, tempo e recursos adequados para esta entrega?

## 3.Processo de gerenciamento

**a) Planejamento**

*[O planejamento é conveniente, incluindo o pessoal técnico, o plano de contingência foi feito?]*

[103] O programa está sendo gerenciado de acordo com o plano?

(sim) (103.a) As pessoas são rotineiramente sacadas para apagar incêndios?

[104] É feito um re-planejamento quando ocorrem interferências?

[105] As pessoas em todos os níveis são incluídas no planejamento de suas próprias tarefas?

[106] Existem planos de contingência para riscos conhecidos?

(sim) (106.a) Como você determina quando ativar as contingências?

[107] Os problemas de longo-prazo estão sendo endereçados adequadamente?

**b) Organização do projeto**

*[Os papéis e relacionamentos de reporte estão claros?]*

[108] A organização do projeto é efetiva?

[109] As pessoas entendem o seu próprio papel e os dos outros no programa?

[110] As pessoas sabem quem tem autoridade sobre o que?

**c) Experiência da gerência**

*[Os gerentes são experientes no desenvolvimento de software, no gerenciamento de software, no domínio da aplicação, no processo de desenvolvimento ou em grandes programas?]*

[111] O programa conta com gerentes experientes?

- Gerenciamento de *software*

- Desenvolvimento de *software*
- Com este processo de desenvolvimento
- Com o domínio da aplicação
- Tamanho e complexidade do programa

**d) Interfaces do programa**

*[A interface com o cliente, outros contratados, gerentes pares ou sênior é pobre?]*

[112] A gerência comunica os problemas para cima e para baixo na estrutura?

[113] Os conflitos com clientes são documentados e resolvidos em tempo?

[114] A gerência envolve os membros apropriados do programa nas reuniões com o cliente?

- Líderes técnicos
- Desenvolvedores
- Analistas

[115] A gerência trabalha para assegurar que todos os grupos de clientes são representados em decisões com respeito a funcionalidades e operação?

[116] É considerada uma boa política apresentar um panorama otimista para o cliente ou gerência sênior?

## 4.Métodos de gerenciamento

**a) Monitoração**

*[Foram definidas métricas gerenciais e o progresso do desenvolvimento é acompanhado?]*

[117] Existem relatórios de status estruturados e periódicos?

(sim) (117.a) As pessoas recebem uma resposta para seus relatórios de status?

[118] As informações apropriadas são reportadas para os níveis organizacionais corretos?

[119] Você acompanha o progresso contra o plano?

(sim) (119.a) A gerência tem um panorama claro do que está acontecendo?

**b) Gerenciamento do pessoal**

*[O pessoal do projeto é treinado e utilizado adequadamente?]*

[120] As pessoas estão sendo treinadas nas capacidades requeridas por este programa?

(sim) (120.a) Isto faz parte do plano do programa?

[121] Pessoas foram designadas para o programa sem ter o perfil para o trabalho?

[122] É fácil para os membros do programa obter ações da gerência?

[123] Os membros do programa em todos os níveis estão cientes de seu status versus o plano?

[124] As pessoas entendem que é importante se ater ao plano?

[125] A gerência consulta as pessoas antes de tomar decisões que afetem o seu trabalho?

[126] A gerência do programa envolve os membros apropriados do programa para reuniões com o cliente?

- Líderes técnicos
- Desenvolvedores
- Analistas

**c) Garantia da qualidade**

*[Existem procedimentos e recursos adequados para garantir a qualidade do produto?]*

[127] A função de garantia da qualidade do *software* tem pessoal adequado alocado para este programa?

[128] Você definiu mecanismos adequados para garantir a qualidade?

(sim) (128.a) Todas as áreas e fases tem procedimentos de qualidade?

(sim) (128.b) As pessoas estão acostumadas a trabalhar com estes procedimentos?

**d) Gerenciamento de configurações**

*[Os procedimentos de mudanças ou controle de versões, incluindo a instalação dos ambientes, são adequados?]*

[129] Você tem um sistema adequado de gerenciamento de configurações?

[130] Foi alocado pessoal adequado para a função de gerenciamento de configurações?

[131] É necessária a coordenação com um sistema já instalado?

(sim) (131.a) O sistema instalado tem um gerenciamento de configurações adequado?

(sim) (131.b) O sistema de gerenciamento de configurações sincroniza o seu trabalho com as alterações no ambiente?

[132] Você está fazendo instalações em múltiplos ambientes?

(sim) (132.a) O sistema de gerenciamento de configurações suporta múltiplos ambientes?

## 5.Ambiente de trabalho

**a) Atitude com a qualidade**

*[Existe uma falta de orientação para com a qualidade no trabalho?]*

[133] Todos os níveis do *staff* estão orientados quanto aos procedimentos de qualidade?

[134] O prazo interfere no caminho da qualidade?

**b) Cooperação**

*[Existe falta de espírito de time? A resolução de conflitos requer a intervenção da gerência?]*

[135] As pessoas trabalham de forma cooperativa através de fronteiras funcionais?

[136] As pessoas trabalham efetivamente para metas comuns?

[137] A intervenção da gerência é requerida algumas vezes para que as pessoas trabalhem juntas?

**c) Comunicação**

*[Existe uma compreensão fraca das metas e objetivos ou uma comunicação pobre das informações técnicas entre os pares e gerentes?]*

[138] Existe uma boa comunicação entre os membros do programa?

- Gerentes
- Líderes técnicos
- Desenvolvedores
- Testadores
- Gerência de configuração
- Garantia da qualidade

[139] Os gerentes são receptivos à comunicação do staff do programa?

(sim) (139.a) Você sente liberdade em pedir ajuda ao seu gerente?

(sim) 139.b) Os membros do programa estão aptos a apontar riscos sem ter uma solução nas mãos?

[140] Os membros do programa são informados em tempo sobre eventos que podem afetar o seu trabalho?

(sim) (140.a) Isto é feito de forma formal ou informal?

**d) Moral**

*[Existe um ambiente não-produtivo ou não-criativo? As pessoas sentem que não há reconhecimento ou recompensa para um trabalho superior?]*

[141] Como está a moral do programa?

(no) (141.a) Qual é o principal fator para a abaixa moral?

[142] Existe algum problema para manter o pessoal que você precisa?

# C. Restrições do programa

## 1.Recursos

**a) Cronograma**

*[O cronograma é inadequado ou instável?]*

[143] O cronograma está sendo estável?

[144] O cronograma é realista?

(sim) (144.a) O método de estimativa é baseado em dados históricos?

(sim) (144.b) O método funcionou bem no passado?

[145] Existe algo para o qual prazos adequados não foram planejados?

- Estudos e análises
- Garantia da qualidade
- Treinamento
- Treinamento e cursos de manutenção
- Equipamento capitalizado
- O sistema que está sendo desenvolvido

[146] Existem dependências externas que podem impactar o cronograma?

**b) *Staff***

*[Faltam experiência, conhecimento do domínio, capacidade ao staff ou ele foi sub-dimensionado?]*

[147] Existe alguma área onde está faltando capacidade técnica?

- Engenharia de *software* e métodos de análise de requisitos
- Expertise em algoritmos
- Métodos de *design*
- Linguagens de programação
- Métodos de integração e testes

- Confiabilidade
- *Maintainability*
- Disponibilidade
- Fatores humanos
- Gerenciamento de configuração
- Garantia da qualidade
- Ambiente de produção
- Nível de segurança
- COTS
- Software reutilizado
- Sistema operacional
- Banco de dados
- Domínio da aplicação
- Análise de performance
- Aplicações críticas ao tempo

[148] Você tem pessoal adequado para alocar ao programa?

[149] O *staff* é estável?

[150] Você tem acesso às pessoas certas quando você precisa delas?

[151] Os membros do programa já implementaram sistemas deste tipo?

[152] O programa é baseado em algumas poucas pessoas chave?

[153] Existe algum problema em ter as pessoas disponíveis?

**c) Orçamento**

*[A verba é insuficiente ou instável?]*

[154] O orçamento é estável?

[155] O orçamento foi baseado em uma estimativa realista?

(sim) (155.a) O método de estimativa é baseado em dados históricos?

(sim) (155.b) O método funcionou bem no passado?

[156] Funções ou recursos foram excluídos como parte de um esforço de desenhar para o custo?

[157] Existe algo para o qual não foi alocada verba adequada?

- Estudos e análises
- Garantia da qualidade

- Treinamento
- Cursos de manutenção
- Equipamento capitalizado
- O sistema que está sendo desenvolvido

[158] As mudanças de requisitos são acompanhadas de mudanças no orçamento?

(sim) (158.a) Isto é uma parte padrão do processo de controle de mudanças?

**d) Instalações**

*[As instalações são adequadas para construir e entregar o produto?]*

[159] O ambiente de desenvolvimento é adequado?

[160] O ambiente de integração é adequado?

## 2.Contrato

**a) Tipo do contrato**

*[O tipo de contrato é uma fonte de risco para o programa?]*

[161] Que tipo de contrato foi utilizado? (Preço fixo, custo mais taxa, ...)

(161.a) Ele apresenta algum tipo de problema?

[162] O contrato é sobrecarregado em algum aspecto do programa?

- *Statement of Work* (SOW)
- Especificações
- Descrições dos itens de dados
- Partes contratuais
- Envolvimento excessivo do cliente

[163] A documentação requerida é exagerada?

- Quantidade excessiva
- Cliente extremamente seletivo
- Longo ciclo de aprovação

**b) Restrições**

*[O contrato causa alguma restrição?]*

[164] Existem problemas com direitos dos dados?

- COTS *Software*
- O *software* que está sendo desenvolvido

- Itens não desenvolvidos pelo projeto

**c) Dependências**

*[O programa tem alguma dependência de produtos ou serviços externos?]*

[165] Existem dependências em produtos ou serviços externos que podem afetar o produto, orçamento ou cronograma?

- *Associate contractors*
- *Prime contractors*
- Sub-contratados
- Vendedores ou fornecedores
- Equipamentos ou *software* fornecidos pelo cliente

## 3.Interfaces do programa

**a) Cliente**

*[Existe algum tipo de problema com o cliente, como por exemplo: longo ciclo de aprovação da documentação, comunicação pobre, e falta de expertise com o domínio?]*

[166] O ciclo de aprovação do cliente é em tempo?

- Documentação
- Revisões do programa
- Revisões formais

[167] Você em alguma situação dá prosseguimento sem a aprovação do cliente?

[168] O cliente entende os aspectos técnicos do sistema?

[169] O cliente entende de *software*?

[170] O cliente interfere com o processo ou com o pessoal?

[171] A gerência trabalha com o cliente para obter decisões mutuamente acordadas e no tempo apropriado?

- Entendimento dos requisitos
- Critérios de teste
- Ajustes no cronograma
- Interfaces

[172] Quão efetivos são seus mecanismos para obter acordos com o cliente?

- Grupos de trabalho (contratuais?)
- Reuniões técnicas (contratuais?)

[173] Todos os grupos de clientes estão envolvidos na obtenção de acordos?

(sim) (173.a) Este é um processo formalmente definido?

[174] A gerência apresenta um panorama realista ou otimista para o cliente?

- **Caso existam contratantes associados**

**b) *Associate Contractors* (contratantes associados)**

*[Existe algum problema com contratantes associados, como por exemplo, interfaces mal definidas ou instáveis, falha de comunicação ou falta de cooperação?]*

[175] Existem interfaces externas sendo alteradas sem a adequada notificação, coordenação ou procedimentos formais de mudança?

[176] Existe um plano de transição adequado?

(sim) (176.a) Ele é suportado por todos os contratados e o pessoal do *site*?

[177] Existe algum problema para obter cronogramas ou dados de interface dos contratantes associados?

(não) (177.a) Eles são precisos?

- **Caso existam sub-contratados**

**c) Sub-contratados**

*[O programa depende de sub-contratados em alguma área crítica?]*

[178] Existe alguma ambigüidade na definição de tarefas dos sub-contratados?

[179] Os procedimentos de monitoramento e reporte do sub-contratado são diferentes dos requisitos de reporte do programa?

[180] O gerenciamento técnico e administrativo do sub-contratado é feito por uma organização separada?

[181] Você é altamente dependente do expertise do sub-contratado em alguma área?

[182] O conhecimento do sub-contratado está sendo transferido para a companhia?

[183] Existe algum problema para obter cronogramas ou dados de interface dos sub-contratados?

- **Caso o programa seja um sub-contrato**

**d) *Prime contractor***

*[O programa está enfrentando dificuldades com o Prime contractor?]*

[184] As definições das suas tarefas feitas pelo *Prime contractor* são ambíguas?

[185] Você lida com duas organizações separadas do *Prime* para gerenciamento técnico e administrativo?

[186] Você é altamente dependente do expertise do *Prime* em alguma área?

[187] Existe algum problema para obter cronogramas ou dados de interface do *Prime*?

**e) Gerência corporativa**

*[Existe falta de suporte ou micro gerenciamento de parte da gerência superior?]*

[188] A gerência do programa comunica os problemas para a gerência sênior?

(sim) (188.a) Isto parece ser efetivo?

[189] A gerência corporativa dá suporte no tempo apropriado para resolver os seus problemas?

[190] A gerência corporativa tende a gerenciar de forma micro?

[191] A gerência apresenta um panorama otimista ou realista para a gerência sênior?

**f) Fornecedores**

*[Os fornecedores respondem às necessidades do programa?]*

[192] Você depende de fornecedores para a entrega de componentes críticos?

- Compiladores
- *Hardware*
- COTS

**g) Política**

*[Itens políticos estão causando algum problema para o programa?]*

[193] Itens políticos estão afetando o programa?

- Companhia
- Clientes
- *Associate contractors*
- Sub-contratados

[194] Itens políticos estão afetando decisões técnicas?

# Apêndice D – TBQ - Questionário resumido

Este Apêndice é uma tradução feita pelo autor do questionário resumido do TBQ apresentado por Pandelios (1999). Novamente, alguns termos foram mantidos pelo autor em inglês devido à sua larga utilização no domínio da tecnologia da informação e da dificuldade de tradução para um termo semelhante em português.

## A. Produto (Engenharia do produto)

Pense sobre os riscos ao projeto que podem surgir da natureza do produto que você está tentando desenvolver.

**A.1 Requisitos** – existem riscos que podem surgir dos requisitos que estão sendo solicitados para o produto ? Exemplos: estabilidade, totalidade, clareza, validade, viabilidade, precedência, escala.

**A.2 *Design*** – existem riscos que podem surgir do *design* que foi escolhido pelo projeto para atender os requisitos? Exemplos: funcionalidade, dificuldade, interfaces, performance, testabilidade, restrições de hardware, *non-developmental software*.

**A.3 Codificação e testes unitários (manufaturabilidade)** – existem riscos que podem surgir da forma com que o projeto está decidindo subdividir o *design* e construir as partes? Exemplos: viabilidade, testes, codificação/implementação.

**A.4 Integração e testes** – existem riscos que podem surgir da forma com que o projeto está decidindo juntar as partes e provar que eles funcionam como um todo? Exemplos: as instalações de *hardware* e *software*, integração das partes do produto, integração com o sistema mais amplo.

**A.5 Especialidades da engenharia** – existem riscos que podem surgir de atributos especiais do produto, como *Maintainability*, confiabilidade, segurança, proteção, fatores humanos, etc ?

**A.99 Outros** – existem outros riscos que podem surgir do produto em si, mas que não foram cobertos em nenhuma das categorias acima?

## B. Processo (Ambiente de desenvolvimento)

Pense nos riscos ao projeto que podem surgir da <u>forma</u> com que você irá desenvolver o produto.

**B.1 Processo de desenvolvimento** – existem riscos que podem surgir da forma com que o processo do projeto decidiu desenvolver o produto? Exemplos: formalidade, adequação, controle do processo, familiaridade, controle do produto.

**B.2 Sistema de desenvolvimento** – existem riscos que podem surgir das ferramentas de hardware e software que o projeto decidiu utilizar para controlar e facilitar o processo de desenvolvimento? Exemplos: capacidade, adequação, usabilidade, familiaridade, confiabilidade, suporte ao sistema e capacidade de entrega.

**B.3 Sistema de gerenciamento** – existem riscos que podem surgir da forma com que o orçamento e cronograma é planejado, monitorado ou controlado ? Ou da estrutura do projeto? Ou da maneira com que ela lida com interfaces organizacionais internas e externas?

**B.4 Métodos de gerenciamento** – existem riscos que podem surgir da forma com que o pessoal do programa ou de desenvolvimento é gerenciado, em áreas como acompanhamento de status, gerenciamento do pessoal, garantia da qualidade ou gerenciamento de configurações?

**B.5 Ambiente de trabalho** – existem riscos que podem surgir do ambiente em geral ou da organização como um todo a que o projeto pertence, com por exemplo em relação a atitude com a qualidade, cooperação, comunicação ou moral?

**B.99 Outros** – existem outros riscos que podem surgir da forma com que o projeto é desenvolvido, mas que não estão cobertos pelas categorias acima?

## C. Restrições (Restrições do programa)

Pense sobre riscos ao projeto que podem surgir de fontes externas ao controle do projeto.

**C.1 Recursos** – existem riscos que podem surgir de recursos que o projeto necessita mas que estão fora de seu controle para serem obtidos ou mantidos? Exemplos: prazos, *staff*, verbas, instalações.

**C.2 Contrato** – existem riscos que podem surgir do contrato? Exemplos incluem áreas como tipo, restrições ou dependências.

**C.3 Interfaces do programa** –existem riscos que podem surgir de interfaces externas que o projeto não espera poder controlar? Exemplos: clientes, *associate contractors*, sub-contratados, *prime contractor*, gerenciamento corporativo, fornecedores, políticas.

**C.99 Outros** – existem outros riscos que podem surgir de fatores externos ao controle do projeto, mas que não estão cobertos pelas categorias acima?

# 7 Referências bibliográficas


1. BARTIÈ, A. Garantia da Qualidade de Software, Rio de Janeiro, Ed. Campus, 2002. 291 p. ISBN 85-352-1124-1

2. BOEHM, B.W. A Spiral model of software development and enhancement. IEEE Computer, p. 61-72, 1988.

3. BOEHM, B.W. Software risk management. USA, IEEE Computer Society Press, 1989. 496p. ISBN 0-8186-8906-4

4. BOEHM, B.W. Software Risk Management: Principles and Practices. IEEE Software, v.8, p. 31-41, 1991.

5. BOEHM, B.W.; DE MARCO, T. Software risk management (guest editor's introduction). IEEE Software, p.17-19, 1997.

6. BRYMAN, A., Research methods and organization studies. Londres, Unwin Hyman Ltd, 1989. p-179-187.

7. BUZIN, P. K., Impressões do 4º Congresso Ibero-americano de gerência de projetos. PMI-RS Journal, v.7, p. 7-8, 2003.

8. CARR, M.J.; KONDA, S.L. et al. Taxonomy-based risk identification. Software Engineering Institute, Carnegie Mellon University, 1993. (Technical report CMU/SEI-93-TR-06)

9. CMMI - Continuous Representation. CMI/SE/SW/PPD/SS, v.1.1, p-288-309.

10. FERREIRA, A. B. H., Dicionário Aurélio básico da língua portuguesa. Brasil, Ed. Nova Fronteira, 1988. 687p.

11. FILIATRAULT, C. L. G.; PETERSON, C. D. , Five behaviors that can reduce schedule risk. Proceedings of the Project Management Institute Annual Seminars & Symposium, USA, 2000.

12. GLUCH, D. P., A Construct for describing software development risks. Software Engineering Institute, Carnegie Mellon University, 1994. (Technical Report CMU/SEI-94-TR-14)

13. GOLDRATT, E. M., Corrente Crítica.São Paulo, Nobel, 1998. 260p. ISBN 85-213-1007-2

14. HALL, E. M. Managing Risk - Methods for software system development. USA, Adison-Wesley, 1998. 374p. ISBN 0-201-25592-8

15. HIGUERA, R.P. et al, An introduction to team risk management (version 1.0). Software Engineering Institute, Carnegie Mellon University, 1994. ( Special report CMU/SEI-94-SR-01)

16. HIGUERA, R.P.; DOROFEE, A. J. et al. Team risk management: a new model for customer-supplier relationship. Software Engineering Institute, Carnegie Mellon University, 1994. (Technical report CMU/SEI-94-SR-05)

17. HIGUERA, R.P.; HAIMES, Y.Y. Software risk management. Software Engineering Institute, Carnegie Mellon University, 1996. (Technical report CMU/SEI-96-TR-12)

18. HUMPREY, W. S. Introduction to the Team Software Process. Addison-Wesley, Reading, MA, 1999.

19. ICE - INTEGRATED COMPUTER ENGINEERING, INC. 16 Critical software practices for performance based management, version 5.1. 1999. Disponível na internet via http://www.spmn.com/16CSP.html.

20. JIAN, J.J. et al. Reducing user related risks during and prior to system development. IJPM, v. 20, p. 507-515, 2002.

21. LIENTZ, B. P.; REA, K. P. Breakthrough Technology Project Management, 2 ed. USA, Academic Press, 2001. 342p. ISBN 0-12-449968-6

22. LINK, J. L. L.; BARBOUR, R. et al. Rollout and Installation of Risk Management at the IMINT Directorate, National Reconnaissance Office. Software Engineering Institute, Carnegie Mellon University, 1999. (Technical report CMU/SEI-99-TR-09)

23. LUNARDI, G. L.; HENRIQUE, J. L. Aplicação da Soft Systems Methodology na avaliação de um programa de pós-graduação em Administração: perspectiva do corpo discente. READ - Revista Eletrônica da Administração (UFRGS), Porto Alegre, v. 8, n. 4, 2002.

24. MACHADO, Cristina A F. A-Risk: um método para identificar e quantificar risco de prazo em projetos de desenvolvimento de software. Curitiba, 2002. 239 pg. Dissertação (Mestrado) – Centro de ciências exatas e tecnologia, Pontifícia Universidade Católica do Paraná.

25. PANDELIOS, G.J. et al. Software risk evaluation (SRE) - Team member's notebook (version 2.0). Software Engineering Institute, Carnegie Mellon University, 1999. (Technical report CMU/SEI-99-TR-29)

26. PINEY, C., Risk Response Planning: Selecting the Right Strategy. Fifth European Project Management Conference, PMI Europe, 2002.

27. PMI – São Paulo, Brazil Chapter. PMI: O Instituto. [OnLine] Disponível na internet via http://www.pmisp.org.br/exe/pmi/instituto.asp. Acessado em 17/out/2004.

28. PROJECT MANAGEMENT INSTITUTE – PMI. A Guide to the Project Management Body of Knowledge (PMBoK Guide), 2000 Edition. USA, 2000. 216p. ISBN 1-880410-23-0

29. RAZ, T.; MICHAEL, E. Use and benefits of tools for project risk management. IJPM, v.19, p.9-17, 2001.

30. SCHMIDT, C. et al. Disincentives for communicating risk: a risk paradox. Information & Software Technology, v.41, p. 403-411, 1999.

31. SILVA, E. L.; MENEZES, E. M., Metodologia da pesquisa e elaboração de dissertação, 3ª ed. Florianópolis, Universidade Federal de Santa Catarina, 2001. 121p.

32. SISK, T., History of Project Management. 1998. Disponível na internet via http://www.microsoft.com/downloads/details.aspx?FamilyID=c1f9b881-d879-4b54-b07b-55041685f15f&displaylang=en.

33. THE STANDISH GROUP INTERNATIONAL, INC. CHAOS: A recipe for success. 1999. Disponível na internet via http://www.standishgroup.com/sample_research/index.php.

34. THE STANDISH GROUP INTERNATIONAL, INC. Extreme CHAOS. 2001. Disponível na internet via http://www.standishgroup.com/sample_research/index.php.

35. THE STANDISH GROUP INTERNATIONAL, INC. Press Release: Latest Standish Group CHAOS Report shows project success rates have improved by 50%. 2003. Disponível na internet via http://www.standishgroup.com/press/.

36. THE STANDISH GROUP INTERNATIONAL, INC. The CHAOS Report. 1995. Disponível na internet via http://www.standishgroup.com/sample_research/index.php.

37. THEMISTOCLEOUS, G.; WEARNE S.H. Project management topic coverage in journals. International Journal of Project Management,v.18, p. 7-11, 2000.

38. VAN SCOY, R. L. Software development risk: opportunity, not problem. Software Engineering Institute, Carnegie Mellon University, 1992. (Technical report CMU/SEI-92-TR-30)

39. WILLIAMS, R.C.; PANDELIOS, G.J.; BEHRENS, S.G. Software risk evaluation (SRE) - Method description (version 2.0). Software Engineering Institute, Carnegie Mellon University, 1999. (Technical report CMU/SEI-99-TR-29)

40. YIN, R. K. Case study research: design and methods. New Delhi, Sage, 1984.