Capítulo 1 - Introducão

How risky are software projects? Several studies about effectiveness of software cost, scope, schedule estimation techniques; surveys from software professionals in industry; and analysis of project portfolios have been done to answer this question \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 na parede parece indicar. Projetos de alta tecnologia são particularmente arriscados, por uma série de motivos. Primeiro, projetos técnicos são muito variados. Esses projetos tem aspectos únicos e objetivos que diferem significativamente dos trabalhos anteriores, e o ambiente em projetos técnicos evoluem rapidamente. Além disso, projetos técnicos são frequentemente "enxutos", desafiados a trabalhar com financiamento, pessoal e equipamentos inadequados. Para piorar a situação, há uma falsa expectativa generalizada de que por mais rápido que tenha sido o último projeto, o próximo deve ser ainda mais rápido \cite{kendrick2003identifying}.

Projetos que tiveram sucesso geralmente conseguem isso porque seus líderes fazem bem duas coisas. Primeiro, eles reconhecem que alguns dos trabalhos em qualquer projeto, mesmo um projeto de alta tecnologia, não são novos. Para esse trabalho, as notas, registros e lições aprendidas em projetos anteriores podem ser um roteiro para identificar, e em muitos casos evitar muitos problemas potenciais. Segundo, eles planejam o trabalho do projeto a fundo, especialmente as partes que exigem inovação, para compreender os desafios futuros e antecipar muitos dos riscos \cite{kendrick2003identifying}.

Alguns benefícios da boa gestão de riscos de projetos de software são: a redução de custos incorridos com mudanças no software; o desenvolvimento de um plano de respostas a eventos inesperados, ou seja, um plano de contingência de riscos; a previsão da probabilidade da ocorrência de eventos indesejados; o 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 alcançaram sucesso - foram entregues no prazo, no orçamento e com os requisitos prometidos -, 44\% dos projetos foram desafiados; 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 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 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 entregam seus benefícios prometidos. Além disso, o autor mostra evidências de que isso é um assunto global, impactando organizações do setor privado e público \cite{KPMG2005}.

Como prever possíveis eventos a curto, médio e longo prazos? Ao analisar os riscos e incertezas, os gerentes de projeto muitas vezes confiam na intuição, em vez de lógica e análise detalhada. No entanto, o pensamento intuitivo é frequentemente alvo de ilusões, que causam erros mentais previsíveis e decisões eventualmente 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 deveriam 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 realização da mitigação \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.

Já que é uma área de pesquisa que vem crescendo, novas e melhores metodologias para identificar, medir e controlar itens de risco de software precisam ser desenvolvidas. Keshlaf e Riddle \cite{KESHLAFRIDDLE2010} concluem que 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}. Kwak e Ibbs \cite{kwak2000calculating} identificaram o gerenciamento de risco como 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 com incerteza como trabalho e custo extras.

Barry Boehm \cite{BOEHM1991} definiu risco como a possibildiade 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 foi visto refletindo 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, que 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 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 e conhecimento de projetos anteriores similares e de outra fonte de informação \cite{PMBOK2008}.

Risco é um conceito que muitos consideram difícil de ser compreendido: uma medida da probabilidade e da severidade de efeitos adversos \cite{Haimes2009}. Uma limitação dessa definição é a dificuldade prática em estimar a probabilidade e o impacto de diversos fatores de risco, especialmente em projetos de software. As probabilidades somente podem ser definidas com significância 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. Uma abordagem clássica na teoria da decisão \cite{bannerman2008risk} define risco como uma variação numa distribuição de probabilidade de resultados, não um resultado provável.

Outra limitação dessa definição é que ela abrange somente ameaças conhecidas ou previsíveis, oferecendo opções limitadas para gerenciar ameaças não percebidas, como também, não reconhece ameaças imprevisíveis. Essa é uma consequência da definição de risco em termos de probabilidade e impacto; já que, para avaliar a probabilidade e o impacto é necessário ser capaz de prever uma eventualidade.

Além disso, existe outra questão: 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 é observar o risco de um modo mais geral, em termos da incerteza, e avaliá-lo qualitativamente.

Haimes \cite{Haimes2009} considera duas premissas na pesquisa de análise de riscos, que também serão consideradas neste estudo. Uma, 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, por sua vez, afirma que uma das tarefas mais difícil em análise de sistemas é saber como modelá-lo. Portanto, novas propostas para a análise quantitativa e modelagem de sistemas sob o ponto de vista de seus riscos contribuirão 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. 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 realizam todos os processos de planejamento e em apenas 35% dos projetos o gerenciamento de riscos é realizado de acordo com uma metodologia formal, estruturada por políticas, procedimentos e formulários. Além disso, 46% dos gerentes realizam atividades de gerenciamento em tempo parcial.

A necessidade de gerenciar riscos (eventos indesejados) cresceu exponencialmente com a complexidade dos sistemas. Gerenciar esses eventos nesses sistemas complexos torna difícil identificar e estimar a ocorrência de eventos indesejados 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 ferramentas para suplementar o conhecimento individual, julgamento e experiência. Essas características humanas são muitas vezes suficientes para enfrentar riscos de menor complexidade e isolados. Por exemplo, uma parte dos problemas mais sérios encontrados na aquisição de um sistema são os resultados de riscos que são ignorados, devido a sua baixa probabilidade, até que eles já tenham criado 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, são inseridas no modelo. Comumente, o usuário deve estar preparado para realizar os ajustes necessários se os resultados que são gerados parecem fora de rumo. Além disso, 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 mais fácil, do ponto de vista dos usuários. Esse trabalho considera redes neurais artificial uma alternativa 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 objetivo principal dessa dissertação é analisar qual é uma abordagem mais eficiente para a análise de riscos em projetos de software: Simulação de Monte Carlo (SMC), Modelos de Regressão Linear (MRL's) ou as alternativas de Redes Neurais Artificiais (RNA's) - especificamente, Perceptron de Múltiplas Camadas (MLP), Máquina de Vetor de Suporte(SVM) e Redes de Função de Base Radial (RBF) - para melhorar a precisão e diminuir a estimativa de impacto de riscos propensa ao erro.

# Specific Objectives

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

Para alcançar o primeiro 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 foi incluído nesse estudo. O segundo 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). Por fim, o terceiro objetivo específico é obtido através da coleção de registros de risco a partir da experiência de um coordenador de projetos de pequeno e médio porte, com o auxílio do conhecimento presente na base de dados do PERIL e no livro de Kendrick \cite{kendrick2003identifying}.

Em resumo, a metodologia adotada nesse estudo é fazer experimentos estatísticos para avaliar os 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 de impactos de risco. Erro Médio Absoluto (EMA) será calculado trinta vezes para cada abordagem, e, então, um teste de hipótese pode ser necessário para afirmar qual deles é um método mais preciso que 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 vencedora para estimar o impactos de riscos. Além disso, observou-se que todas as alternativas de redes neurais artificiais são melhor que os modelos de regressão linear e simulação de monte carlo. Portanto, não foi descoberto qualquer motivo para atribuir simulação de monte carlo como um método recomendado para a análise de riscos, de acordo com os experimentos estatísticos conduzidos.

O restante da dissertação está organizado nos seguintes capítulos: Capítulo \ref{cap:background} aborda gerenciamento de risco de projetos, conceitos de análise de risco qualitativa e quantitativa, simulação de monte carlo, modelos de regressão linear e definições de redes neurais artificiais e suas características. O Capítulo \ref{cap:methodology} descreve o banco de dados do PERIL, métodos de pré processamento de dados para preparar os dados para esse estudo e define os experimentos a ser realizados. O Capítulo \ref{cap:experiments} descreve cada experimento. O Capítulo \ref{cap:casestudy} apresenta um banco de dados de registros de risco 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.