Capítulo 1 - Introducão

How risky are software projects? to answer this question, several studies about effectiveness of software cost, scope, schedule estimation techniques, beyond of surveys from software professionals in industry and also analysis of project portfolios have been done. However, there is not a consensus. \cite{budzier2013double}. However, there is not a consensus.

Todos os projetos envolvem risco. Há sempre pelo menos algum nível de incerteza no resultado de um projeto, independentemente do que o gráfico de Gannt pareça indicar.na parede parece indicar. Projetos de alta tecnologia são particularmente arriscados, por uma série de motivos. Primeiro, há uma grande variação de projetos técnicos. Esses projetos têm objetivos e aspectos únicos que os diferem significativamente dos trabalhos anteriores, além de apresentar um ambiente de projetos que evolue rapidamente. Além disso, projetos técnicos são frequentemente desafiados a trabalhar com orçamento financiamento, pessoal e equipamentos inadequados. Para piorar a situação, há uma expectativa generalizada que não corresponde à realidade de que por mais rápido que tenha sido o último projeto, o próximo deve ser ainda mais rápido \cite{kendrick2003identifying}.

Projetos que obtiveram sucesso geralmente conseguiram isso porque duas das ações tomadas por seus líderes foram determinantes. Primeiro, eles reconheceram que alguns dos trabalhos em qualquer projeto, mesmo projetos de alta tecnologia, não são novos. Nos trabalhos por eles desenvolvidos as notas, registros e lições aprendidas em projetos anteriores puderam ser utilizados como um roteiro para identificar, e em muitos casos evitar, muitos problemas potenciais. Segundo, eles planejaram com afinco o trabalho do projeto a fundo, especialmente as partes que exigiam inovação, para possibilitar a compreensão dos desafios futuros e antecipar muitos dos riscos \cite{kendrick2003identifying}.

Alguns dos benefícios obtidos quando da ocorrência de uma boa gestão de riscos de projetos de software, são:

* redução de custos incorridos com mudanças no software;
* desenvolvimento de um plano de respostas a eventos inesperados, também denominado plano de contingência de riscos;
* previsão da probabilidade da ocorrência de eventos indesejados;
* seguimento das linhas de base de custo, cronograma e qualidade.

Tais fatores podem determinar o sucesso dos projetos \cite{HIGUERAHAIMES1996} \cite{PMBOK2008}.

# Motivation

Em 2009, o CHAOS Report \cite{CHAOS2009} mostrou que 32\% dos projetos de software alcançaram sucesso, foram entregues no prazo,de acordo com o orçamento estabelecido e com os requisitos prometidos; 44\% dos projetos foram desafiados (explicar); não menos importante, 24\% dos projetos falharam e foram cancelados. Isso ocorre devido aos riscos envolvidos nas atividades e a um gerenciamento de risco de software ausente ou deficiente \cite{ISLAM2009}.

Schmidt e outros autores \cite{schmidt2001identifying} notaram que muitos dos projetos de desenvolvimento de software terminavam com falha. Eles mostraram que cerca de 25\% de todos os projetos de software são cancelados e cerca de 80\% de todos os projetos de software ultrapassaram atropelaram seus orçamentos, excedendo-os em 50\% na média. Paul Bannerman \cite{bannerman2008risk} afirma que pesquisas na indústria sugerem que somente um quarto dos projetos de software tem sucesso imediato, e bilhões de dólares são perdidos anualmente por meio de falhas ou projetos que não cumprem a entrega dos benefícios prometidos. Além disso, o autor mostra evidências de que isso é um assunto global, impactando organizações do setor privado e público \cite{KPMG2005}.

A previsão de possíveis eventos a curto, médio e longo prazos muitas vezes é falha. Ao analisar os riscos e as incertezas, os gerentes de projeto muitas vezes confiam na própria intuição, em vez de utilizarem a lógica e uma análise detalhada. No entanto, o pensamento intuitivo é frequentemente alvo de ilusões, que causam erros mentais previsíveis e decisões eventualmente não embasadas pobres. O método para conciliar o efeito dessas ilusões psicológicas é uma avaliação sistemática dos riscos e esforços na mitigação dos mesmos através de métodos analíticos.

É difícil gerenciar algo que não pode ser medido. Gerentes de projeto devem quantificar a probabilidade de risco, os resultados, e seu efeito cumulativo em um projeto. Além disso, é importante avaliar as várias opções de mitigação: o custo de cada opção e o tempo necessário para a sua realização \cite{VIRINE2009}.

Existe uma dificuldade na interpretação do conceito de risco, principalmente quanto a aplicação desse conhecimento no desenvolvimento e utilização de técnicas eficientes para a análise de risco no gerenciamento de projetos de software. A gestão de riscos e incertezas em projetos de software, é fundamental para a disciplina de gerenciamento de projetos. Entretanto, em momentos econômicos de crise torna-se muito mais difícil realizar o gerenciamento de riscos, devido aos custos incorridos.

Esta é uma área de pesquisa que vem evoluindo e, portanto, novas e melhores metodologias para identificação, medição e controle de itens de risco de software precisam ser desenvolvidas. Segundo Keshlaf e Riddle \cite{KESHLAFRIDDLE2010}, mesmo que existam muitas abordagens ainda há uma grande lacuna com relação ao que é praticado pelas indústrias de software.

# Problem Description

Percepções recentes sobre o gerenciamento de risco e os seus desafios inerentes, devido a natureza dos projetos de software, contribuem para a falta de estabilidade do projeto da maioria das organizações de projetos de software \cite{kwak2004project}. Conforme identificado por Kwak e Ibbs \cite{kwak2000calculating}, o gerenciamento de risco é a disciplina menos praticada dentre as diferentes áreas do conhecimento no gerenciamento de projetos. Os autores mencionam que, provavelmente, um motivo é que desenvolvedores de software e gerentes de projetos consideram gerenciar processos e atividades que envolvam incerteza como trabalho e custo extras.

Barry Boehm \cite{BOEHM1991} definiu risco como a possibilidade de perda ou dano. Essa definição pode ser expressa pela fórmula de exposição ao risco. Mesmo que Boehm cite a exposição ao risco como a técnica mais efetiva para a priorização do risco depois de sua análise, Paul Bannerman \cite{bannerman2008risk} considera essa definição limitada e inapropriada. Na teoria clássica da decisão, risco reflete a variação na distribuição de probabilidade de possiveis resultados, seja negativo ou positivo, associado a uma decisão particular. Esse estudo leva em consideração a definição do \textit{Project Management Institute} \cite{PMBOK2008} em que risco em projeto é um evento ou condição específica que, se ocorrer, tem um efeito positivo ou negativo em ou mais objetivos do projeto. Uma definição complementar proposta por Haimes \cite{haimes2011risk} também é considerada, a qual expressa o risco como uma medida da probabilidade e severidade de efeitos adversos.

Um fator de risco é uma variável associada com a ocorrência de um evento inesperado. Fatores de risco são correlacionados, não necessariamente causais e, se um deles ocorrer, implicará em um ou mais impactos. De acordo com Haimes \cite{haimes2011risk}, riscos podem comumente surgir como o resultado não apenas de um processo estocástico embutido ocorrendo no passar do tempo e no espaço, mas também, pode ocorrer baseado em fatores de riscos determinísticos. A estimativa do risco pode ser alcançada baseada em informações históricas ou e conhecimento de projetos anteriores similares, ou ainda e de outra fonte de informação \cite{PMBOK2008}.

Risco é um conceito que muitos consideram difícil de ser compreendido por envolver duas métricas complexas: uma medida de probabilidade e uma medida de severidade dos efeitos adversos \cite{Haimes2009}. Uma limitação dessa definição é a dificuldade prática em se estimar a probabilidade e o impacto de diversos fatores de risco, especialmente em projetos de software. As probabilidades somente podem ser definidas de um modo preciso para atividades que são repetidas muitas vezes sob circunstâncias controladas. No entanto, a natureza única de muitas atividades de projetos de software não permite a estimativa precisa de suas probabilidades. Outra limitação dessa definição é que ela abrange somente ameaças conhecidas ou previsíveis, oferecendo opções limitadas para gerenciar ameaças não percebidas, além de não reconhecer ameaças imprevisíveis. Essa é uma consequência da definição de risco em termos de probabilidade e impacto, uma vez que para se avaliar a probabilidade e o impacto é necessário que se tenha a capacidade de se prever uma eventualidade. Existe ainda uma outra questão onde se questiona se as melhores decisões são baseadas na quantificação numérica, determinada pelos padrões do passado, ou na avaliação subjetiva das incertezas. Não é possível quantificar o futuro com certeza, mas através da probabilidade, é possível prevê-lo a partir do passado. Apesar de ser difícil encontrar um projeto de software padrão, é possível classificar atividades e definir padrões que possibilitem a estimativa. Para Bannerman \cite{bannerman2008risk}, a solução comum para esse problema em projetos de software consiste em se observar o risco de um modo mais geral, em termos da incerteza, e avaliá-lo qualitativamente.

Duas das premissas consideradas por Haimes \cite{Haimes2009} na pesquisa de análise de risco também serão consideradas neste estudo. A primeira delas considera que o risco é comumente quantificado através da fórmula matemática de expectativa. No entanto, mesmo que essa fórmula permita uma medida valiosa do risco, ela falha em reconhecer e/ou acentuar consequências de eventos extremos. Tom Kendrick apresenta no seu livro \cite{KEND2003BOOK} um \textit{framework} para a identificação e gerenciamento de catástrofes. A outra premissa, por sua vez, afirma que uma das tarefas mais difícil em análise de sistemas é saber como modelar o risco modelá-lo. Portanto, novas propostas para a análise quantitativa e modelagem de sistemas, sob o ponto de vista de seus riscos, poderão contribuir para o avanço científico da área.

Embora o gerenciamento de risco na gestão de projetos de software seja um processo saudável, sua utilização ainda está aquém das expectativas. Algumas causas disso são o acúmulo de responsabilidades dos gerentes de projetos, a baixa importância atribuída a essa área, a falta de conhecimento em gestão de riscos, os custos envolvidos nas atividades de gestão de risco, a falta de habilidade para lidar com as técnicas e ferramentas específicas. [ REFERÊNCIA] Como consequência, o projeto está sujeito à influência negativa de riscos sem haver um plano de contingência, o que pode ocasionar o fracasso do projeto. De acordo com o benchmarking realizado em 2009 pelo Project Management Institute, em 20% dos projetos os seus gerentes não realizaram todos os processos de planejamento e em apenas 35% dos projetos o gerenciamento de riscos foi realizado de acordo com uma metodologia formal, estruturada por políticas, procedimentos e formulários. Além disso, 46% dos gerentes realizaram atividades de gerenciamento em tempo parcial.

A necessidade de gerenciar riscos (eventos indesejados) tem crescido cresceu exponencialmente com a complexidade dos sistemas. O gerenciamento nesses sistemas complexos apresenta dificuldade na identificação e na estimativa de ocorrência desses eventos indesejáveis, esperados ou inesperados, por conta da imensa quantidade de fatores de riscos envolvidos e suas relações. Há uma necessidade crescente por métodos mais sistemáticos e por ferramentas para suplementar o conhecimento individual, o julgamento e a experiência. Essas características humanas são muitas vezes suficientes para enfrentar riscos de menor complexidade e que se manifestam de forma isolada. Por exemplo, uma parte dos problemas mais sérios encontrados na aquisição de um sistema é resultante os resultados de riscos que são ignorados devido a sua baixa probabilidade, até que eles criem consequências mais sérias \cite{higuera1996software}.

O guia para o gerenciamento de projetos de PMBOK \cite{PMBOK2008} apresenta a simulação de Monte Carlo como um método de boa prática para a análise de risco de projetos. No entanto, existem algumas limitações na adoção dessa abordagem que o torna inviável \cite{Ibbotson2005}. As simulações podem levar a resultados enganosos se entradas inapropriadas, derivadas da parametrização subjetiva, forem são inseridas no modelo. Comumente, o usuário deve estar preparado para realizar os ajustes necessários se os resultados que forem são gerados apresentarem-se parecem fora de rumo. Além disso, a simulação de Monte Carlo não pode modelar correlações entre riscos. Isso significa que os números que surgem em cada sorteio são aleatórios e, em consequência, um resultado pode variar de seu valor mais baixo, em um período, para o mais alto no próximo. Portanto, abordagens alternativas devem ser consideradas para prever a probabilidade de risco e impacto, levando em consideração as características de risco do projeto e as limitações da simulação de Monte Carlo. Assim, a análise de risco deve ser uma tarefa mais precisa e de maior facilidade de uso do ponto de vista dos usuários. Esse trabalho considera as redes neurais artificiais como uma alternativa válida e importante valiosa a ser considerada na análise de risco de projetos de software.

# Objectives

# Research Questions

Como definir e modelar os riscos no gerenciamento de projetos de software, também considerando as catástrofes?

Como analisar quantitativamente os riscos no gerenciamento de projetos de software?

Quais os dados disponíveis de registros de riscos de projetos de software para realizar os estudos?

Como desenvolver um modelo para previsão de riscos em gerenciamento de projetos de software eficiente para o suporte a tomada de decisão?

# General Objectives

O principal objetivo dessa dissertação consiste na avaliação da abordagem mais eficiente para a análise de riscos em projetos de software, levando-se em conta os seguintes modelos: Simulação de Monte Carlo (SMC), Modelos de Regressão Linear (MRL's) e Redes Neurais Artificiais (RNA's) – representadas pelas redes Perceptron de Múltiplas Camadas (MLP), Máquina de Vetor de Suporte(SVM) e Redes de Função de Base Radial (RBF). A abordagem mais eficiente será aquela que melhorará a precisão e diminuir a possibilidade de erro na estimativa de impacto de riscos.

# Specific Objectives

* Avaliar abordagens tradicionais de estimativa de impacto de riscos no que se refere a diminuição do erro de previsão;
* Desenvolver uma pequena base de dados a partir da experiência de um coordenador de projetos, com o auxílio da base de dados do PERIL [Referência;
* Desenvolver uma metodologia ~~técnica~~ para prever o impacto de riscos, através de uma rede neural artificial, para gerenciar os riscos num projeto de software

O primeiro objetivo específico está relacionado a análise da Simulação de Monte Carlo (SMC), Modelo de Regressão Linear Múltipla (MRLM) e Modelo de Árvore de Regressão (MAR). O segundo objetivo específico é obtido através da coleção de registros de risco a partir da experiência de um coordenador de projetos de pequeno e médio portes, com o auxílio do conhecimento presente na base de dados do PERIL e no livro do pesquisador Kendrick \cite{kendrick2003identifying}.Por fim, para se alcançar o terceiro objetivo específico as seguintes redes neurais são analisadas: \textit{Multilayer Perceptron} (MLP), \textit{Support Vector Machine} (SVM), \textit{Radial Basis Function} (RBF). Além disso, um sistema \textit{Neuro-fuzzy} (NFS) também está incluído nesse estudo.

Em resumo, a metodologia adotada nesse estudo está associada a realização de experimentos estatísticos para avaliação de erros de previsão dos impactos dos riscos oriundos da base de dados do PERIL \cite{kendrick2003identifying}, um \textit{framework} para identificar riscos no gerenciamento de projetos de software. As sete técnicas selecionadas estimarão o resultado dos impactos de risco. O Erro Médio Absoluto (EMA) será calculado trinta vezes para cada abordagem, e, então, um teste de hipótese, se necessário, será utilizado para afirmar qual desses métodos é o mais preciso e o que melhor se ajusta as particularidades desses dados. Mais detalhes sobre essa metodologia é apresentado no Capítulo \ref{cap:experiments}.

É concluído que uma variação da MLP chamada de \textit{MLPReg}, foi a abordagem mais eficiente para estimação dos impactos de riscos. Além disso, observou-se que todas as alternativas de redes neurais artificiais são melhores que os modelos de regressão linear e simulação de Monte Carlo. Notou-se que a utilização da simulação de MonteCarlo não se mostrou um método adequado para a análise de riscos, em comparação com os métodos associados as Redes Neurais Artificiais, conforme os experimentos estatísticos realizados.

O restante da dissertação está organizado nos seguintes capítulos: Capítulo \ref{cap:background} aborda o gerenciamento de risco de projetos, os conceitos de análise de risco qualitativa e quantitativa, a simulação de Monte Carlo, os modelos de regressão linear e as definições de redes neurais artificiais e suas características. O Capítulo \ref{cap:methodology} descreve o banco de dados do PERIL, os métodos de pré processamento de dados para preparação dos dados para esse estudo e também define os experimentos a serem realizados. O Capítulo \ref{cap:experiments} descreve cada um dos experimentos. O Capítulo \ref{cap:casestudy} apresenta um banco de dados de registros de risco conforme mencionado anteriormente. Finalmente, o Capítulo \ref{cap:conclusion} apresenta as conclusões e as sugestões de trabalhos futuros.

O trabalho descrito nessa dissertação teve seus resultados publicados nos seguintes artigos:

* C. H. M. S. Timoteo, M. J. S. Valença, S. M. M. Fernandes, "\textit{Evaluating Artificial Neural Networks and Traditional Approaches for Risk Analysis in Software Project Management - A case study with PERIL dataset}", ICEIS 2014: \textit{16th International Conference on Enterprise Information Systems}, Abril, 2014.