



## A IMPORTÂNCIA DA ADOÇÃO DA ABORDAGEM DE RISCOS NO ENSINO DA ENGENHARIA DE SOFTWARE

Marcelo Nogueira – marcelo@noginfo.com.br  
UNIP – Universidade Paulista, Grupo de Pesquisa em Engenharia de Software  
R. Antonio de Macedo, 505, Tatuapé  
03087-040 – São Paulo – SP - Brasil  
Ricardo J. Machado – rmac@dsi.uminho.pt  
Universidade do Minho, Escola de Engenharia - Centro Algoritmi  
Campus Azurém  
4800-058 – Guimarães - Portugal

*Resumo: Num mercado global cada vez mais competitivo, os engenheiros de software, para atender as demandas empresariais as quais são submetidos, realizam atividades e procedimentos com alta exposição a riscos, nem sempre conhecidos e mensurados. Tendo em vista que, uma minoria dessas empresas adota a gestão de riscos em seus processos, essa exposição pode comprometer a participação e o sucesso dos projetos que estão envolvidos. Para garantir a qualidade do processo, uma abordagem focada em riscos deve ser adotada sistematicamente. Entre as incertezas dos projetos de software, alguns fatores de riscos devem ser tratados. Os prazos e custos estimados, a conformidade com os requisitos de negócios, entre outros, podem ser citados. Através de uma pesquisa bibliográfica foi possível compor um roteiro de atividades de riscos para propiciar um tratamento mais adequado. Para contribuir com o ensino dessas temáticas que envolvem projetos de software, neste trabalho é apresentado um processo de gestão de riscos com a finalidade de inserir a cultura e a capacidade nos profissionais que neles atuam, podendo direcionar objetivamente como mitigar os riscos aos quais são expostos. Completa a abordagem, o fato de este roteiro estar em conformidade com a ISO 31000:2009.*

*Palavras-chave: Engenharia de Software, Gestão de Riscos, Qualidade de processos.*

### 1. INTRODUÇÃO

Num ambiente competitivo e de mudança cada vez mais complexo, a gestão adequada da Informação assume uma importância decisiva no processo de tomada de decisão nas organizações (NOGUEIRA, 2009).

Tratando-se de um tema simultaneamente abrangente e especializado, a adoção das práticas da Engenharia de Software como linha base da Gestão da Informação, possibilita, não só desenvolver e consolidar os conhecimentos na produção de software, bem como preparar os futuros profissionais para encarar com confiança os novos desafios no mundo dos negócios, reforçando as competências e habilidades, mantendo-se atualizado em relação ao

Realização:



Organização:





potencial dos sistemas de informação e das novas tecnologias numa perspectiva empresarial competitiva globalmente.

## 2. ENGENHARIA DE SOFTWARE

Engenharia de Software é: O estabelecimento e uso de sólidos princípios de engenharia para que se possa obter economicamente um software que seja confiável e que funcione eficientemente em máquinas reais. Descendente da engenharia de sistemas e de hardware. Abrange um conjunto de 3 elementos fundamentais (métodos, ferramentas e procedimentos), que possibilita, ao gerente, o controle do processo de desenvolvimento do software e oferece ao profissional uma base para a construção de software de alta qualidade (PRESSMAN, 2006).

A engenharia de software pode ser definida pelo conjunto de métodos, procedimentos e ferramentas com o objetivo de construir software com qualidade, ou seja, em conformidade com as necessidades dos clientes (NOGUEIRA, 2009).

## 3. OBJETIVOS DA ENGENHARIA DE SOFTWARE

A Engenharia de Software tem como objetivo primário o aprimoramento da qualidade dos produtos de software e o aumento da produtividade dos engenheiros de software, além do atendimento aos requisitos de eficácia e eficiência, ou seja, efetividade (MAFFEO, 1992).

A Engenharia de Software visa sistematizar a produção, a manutenção, a evolução e a recuperação de produtos de software, de modo que ocorra dentro de prazos e custos estimados, com progresso controlado e utilizando princípios, métodos, tecnologia e processos em contínuo aprimoramento. Os produtos desenvolvidos e mantidos, seguindo um processo efetivo e segundo preceitos da Engenharia de Software asseguram, por construção, qualidade satisfatória, apoiando adequadamente os seus usuários na realização de suas tarefas, operam satisfatoriamente em ambientes reais e podem evoluir continuamente, adaptando-se a um mundo em constante evolução (FIORINI, 1998).

Associando a esses objetivos, o termo engenharia pretende indicar que o desenvolvimento de software deve submeter-se a leis similares às que governam a manufatura de produtos industriais em engenharias tradicionais, pois ambos são metodológicos (MAFFEO, 1992).

Com base nos objetivos da Engenharia de Software, fica demonstrada a necessidade da adoção de um modelo sistêmico para padronizar e gerenciar os processos de desenvolvimento de software com o intuito de buscar qualidade nos processos e produtos de software.

## 4. CRISE DO SOFTWARE

No estudo da Engenharia de Software, o autor Roger S. Pressman, cita a “Crise do Software” onde se apresentam números que expressam o problema com a não finalização de projetos de software (PRESSMAN, 2006).

O mesmo autor aponta que um dos principais fatores que causam a tal “Crise do Software” é a falta de adoção de métodos, procedimentos e ferramentas na construção de software.



A expressão “Crise do Software”, que começou a ser utilizada na década de 60, tem historicamente aludido a um conjunto de problemas recorrentemente enfrentados no processo de desenvolvimento (Construção, implantação e manutenção) de software (MAFFEO, 1992).

Apesar da enorme variedade de problemas que caracterizam a crise do software, engenheiros de software e gerentes de projetos para desenvolvimento de sistemas computacionais tendem a concentrar suas preocupações no seguinte aspecto: “Há enorme imprecisão das estimativas de cronogramas e de custos de desenvolvimento” (LEE, 2002).

Muitos desses erros poderiam ser evitados se as organizações pudessem dispor de um processo de engenharia de software definido, controlado, medido e aprimorado. No entanto, percebe-se que para muitos profissionais de informática esses conceitos não são muito claros, o que certamente dificulta a ação dos gerentes no sentido de aprimorar os seus processos de produção (BLASCHEK, 2003).

Existem várias técnicas, metodologias e normas de qualidade para contribuir com o desenvolvimento do software, entre elas a Gestão de Riscos, e os profissionais que não as adotam, apresentam dificuldades de realizar projetos de software livres de manutenções e retrabalhos, condenando diretamente a qualidade dos produtos.

A adoção da Engenharia de Software direciona o aluno a realizar as atividades inerentes ao seu papel profissional através de métodos sistêmicos por todo o ciclo de vida do software, permitindo que o produto desenvolvido, represente os processos reais da empresa e que atenda de fato as suas necessidades.

## 5. CICLO DE VIDA DO SOFTWARE

Durante o ciclo de vida de software são executados vários processos, sendo que cada um contribui para atingir os objetivos de um estágio do ciclo (MACHADO, 2002).

Os modelos de ciclo de vida são categorizados pela definição de uma sequência de atividades pré-definidas, que têm como objetivo o desenvolvimento ou a manutenção de software (PRESSMAN, 2006).

Vários modelos foram criados e experimentados na produção de software, entre eles o modelo espiral de Boehm (BOEHM, 1988).

Embora não exista nenhum processo de software “ideal”, existem muitas oportunidades de trabalho para melhorá-lo, em muitas organizações. Os processos podem incluir técnicas desatualizadas ou podem não tirar vantagem das melhores práticas da Engenharia de Software (SOMMERVILLE, 2007).

A melhoria dos processos de software pode ser implantada de diferentes maneiras. Ela pode ocorrer por meio da padronização de processos, pois é a primeira etapa essencial na introdução de novos métodos e novas técnicas de engenharia de software (SOMMERVILLE, 2007).

## 6. QUALIDADE DE SOFTWARE

Atingir um alto nível de qualidade de produto ou serviço é o objetivo da maioria das organizações. Atualmente não é mais aceitável entregar produtos com baixa qualidade e reparar os problemas e as deficiências depois que os produtos foram entregues ao cliente (SOMMERVILLE, 2007).

Para muitos engenheiros de software, a qualidade do processo de software é tão importante quanto à qualidade do produto. Assim na década de 90 houve uma grande



preocupação com a melhoria no processo de software. Abordagens importantes como as normas NBR ISO 9000 e a NBR ISO / IEC 12207, o modelo CMMI (Capability Maturity Model Integration) e o SPICE (Software Process Improvement and Capability dEtermination) sugerem que melhorando o processo de software, pode-se melhorar a qualidade dos produtos (MACHADO, 2001).

A qualidade é consequência dos processos, das pessoas e da tecnologia. A relação entre e qualidade do produto e cada um desses fatores é complexa. Por isso, é muito mais difícil controlar o grau de qualidade do produto do que controlar os requisitos (PAULA FILHO, 2009).

Os profissionais que ajustarem seus processos para a produção de software de qualidade dentro de prazos e orçamentos confiáveis, quando forem submetidos à concorrência, obrigando a redução substancial dos prazos para a entrega de produtos, e tiverem capacidade de integrar, harmonizar e acelerar o processo de desenvolvimento de software, terão a primazia do mercado (MACHADO, 2001).

Visando a qualidade no processo de produção de software, a gestão dos riscos tem o foco de tratar as incertezas inerentes aos projetos de software, pois muitos fatores que envolvem tecnologia, pessoas e processos, se conflitam e podem determinar ou não o sucesso no desenvolvimento do produto de software.

## 7. RISCOS E A ENGENHARIA DE SOFTWARE

Risco, como ciência, nasceu no século XVI, no Renascimento. Foi numa tentativa de entender os jogos de azar que Blaise Pascal (Figura 1), em 1654, descobriu a “Teoria da Probabilidade” e criou o “Triângulo de Pascal”, que determina a probabilidade de ocorrer possíveis saídas, dado certo número de tentativas (BERNSTEIN, 1997).



Figura 1 – Blaise Pascal (BERNSTEIN, 1997).

No século XX, a gerência de risco foi difundida, estudada e utilizada principalmente nas áreas de saúde, finanças, seguro de vida e etc. Para essas empresas, a gerência de risco não é coisa ruim, ao contrário, a gerência de riscos é o negócio. Todos os projetos nessas áreas tratam riscos, pois os lucros dependem de oportunidades atrativas, balanceadas por riscos bem calculados (BERNSTEIN, 1997).

O risco na área de software foi representado de forma sistemática por Barry Boehm, nos anos 80, através do modelo em Espiral, que tem como princípio ser iterativo e dirigido a riscos, pois cada iteração é feita uma análise de risco (BOEHM, 1988).



Cada iteração do modelo espiral é dividida em quatro setores:

- Definição de objetivos;
- Avaliação e redução de riscos;
- Desenvolvimento e validação;
- Planejamento;

A escolha do modelo espiral para embasar esta pesquisa, dentre vários modelos de ciclo de vida de software, refere-se a sua ênfase e sua relação direta com a análise de riscos, conforme demonstrado na figura 2.

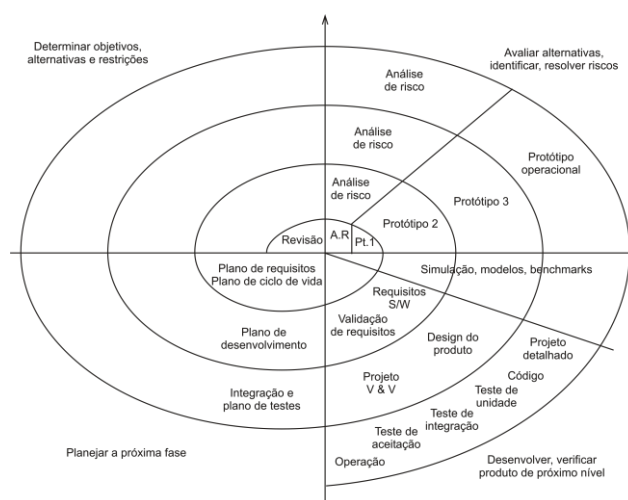


Figura 2 – Adaptado do Modelo Espiral de Boehm (BOEHM,1988).

A importante distinção entre o modelo em espiral e outros modelos de processo de software é a explícita consideração dos riscos (SOMMERVILLE, 2007).

Os riscos, em software, não podem ser meros tópicos da agenda. Devem ser o “coração” do negócio, como ocorre em outras áreas (CHADBOURNE, 1999).

Atualmente, a área que trata de riscos na engenharia de software evoluiu, passando de uma análise dentro do modelo de desenvolvimento, como era a proposta do modelo em espiral, para se tornar uma gerência que deve permear todos os processos do ciclo de vida do software.

A gerência de risco é entendida como um procedimento geral para a resolução de riscos, ou seja, quando for aplicada em alguma instância, as possíveis conseqüências são todas aceitáveis, podendo haver convivência com o pior resultado esperado. O risco é apresentado de alguma forma e em algum grau na maioria das atividades humanas e é caracterizado por ser parcialmente conhecido, mudar com o tempo e ser gerenciável no sentido que uma ação humana pode ser aplicada para mudar a sua forma e o grau do seu efeito. O processo de gerência de risco inicia com incertezas, preocupações, dúvidas e desconhecimentos que se transformam em riscos aceitáveis (MACHADO, 2002).

O gerenciamento de riscos é um processo de software, definido e sistemático com o objetivo de tratar os fatores de risco com a finalidade de mitigar ou minimizar seus efeitos, produzindo um produto de software com qualidade, que atenda as necessidades do cliente, dentro do prazo e custos estimados (NOGUEIRA, 2009).





## 8. MOTIVAÇÃO DO TRABALHO

Através de um estudo chamado "Chaos Report", para projetos na área de Tecnologia da informação, obteve as seguintes conclusões (STANDISH GROUP, 2009):

- 32% dos projetos terminam no prazo e no orçamento previsto;
- 44% dos projetos são contestados.
- 24% são cancelados antes de sua implantação.

Tabela 1 – Adaptado do “Chaos Report” de 1994 á 2009.

Chaos Report - 2009 - Standish Group								
Projetos / Ano	1994	1996	1998	2000	2002	2004	2006	2009
Bem Sucedidos	16	27	26	28	34	29	35	32
Contestados	53	33	46	49	51	53	46	44
Cancelados	31	40	28	23	15	18	19	24
Fracassados	84	73	74	72	66	71	65	68

Fonte: STANDISH, 2009.

Num mundo cada vez mais de recursos financeiros escassos, como é possível aceitar tal desperdício de tempo e dinheiro. Números desse porte podem comprometer seriamente a faceta competitiva de uma organização, bem como ocasionar até a sua saída do mercado.

Apesar da “Crise do Software” não ser um problema novo, até hoje se enfrenta os impactos e seus efeitos negativos. A baixa utilização de metodologias e modelos de qualidade no Brasil denota esta realidade que tem que ser modificada.

Segundo o Ministério da Ciência e Tecnologia, apenas 11,8 % das empresas no Brasil adotam a gestão de riscos em projetos de software (MCT, 2002).

Devido à relevância do tema e seu impacto direto no sucesso na produção de software (NOGUEIRA, 2009), o número apresentado pelo ministério é alarmante, pois a amostra utilizada pela pesquisa apresenta desde as principais empresas de software do país até as pequenas e médias empresas.

O fato preocupante é que as pequenas e médias empresas, que detêm 65,1% do mercado de software no Brasil (MCT, 2002), se não possuir a cultura de gestão de riscos, além da possibilidade de fracassar nos atuais projetos, comprometem o futuro ainda promissor que a área tem que explorar tanto no mercado interno, bem como no mercado externo.

Novas pesquisas em 2005 e 2008 foram feitas pelo Ministério da Ciência e Tecnologia, mas o item, gestão de riscos, não foi adicionado à pesquisa.

## 9. ENGENHARIA DE RISCOS

Segundo Robert Charette, a definição de risco é: Em primeiro lugar, risco afeta acontecimentos futuros. Presente e passado não preocupam, pois o que colhemos hoje já foi semeado por nossas ações anteriores. A questão é mudando nossas ações hoje, podemos criar oportunidade para uma situação diferente e possivelmente melhor para nós amanhã? Isso significa, em segundo lugar, que risco envolve mudança, como por exemplo, mudança de pensamento, opinião, ações ou lugares..., e em terceiro lugar, o risco envolve escolha e a



incerteza que a própria escolha envolve. Assim, paradoxalmente, o risco, como a morte e os impostos, é uma das poucas certezas da vida (CHARETTE, 1989).

Peter Drucker disse certa vez, já que é fútil tentar eliminar riscos e questionável tentar minimizá-los, o essencial é que os riscos considerados sejam os certos. Antes que possamos identificar os "riscos certos", que acontecerão durante um projeto de software, é importante identificar todos os demais que são óbvios, tanto para gerentes quanto para profissionais (PRESSMAN, 2006).

Segundo Higuera, um risco 100% provável é uma restrição ao projeto de software.

De modo simplificado, podemos pensar no risco como uma probabilidade de que alguma circunstância adversa realmente venha a ocorrer. Os riscos podem ameaçar o projeto, o software que está sendo desenvolvido ou a organização (HIGUERA, 1995).

Essas categorias de riscos podem ser definidas como se segue (SOMMERVILLE, 2007):

1. Riscos relacionados ao Projeto: São os riscos que afetam a programação ou os recursos do projeto;
2. Riscos relacionados ao Produto: São os riscos que afetam a qualidade ou o desempenho do software que está em construção;
3. Riscos para os negócios: São os riscos que afetam a organização que está construindo ou adquirindo o software.

A engenharia de riscos envolve duas áreas chaves no seu processo. São elas: análise de riscos e gerenciamento de riscos (PETERS, 2001).

Segundo o mesmo autor, a análise de riscos é composta por 3 processos:

- Identificação dos Riscos
- Estimativa dos Riscos
- Avaliação dos Riscos

E o gerenciamento dos riscos é composto por 5 processos:

- Planejamento de riscos
- Controle dos riscos
- Monitoração dos riscos
- Direcionamento de riscos
- Recrutamento (Respostas aos riscos)

O risco de projeto pode ser estimado quantitativamente ou qualitativamente. O principal objetivo da análise de riscos é desenvolver um conjunto de estratégias de prevenção de riscos (IEEE, 1995).

Os riscos devem ser estimados e monitorados (PAULA FILHO, 2009).

A estimativa de riscos é uma atividade muito importante e pouco praticada; um bom planejamento não apenas faz a previsão do que deve acontecer se tudo correr bem, mas também o que pode correr mal, quais as conseqüências dos problemas e o que pode ser feito para combatê-los. Entre os fatores de riscos que devem ser considerados podem ser incluídos:

- Riscos legais;
- Riscos Tecnológicos;
- Riscos devidos ao tamanho e à complexidade do produto;
- Riscos relativos ao pessoal envolvido no projeto;
- Riscos relativos à aceitação pelos usuários;

Já Sommerville descreve os tipos de riscos que podem afetar o projeto e do ambiente organizacional em que o software está sendo construído. Contudo, muitos riscos são considerados universais e eles envolvem as seguintes áreas: Tecnologia; Pessoal; Organizacional; Ferramentas; Requisitos e Estimativa (SOMMERVILLE, 2007).



A estimativa dos riscos compreende as seguintes tarefas:

- Identificação dos riscos possíveis em relação ao projeto;
- Análise desses riscos, avaliando-lhes a probabilidade e o provável impacto;
- Previsão de contramedidas curativas ou preventivas;
- Priorização dos riscos, organizando-os de acordo com a probabilidade e o impacto.

Os riscos não permanecem constantes durante a execução de um projeto.

Alguns desaparecem, outros novos surgem, e outros sofrem alterações de probabilidade e impacto, mudando, portanto a prioridade. Um relatório de acompanhamento do projeto juntamente com uma tabela atualizada, devem ser utilizados para a monitoração dos riscos. A tabela de estimativa deve ser repetida e atualizada para refletir as modificações ocorridas, até que os riscos sejam concretizados ou completamente eliminados (PAULA FILHO, 2009).

A adoção da Engenharia de riscos faz parte dos fatores críticos de sucesso nos projetos de software. A gestão dos riscos em todo o ciclo de vida do desenvolvimento é fundamental para o sucesso do projeto (NOGUEIRA, 2009).

## 10. GESTÃO DE RISCOS

Atualmente, há uma grande expectativa acerca das aplicações da tecnologia da informação nas empresas, graças às novas alternativas para os negócios que elas proporcionam e às melhorias que trazem aos processos existentes. Contudo há um questionamento sobre os ganhos reais advindos dos investimentos maciços em tecnologia, pois, muitas vezes, o retorno fica aquém do esperado (NOGUEIRA, 2008).

Para a maioria das organizações, a informação e a tecnologia que suportam os negócios representam seus ativos mais valiosos. Em um ambiente cada vez mais competitivo e dinâmico, a administração requer qualidade, funcionalidade e facilidade no uso dos recursos de TI, assim como alta disponibilidade a custos mais baixos. Não há dúvidas quanto aos benefícios do uso da tecnologia, entretanto, para ser bem sucedida, uma organização deve adotar um modelo de gestão que possibilite a eficácia e eficiência da TI. É nesse contexto que a gestão de riscos em projetos de software surge como um dos processos essenciais, propiciando à representação das melhores práticas e ainda aderentes a governança de TI no que se diz respeito a controle, transparência e monitoração (NOGUEIRA, 2008).

A gestão de riscos é particularmente importante para projetos de software, devido às incertezas inerentes que a maioria dos projetos enfrenta (SOMMERVILLE, 2007).

A gestão de riscos é composta por atividades coordenadas para direcionar uma organização em relação ao risco. A gestão de riscos, geralmente inclui avaliação, tratamento, aceitação e comunicação de riscos (MCT, 2002).

O processo de gestão de riscos envolve vários estágios ou atividades (SOMMERVILLE, 2007):

1. Identificação dos riscos: São identificados os possíveis riscos de projeto, produto e negócios;
2. Análise de riscos: São avaliadas as possibilidades e as conseqüências da ocorrência desses riscos;
3. Planejamento de riscos: São traçados planos para enfrentar os riscos, seja evitando-os, seja minimizando seus efeitos sobre o projeto;
4. Monitoramento de riscos: O risco é constantemente avaliado e os planos para a diminuição dos riscos revisados, à medida que mais informações sobre eles se tornam disponíveis.





Os gerentes de projetos de sistemas de informação deveriam avaliar regularmente os riscos durante o processo de desenvolvimento para minimizar as chances de fracassos (PIVETTA, 2002).

Em particular, os problemas de cronograma, orçamento e funcionalidade dos softwares não podem ser totalmente eliminados, mas podem ser controlados através da aplicação de ações preventivas (HIGUERA, 1996).

A gestão de riscos possui seis atividades bem definidas. São elas: Identificação dos Riscos, Análise, Planejamento, Monitoração, Controle e Comunicação (HIGUERA, 1996).

A NBR ISO 31.000 (2009) direciona a política para a gestão de riscos com as seguintes atividades: Estabelecimento do contexto, identificação de riscos, análise de riscos, avaliação de riscos, tratamento de riscos, comunicação e consulta e monitoramento e análise crítica.

## 11. PROCESSO DE RISCOS

Após o levantamento bibliográfico realizado, foi possível identificar áreas críticas no processo de desenvolvimento de software.

No entanto, para apoiar a gestão de riscos nos projetos de software, faz-se necessário à utilização de um fluxo de trabalho com as atividades, onde o tomador de decisão possa utilizá-lo como um instrumento de auxílio no processo de gerenciamento dos riscos (Nogueira, 2009).

Portanto, as atividades a seguir compõem este fluxo de trabalho:

- Comunicação e consulta;
- Estabelecimento de contextos;
- Identificação de riscos;
- Análise de riscos;
- Tratamento de riscos;
- Avaliação de riscos;
- Monitoramento e análise crítica.

Estas atividades serão descritas a seguir de acordo com a sua complexidade de aplicação, em conformidade com a ISO 31000.

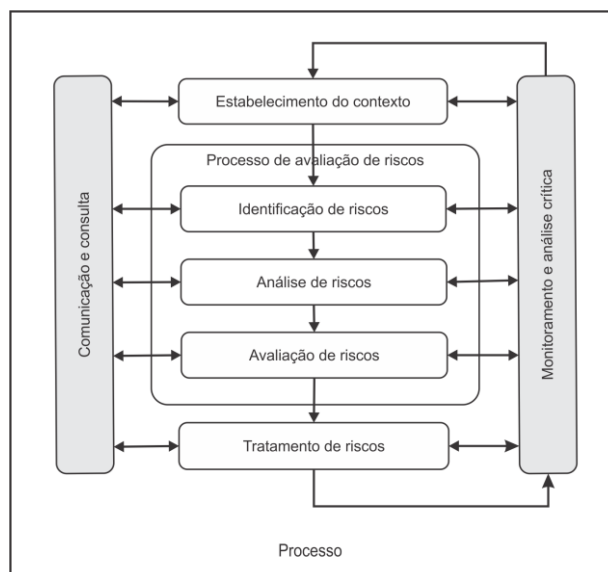


Figura 3 – Processo de Gestão de Riscos – ISO 31.000:2009.



### 11.1. Comunicação e consulta

A comunicação e consulta com as partes interessadas, externas e internas, devem acontecer durante todas as fases do processo de gestão de risco. No início, com a primeira reunião de sensibilização. Durante as atividades. E no final, a apresentação dos resultados.

### 11.2. Estabelecimento do contexto

Ao estabelecer o contexto, a organização articula seus objetivos e define os parâmetros externos e internos a serem levados em consideração na gestão de riscos, e define o escopo e os critérios de risco para o restante do processo.

### 11.3. Identificação de riscos

A organização deve identificar as fontes de risco, áreas de impactos, eventos e as suas causas e suas possíveis consequências. O objetivo desta etapa é gerar uma lista abrangente de riscos com base nesses eventos que possam criar, melhorar, prevenir, diminuir, acelerar ou atrasar a realização dos objetivos. Uma identificação abrangente é crítica, porque o risco de que não é identificado, nesta fase, não será incluído em análise posterior. Recomenda-se a utilização de um framework com os riscos universais, comuns a vários projetos, quando se tratar da primeira iteração.

### 11.4. Análise de riscos

A análise de risco envolve desenvolver uma compreensão do risco. A análise de risco fornece uma entrada para avaliação de riscos e para as decisões sobre a necessidade dos riscos serem tratados, e sobre as estratégias e métodos mais adequadas ao tratamento dos riscos. A análise de riscos também pode fornecer uma entrada para a tomada de decisões em que as escolhas precisam ser feitas e as opções envolvem diferentes tipos e níveis de risco. Um framework de riscos universais estabelecidos a partir de um peso atribuído por especialistas, especialmente quando você não tem uma base de conhecimento.

### 11.5. Avaliação de riscos

O objetivo da avaliação de risco é auxiliar na tomada de decisões, com base nos resultados da análise de risco, sobre quais os riscos necessitam de tratamento e a prioridade para o tratamento.

### 11.6. Tratamento de riscos

Tratamento de riscos envolve a seleção de uma ou mais opções para modificar os riscos e implementação dessas opções. Uma vez implementado, o tratamento fornece novos controles ou modifica existentes.

### 11.7. Monitoramento e análise crítica

Acompanhamento e avaliação deve ser uma parte planejada do processo de gestão de riscos e envolvam verificação regular. Pode ser periódica ou acontecer em resposta a um fato específico.



## 12. CONSIDERAÇÕES FINAIS

Com o levantamento bibliográfico, foi possível identificar que os autores reconhecem as dificuldades existentes no processo de desenvolvimento de software. Mesmo apresentando formas e procedimentos para facilitar a compreensão, bem como difundir a utilização desses métodos, nem sempre isso acontece, pois eles mesmos reconhecem que nenhum método é infalível, devido à complexidade dos problemas a serem solucionados na construção de um sistema de informação. No entanto, diante de tantas incertezas inerentes aos projetos de software, foi identificada a necessidade da elaboração de um roteiro com as atividades da gestão de riscos com o objetivo de criar cultura e disseminá-la para os alunos de graduação e pós-graduação, pois a aplicabilidade formal nas empresas ainda é muito baixa. Com a ISO 31000, o escopo pode ser delimitado e validado. Como trabalho futuro, pretende-se apresentar a aplicação deste roteiro aqui exposto, em projetos acadêmicos que nunca utilizaram a gestão de riscos e colher todos os resultados, positivos e negativos.

## REFERÊNCIAS BIBLIOGRÁFICAS

- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR ISO 31000: Gestão de riscos: Princípios e diretrizes. Rio de Janeiro, 2009.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR ISO/IEC 12207: Tecnologia de Informação: Processo de ciclo de vida de software. Rio de Janeiro, 1997.
- BERNSTEIN, Peter. Desafio aos deuses: a fascinante história do risco. Rio de Janeiro: Campus, 1997.
- BLASCHEK, JOSÉ ROBERTO. Gerência de Requisitos, o principal problema dos projetos de software. Artigo Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2003.
- BOEHM, Barry. A spiral model of software development and enhancement. IEEE Computer, V.21, n.5, p.61-72, 1988.
- CHADBOURNE, Bruce C. To the heart of risk management: teaching project teams to combat risk. In: Annual Project Management institute seminar & symposium, Pennsylvania, 1999.
- CHARETTE, R. N. Software Engineering risk analysis and management. McGraw Hill, 1989.
- DRUCKER, P. Management. 1975 apud PRESSMAN, Roger S. Engenharia de Software. Rio de Janeiro: McGraw Hill, 2002. 5º ed.
- Fiorini, Soeli T., et al. Engenharia de Software com CMM. Rio de Janeiro: Brasport, 1998.
- HIGUERA, R. P.; HAIMES, Y. Y. Software risk management technical report CMU/SEI 96 TR 012. SEI Carnegie Mellon University, 1996.
- IEEE 1044.1-1995, IEEE Standard Glossary of Software Engineering Terminology, IEEE, 1995.
- Lee, Richard C.; TEPFENHART, William M. UML e C++: Guia de desenvolvimento orientado a objeto. São Paulo: Makron Books, 2002.
- MACHADO, Cristina Ângela Filipak. A-Risk: um método para identificar e quantificar risco de prazo em projetos de desenvolvimento de software. 2002. 239 f. Dissertação (Mestrado) – Programa de Pós-Graduação em Informática Aplicada, Pontifícia Universidade Católica do Paraná, Curitiba.
- MACHADO, Cristina Ângela Filipak. NBR ISO/IEC 12207: Processos de ciclo de vida do software. In: ROCHA, Ana Regina Cavalcanti; MALDONADO, José Carlos; WEBER, Kival Chaves. Qualidade de Software: Teoria e Prática. São Paulo: Prentice Hall, 2001. p. 9-16.



- Maffeo, Bruno. Engenharia de Software e Especificação de Sistemas. Rio de Janeiro: Campus, 1992.
- MINISTÉRIO DA CIÊNCIA E TECNOLOGIA. Secretaria de Política de Informática. Qualidade e Produtividade no Setor de Software Brasileiro. Brasília, 2002
- NOGUEIRA, Marcelo, Abe, Jair Minoru, Normas e modelos de qualidade como política de produção de software no contexto brasileiro. In: XV SIMPEP Simpósio Engenharia de Produção, 2008, Bauru. Anais. Bauru: UNESP, 2008.
- Nogueira, Marcelo. Engenharia de Software. Um Framework para a Gestão de Riscos em Projetos de Software, Rio de Janeiro, Ed. Ciência Moderna, 2009.
- PAULA FILHO, Wilson de Pádua. Engenharia de Software: fundamentos, métodos e padrões. 3º ed. Rio de Janeiro: LTC, 2009.
- Peters, James F.; Pedrycz, Witold. Engenharia de Software: Teoria e prática. Rio de Janeiro: Elsevier, 2001.
- PIVETTA, Valdimir Uliana. Modelo de apoio à gestão de riscos no desenvolvimento de software. 2002. 103 f. Monografia (MBA) – Programa de MBA em Engenharia de Software, Escola Politécnica da Universidade de São Paulo, São Paulo.
- Pressman, Roger S. Engenharia de Software. 6. ed. São Paulo: McGraw-Hill, 2006.
- Pressman, Roger S. Engenharia de Software. 6. ed. São Paulo: McGraw-Hill, 2006.
- Sommerville, Ian. Engenharia de Software. 8. Ed. São Paulo: Pearson Addison Wesley, 2007.
- STANDISH, CHAOS Report 1995, Standish Group, Boston, 1995, Disponível: <http://www.standishgroup.com> Acesso em 18/10/2009.
- STANDISH, CHAOS Summary 2009, Standish Group, Boston, 2009, Disponível: [http://www1.standishgroup.com/newsroom/chaos\\_2009.php](http://www1.standishgroup.com/newsroom/chaos_2009.php) Acesso em 18/10/2009.

## THE IMPORTANCE OF THE ADOPTION OF RISK APPROACH IN TEACHING OF SOFTWARE ENGINEERING

*Abstract: In a global market increasingly competitive, software engineers, to meet the business needs which are submitted, perform tasks and procedures with high risk exposure, not always known and measured. Given that a minority of companies adopts the risk management processes, such exposure may affect the participation and success of the projects involved. To ensure the quality of the process, a risk-focused approach should be adopted systematically. Among the uncertainties of software projects, some risk factors should be treated. The duration and estimated costs, compliance with business requirements, among others, can be cited. Through a literature search was possible to compose a script risk activities to provide appropriate treatment. To contribute to the teaching of these issues involving software projects, this work presents a risk management process in order to enter the culture and capacity in the professionals who serve them, and may direct as objectively mitigate the risks they are exposed. Complete the approach, the fact that this script be in accordance with ISO 31000:2009.*

*Key-words: Software Engineering, Risk Management, Quality processes.*