



## Programa de Pós-Graduação em Engenharia da Computação



### Uma Abordagem para Análise de Risco no Gerenciamento de Projetos de Software

Mestrado Acadêmico  
Engenharia da Computação

**\*\* 2º Estudo \*\***

Carlos Henrique Maciel Sobral Timóteo<sup>1</sup>  
Sérgio Murilo Maciel Fernandes<sup>2</sup>  
Mêuser Jorge Silva Valença<sup>3</sup>

---

<sup>1</sup>Email: chmst@ecomp.poli.br

<sup>2</sup>Email: smurilo@ecomp.poli.br

<sup>3</sup>Email: mjsv@ecomp.poli.br

# 1 Descrição do Problema

Risco é um conceito que muitos consideram difícil de ser compreendido: uma medida da probabilidade e da severidade de efeitos adversos [Hai09]. 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 [Ban08] 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 [Ban08].

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 [Ban08], 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 [Hai09] 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 [Ken03a] um *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.

Os processos realizados na fase de execução de projetos de software são atividades de alto risco, ocasionando resultados com desempenho variável. Uma pesquisa na indústria [Ban08] sugere que somente cerca de um quarto dos projetos de software alcançam sucesso - isto é, eles finalizam dentro do cronograma, no orçamento e como especificado - e bilhões de dólares são perdidos anualmente através de falhas de projeto ou projetos que não entregam os benefícios prometidos. Evidências sugerem que isso é um problema global [KPM05], impactando organizações do setor público e privado.

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* [Ins09b], em 20% dos projetos seus gerentes não realizam todos os processos de planejamento; 46% dos gerentes realizam atividades de gerenciamento em tempo parcial; apenas 35% dos projetos o gerenciamento de riscos é realizado de acordo com uma metodologia formal, estruturada por políticas, procedimentos e formulários.

## 2 Objetivos

### 2.1 Questões da Pesquisa

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

Como analisar quantitativamente os risco 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?

### 2.2 Objetivo Geral

O objetivo geral deste projeto de mestrado está associado à pesquisa e desenvolvimento de uma nova abordagem para a análise de risco no gerenciamento de projetos de software, baseado em técnicas de computação inteligente e em formalismos matemáticos para a modelagem de sistemas dinâmicos.

### 2.3 Objetivos Específicos

- Desenvolver uma técnica baseada em Redes de Petri para a previsão da probabilidade de riscos em gerenciamento de projetos de software
- Desenvolver uma técnica para a previsão do impacto dos riscos baseada em computação inteligente para o gerenciamento de riscos de projetos de software
- Desenvolver um método para o suporte a tomada de decisão para a definição de estratégias de mitigação de riscos e seu gerenciamento em projetos de

software

### 3 Motivação

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.

Em 2009, o CHAOS Report [Gro09] 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 [Isl09].

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 [Vir09].

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 [Kes10] concluem que mesmo que existam muitas abordagens ainda há uma grande lacuna com relação ao que é praticado pelas indústrias de software. Além disso, vale a pena perceber que não existe uma solução única para o conduzir o gerenciamento de riscos adequadamente [Vir09].

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 [Hig96] [Ins08].

### 4 Estado da Arte

A Gerência de Riscos de Projetos foi formalmente apresentada à comunidade de Engenharia de Software através da proposta do modelo Espiral de Barry Boehm [Boe91]. Seguindo-se a esse evento muitas abordagens e modelos foram apresen-

tados, destacando-se o programa do SEI (*Software Engineering Institute*) [Hig96] e, atualmente, os mais conhecidos modelo CMMI (*Capability Maturity Model Integration*) [Ins07] e o Guia PMBOK (*Project Management Book of Knowledge*) [Ins08].

O Gerenciamento de riscos de projetos é um dos principais tópicos de interesse dos pesquisadores e trabalhadores da área de gerenciamento de projetos. Tanto que um *survey* realizado por Williams em 1995 [Wil95] inclui 241 referências.

## 4.1 Gerenciamento de Riscos de Projetos

O Gerenciamento de Risco envolve processos de planejamento, identificação, avaliação e priorização dos mesmos. De acordo com o PMBOK [Ins08], riscos podem ser definidos como um evento incerto ou condição que, se ocorrer, tem um efeito em pelo menos um dos objetivos do projeto. Ele pode ser considerado tanto como ameaça (impacto negativo) ou oportunidade (impacto positivo). Numa definição melhor, gerenciamento de riscos pode ser definido como o processo de analisar a exposição ao risco e determinar como melhor lidar com essa exposição. Um objetivo dessa área não é somente a identificação, mas também desenvolver uma abordagem robusta para gerenciar proativamente o impacto no projeto [Osu12].

Para o PMBOK [Ins08], os processos de gerenciamento de risco do projeto são direcionados para aumentar a probabilidade e o impacto de eventos que são esperados para afetar positivamente o projeto, bem como para diminuir o probabilidade e o impacto dos eventos que são esperados para negativamente afetar o projeto ou a realização de seus objetivos.

O Software Engineering Institute(SEI) desenvolveu uma metodologia de gerenciamento de risco chamada *Software Risk Evaluation*(SRE) que é especificamente voltada para projetos na indústria de software. O paradigma SRE é composto por seis elementos: cinco módulos (identificar, analisar, planejar, acompanhar, controlar) e um elemento central (comunicação), que é a chave para a efetiva gestão de riscos [Hig96] [WPB99].

Boehm [Boe91], Chapman [CW96], Fairley [Fai94], Bandyopadhyay et al. [BMM99] apresentaram métodos e modelos diferentes de gerenciamento de riscos de projetos de software. No entanto, há elementos em comum em todas as abordagens anteriores: a identificação, a avaliação da probabilidade e impacto, e o planejamento a respostas para manusear esses riscos se eles ocorrerem [HS11].

### 4.1.1 Análise Qualitativa de Riscos em Projetos

O processo de análise qualitativa avalia as características dos riscos de projetos identificados individualmente e prioriza-os baseado nas requisições estabelecidas para o projeto. Em outras palavras, a análise qualitativa avalia a probabilidade de cada evento ocorrer e o efeito individual de cada um deles nos objetivos do projeto. Como tal processo não aborda diretamente o risco global para os objetivos do projeto, que resultam do efeito combinado dos eventos individuais e suas

interações entre si, então isso pode ser obtido através do uso de técnicas de análise quantitativa [Ins09a].

#### 4.1.2 Análise Quantitativa de Riscos em Projetos

A análise quantitativa de riscos envolve o processo de mapear eventos de riscos relacionados a atividades e executar a simulação de Monte Carlo considerando a probabilidade de ocorrência.

O objetivo da análise quantitativa de riscos é criar um "perfil do risco" do projeto. Para tanto, são necessárias as seguintes informações: a chance de o projeto ser finalizado dentro de um certo período de tempo ou orçamento; a taxa de sucesso de projetos; as estimativas pior, média e melhor de duração e outros parâmetros do projeto [Ins08].

As ferramentas utilizadas na análise quantitativa são:

1. Análise de Sensibilidade: determina qual incerteza tem o maior potencial de impacto.
2. Simulação de Monte Carlo: método matemático utilizado para aproximar a função de distribuição de probabilidade de resultados potenciais (duração do projeto, custo, taxas de sucesso, e outros parâmetros baseado em entradas probabilísticas). Nesse procedimento amostras aleatórias são criadas com base na amostra apresentada, de modo que a função de distribuição de probabilidade alcance o nível de confiança desejado, comumente 95% ou 99%.
3. Análise de Árvore de Falha: tipo de análise que determina qual decisão é a melhor. Por exemplo, para o gerenciamento do custo do projeto, a árvore de decisão suporta o cálculo do Valor Monetário Agregado (*Estimated Monetary Value*) que determina qual decisão é menos onerosa.

Alguns trabalhos propõem novas ferramentas de análise quantitativa de riscos. Entre eles, Virine [Vir09] apresenta a metodologia da Cadeia de Eventos. Nesse trabalho, as atividades de um projeto não são um procedimento uniforme e contínuo, essas tarefas são afetadas por eventos externos, que transformam as atividades de um evento para outro. O momento em que os eventos externos ocorrem são probabilísticos e podem ser definidos utilizando uma distribuição estatística. Além disso, eventos podem causar outros eventos, criando, portanto, a Cadeia de Eventos. A análise dessas combinações é realizada através da simulação de Monte Carlo.

### 4.2 Técnicas Convencionais de Análise de Risco

#### 4.2.1 Simulação de Monte Carlo

A invenção desse método, especialmente o uso de computadores para fazer os cálculos, foi creditado a Stanislaw Ulam, um matemático trabalhando no Projeto

Americano de Manhattan durante a Segunda Guerra Mundial. O seu trabalho com Jon von Neuman e Nicholas Metropolis transformou a amostragem estatística de uma curiosidade matemática para uma metodologia formal aplicável a uma grande variedade de problemas [KI07].

Simulação de Monte Carlo é uma abordagem detalhada de simulação através de computação intensiva para determinar a probabilidade de resultados possíveis de um objetivo do projeto; por exemplo, a data de conclusão ou o custo total. As entradas para o procedimento são obtidas aleatoriamente a partir de intervalos específicos com funções de distribuição de probabilidade para as durações das atividades do cronograma ou itens da linha de custo. Esses valores de entrada diferentes são usados para construir um histograma de possíveis resultados do projeto e sua probabilidade relativa, como também, a probabilidade cumulativa para calcular as reservas de contingência desejadas de tempo ou custo. Resultados adicionais incluem a importância relativa de cada entrada na determinação do custo geral do projeto e cronograma. Um exemplo de resultados de estimativa de riscos de cronograma e custo são apresentados na Figura 1 [Ins09a].

Esse método tem sido utilizado em projetos de construção para entender melhor certos riscos do projeto. Por exemplo, o ruído e seus efeitos negativos sobre a comunidade no entorno de uma obra é um risco em muitos projetos de construção urbana. Nesse caso, o modelo de simulação de Monte Carlo permite aos construtores prever e mitigar a ocorrência e o impacto do ruído da construção [KI07] [Ins09a].

Na Figura 1, observa-se a previsão de finalização para um cronograma de um projeto. No eixo horizontal, encontram-se as possíveis datas de finalização do cronograma, a frequência de ocorrência dessas datas na amostragem são representada pelas barras verticais, a frequência cumulativa é representada pela curva escura e pelos respectivos valores na borda direita do gráfico. A partir da frequência relativa, é possível prever a probabilidade do projeto finalizar até determinada data, por exemplo, esse projeto tem 90% de chance de finalizar até 08 de Maio de 2009.



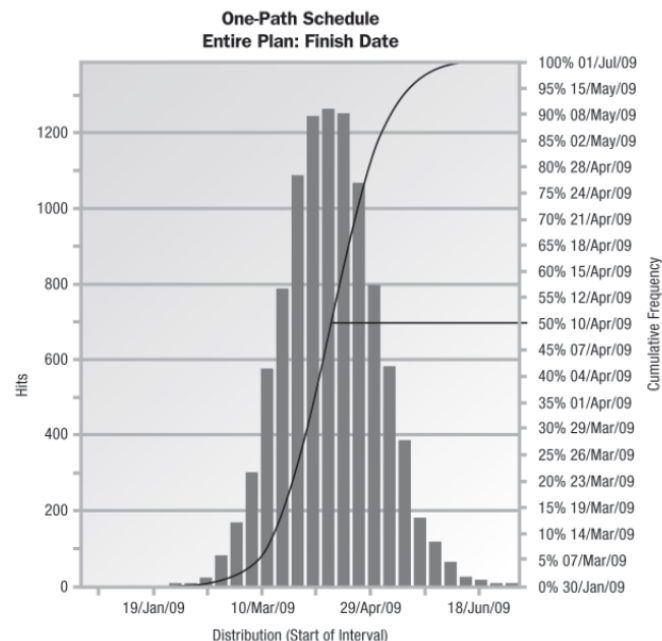


Figura 1: Exemplo de Resultado da Simulação Monte Carlo

#### 4.2.2 Análise de Árvore de Falha ou Análise de Modos de Falha e Efeitos

Análise de Árvore de Falha, do inglês *Fault Tree Analysis* (FTA), é um método importante para analisar e estimar a confiabilidade e disponibilidade de um sistema. No processo de concepção do sistema, depois de analisar vários fatores de fracasso (como *hardware*, *software*, ambiente, fatores individuais), podemos construir o diagrama lógico (isto é, a árvore de falha), sequencialmente, identificando todas as causas para prever a probabilidade de falha do sistema e tomar medidas corretivas para melhorar a confiabilidade e segurança [YZ<sup>+</sup>09].

No processo de construção da árvore de falha, um efeito indesejado é tomado como alvo na análise lógica, chamado de "evento de topo"; e, em seguida, todas as possíveis causas diretas desse efeito indesejado são descobertas, rotulados de "eventos do meio"; sequencialmente, todas as possíveis causas diretas desses "eventos do meio", chamados de "eventos de fundo". O diagrama lógico em árvore, que normalmente é escrito usando símbolos de portas lógicas e eventos de topo, do meio e de fundo, é chamado de Árvore de Falha [YZ<sup>+</sup>09].

De modo geral, a construção da Árvore de Falha é dividida em quatro passos:

1. Selecionar e obter o evento de topo. O evento de topo é o evento indesejado, ou o evento de falha da análise lógica da linha destacada.
2. Analisar o evento de topo. Descobrir todas as causas diretas, necessárias e suficientes do evento de topo. Tome o evento de topo como o evento de saída e todos os motivos diretos como eventos de entrada, e interligue esses



eventos usando portas lógicas apropriadas de acordo com as suas relações lógicas atuais.

3. Analisar todos os eventos de entrada diretamente ligado ao evento de topo. Tome-o como evento de saída do próximo nível se o evento pode ser decomposto adiante, como realizado no passo 2.
4. Repetir os passos acima, decompõe-os passo a passo até que todos os eventos de entrada não possam ser desmontados ou não precisam ser quebrados, isso é uma árvore de falha completamente invertida, como mostrado na Figura 2 [YZ<sup>+</sup>09]:

Na Figura 2, observa-se uma árvore de falha com um "evento de topo", três "eventos do meio" (A1, A2 e A3), "seis eventos" de fundo (B1 a B6), uma porta lógica "E" (T1) e três portas lógicas "OU" (T2, T3 e T4). Essa árvore de falha representa o seguinte conhecimento: O evento de topo ocorre, se os eventos A1, A2 e A3 ocorrerem. O evento A1 ocorre, se os eventos B1 ou B2 ocorrerem. O evento A2 ocorre, se os eventos B3 ou B4 ocorrerem. O evento A3 ocorre, se os eventos B5 ou B6 ocorrerem.

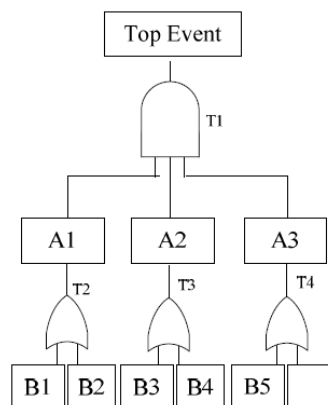


Figura 2: Exemplo de Árvore de Falha

## 4.3 Técnicas Estatísticas e de Computação Inteligente para Análise de Risco

### 4.3.1 Rede Bayesiana

Rede Bayesiana pertence à família de modelos gráficos probabilísticos. Uma rede bayesiana  $B$  é um grafo acíclico direto (GAD) que representa uma distribuição de probabilidade combinada sobre um conjunto de variáveis aleatórias  $V$ , no qual os nós representam variáveis e as arestas expressam as dependências entre variáveis [Jen96]. Logo, a rede pode ser definida por um par  $B = \langle G, \Theta \rangle$ , em que  $G$  é o GAD cujos nós  $X_1, X_2, \dots, X_n$  representam variáveis aleatórias, e cujas arestas representam as dependências diretas entre variáveis. Baseado em premissas independentes, cada nó  $X_i$  no grafo  $G$  é dependente de seus pais em  $G$ . Logo,

o grafo  $G$  leva em consideração a Suposição Markoviana [LLWS12]. A Suposição Markoviana para uma rede bayesiana estabelece que qualquer nó nessa rede é condicionalmente independente de seus não descendentes, dado que existe um parentesco.  $\Theta$  representa o conjunto de parâmetros da rede bayesiana, que contém parâmetros  $\theta_{X_i|\pi} = P_B(X_i|\pi_i)$  para cada valor  $x_i$  do nó  $X_i$  condicionando a  $\pi_i$ , onde  $\pi_i = \text{parent}(X_i)$  [Nea93].

Essa técnica tem sido usada como uma ferramenta importante para explorar diferentes informações, incluindo informações determinísticas ou probabilísticas de relações complexas entre variáveis. Elas constituem um *framework* de pesquisa conveniente que é fácil de usar pelos especialistas, principalmente na área de confiabilidade. Em particular, interesses de pesquisa significantes existem no uso de modelos de redes bayesianas para a análise de confiabilidade de sistemas e risco em engenharia de software, como também previsão de qualidade [LLWS12] [FNM<sup>+</sup>07].

Através do aprendizado da rede bayesiana, todas as probabilidades condicionais entre cada dois riscos adjacentes podem ser calculadas. Uma rede bayesiana, pode não somente capturar o relacionamento entre convicções incertas em proposições, mas também pode explicar o valor de verdade de uma ou mais proposições. Algoritmos de inferência da rede bayesiana podem ser usados para encontrar causalidades ou crenças atualizadas para cada uma das variáveis, e relações atualizadas entre variáveis [JN07]. Uma rede bayesiana codifica a distribuição de probabilidade de um conjunto de variáveis aleatórias através da especificação de um conjunto de suposições de independência ou de dependência condicional junto com um conjunto de relacionamentos entre essas variáveis e suas probabilidades combinadas relacionadas. Ao invés de resultados determinísticos tipicamente gerados pelo aprendizado, redes bayesianas podem produzir um conjunto de resultados probabilísticos em cada processo. Eles são mais adequados para a identificação de riscos em diferentes fases do processo de software em situações reais [LLWS12].

A Figura 3 ilustra um exemplo de uma rede bayesiana simples para um projeto com três tipos de fatores de risco e dois subprocessos. Identificados pelo índice  $R_{ij}$ , onde  $i$  representa os tipos de fatores de risco e  $j$  representa o número de subprocessos. Ele inclui três nós pais e três nós filhos.

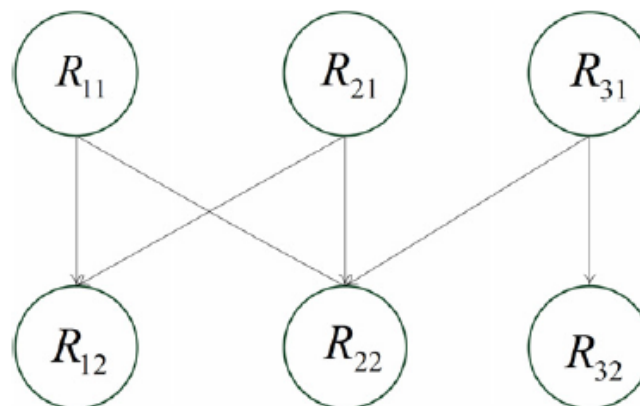


Figura 3: Exemplo simples de Rede Bayesiana

### 4.3.2 Redes de Petri Estocásticas

A Rede de Petri é um formalismo matemático que permite representar sistemas dinâmicos em que há relações de causa e efeito complexas, concorrência e sincronização. Elas foram criadas em 1962 por Carl Adam Petri [Pet62]. A rede de Petri é um grafo direcionado em que há dois tipos de nós: lugares, representados por círculos, e transições, representadas por barras ou retângulos. Os lugares podem conter tokens, que são representados por círculos pretos dentro do lugar.

Em [Jia04], uma técnica baseada em redes de petri foi utilizada para a análise qualitativa e quantitativa de riscos em desenvolvimento de software. O artigo apresenta um *framework* de coordenação de métricas de software e modelo de processos de software para gerenciar os riscos de projetos. Além disso, [Car12] utilizou redes de Petri para identificar, controlar, e gerenciar riscos de desenvolvimento de programas.

A Figura 4 apresenta um exemplo simples de rede de Petri, em que é modelado o ciclo dia e noite. Quando um *token* é colocado no lugar Dia, significa que o sistema está no estado Dia. Isto habilita a transição Anotar a ocorrer. Quando esta transição dispara, o *token* é removido do lugar Dia e um *token* é colocado no lugar Noite, indicando que o sistema está neste estado. Isto, por sua vez, desabilita a transição Anotar e habilita a transição Amanhecer. De forma análoga, o disparo da transição Amanhecer faz com que o *token* seja removido do lugar Noite e um *token* seja colocado no lugar Dia [HTT00] [Fer07].

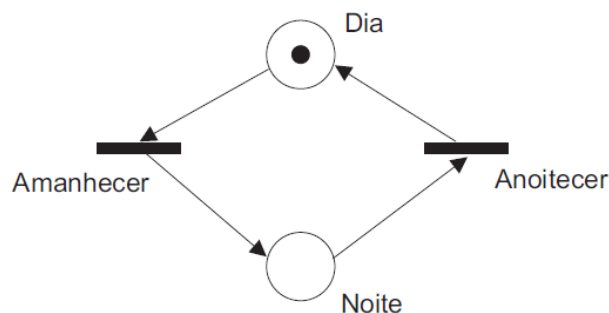


Figura 4: Exemplo de Rede de Petri

As Redes de Petri Estocásticas Generalizadas (*Generalized Stochastic Petri Nets*) [AM95], permitem a utilização de transições temporizadas estocásticas em conjunto com transições não-temporizadas (chamadas imediatas). Esta característica facilita a modelagem de transições lógicas, que não estão associadas a tempo. Transições temporizadas apenas são habilitadas quando não há mais nenhuma transição imediata habilitada. Marções em que transições imediatas estão habilitadas são chamadas de não-tangíveis (*vanishing*). Marções em que transições temporizadas estão habilitadas são chamadas de tangíveis (*tangible*).

### 4.3.3 Redes Neurais Artificiais

As Redes Neurais Artificiais (RNA) são modelos que procuram simular o comportamento e funcionamento do cérebro humano. Assim como existem os neurônios biológicos, componentes essenciais para o processamento das informações do cérebro, a RNA é formada por unidades que objetivam realizar as mesmas funções do neurônio biológico. Esses componentes são denominados neurônios artificiais e foram propostos em 1943 por Mc-Culloch e Pitts [MCC43].

### 4.3.4 *MultiLayer Perceptron*

A rede *MultiLayer Perceptron* (MLP) funciona como um *perceptron* de muitas camadas, ou seja, ela possui pelo menos uma camada de neurônios intermediária ou escondida. A vantagem de ter camadas intermediárias é que a rede neural passa a resolver problemas que não são linearmente separáveis, possibilitando, assim, a aproximação de qualquer função contínua, com apenas uma camada, e qualquer função matemática, quando houver mais de uma camada [Hay07]. A Figura 5 ilustra a forma gráfica da MLP, apresentando as entradas, saídas e as camadas de entrada, intermediária e de saída.

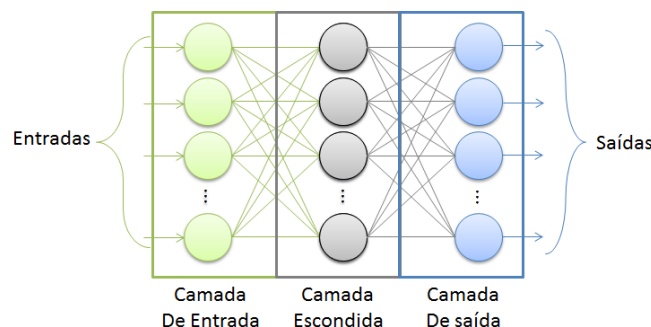


Figura 5: Representação gráfica da MLP com três camadas

O funcionamento da MLP é dividido em duas fases: *forward* e *backward*. Na fase *forward*, um neurônio de uma camada está ligado a todos os neurônios da camada seguinte. Os sinais da entrada são propagados de camada de entrada para a camada escondida e da camada escondida para a camada de saída; cada neurônio processa as entradas e apresenta uma saída. Nessa fase é possível calcular o erro entre a saída desejada para a rede e a saída apresentada pela MLP. Na fase *backward* o erro é retropropagado e os pesos são ajustados, utilizando o algoritmo de ajustes de pesos, inicialmente aleatórios, chamado *Backpropagation* [Val05].

Em [Hu07], algumas técnicas inteligentes são utilizadas para a análise do risco projetos de software. O artigo conclui que uma técnica híbrida de redes neurais e algoritmos genéticos obteve uma precisão de 85% para prever se o projeto obterá sucesso, será desafiado ou falhará totalmente.

#### 4.3.5 Máquina de Vetor de Suporte

A Máquina de Vetor de Suporte, do inglês *Support Vector Machine* (SVM), é uma técnica de aprendizado de máquina aplicável a problemas de reconhecimento de padrões nos quais se busca atingir alto potencial de generalização [Hay07] [Val05]. O objetivo da SVM é encontrar um hiperplano particular, denominado de hiperplano ótimo, que maximize a margem de separação, conforme pode ser visualizado na Figura 6.

Na Figura 6, observamos dois vetores de suporte capazes de separar linearmente as saídas no hiperplano em duas classes. A variável  $r$  é a distância algébrica desejada dos vetores de suporte para o hiperplano ótimo de separação das classes. Quanto maior essa distância, maior a capacidade de generalização da máquina de vetor de suporte.

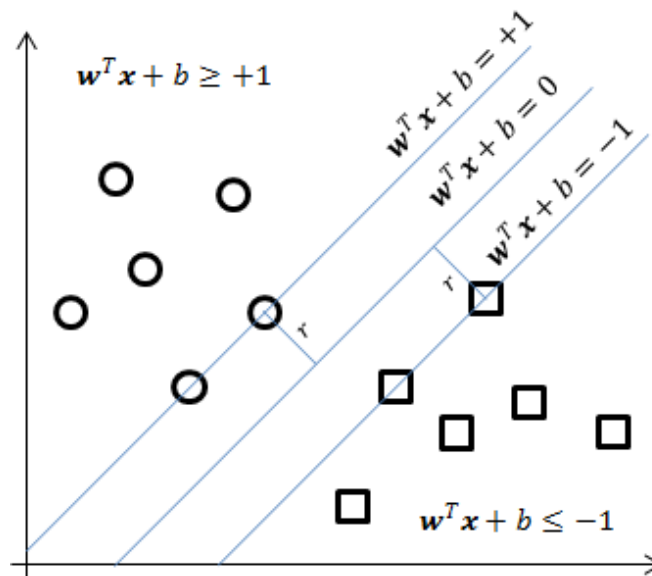


Figura 6: Vetores de Suporte e Hiperplano de Separação ótimo

#### 4.3.6 Lógica Fuzzy

A aplicação da Teoria de Conjuntos Fuzzy (TCF) para a análise de riscos parece apropriada, devido ao fato dessas análises serem altamente subjetivas e relacionadas a informações inexatas e vagas. Desde que TCF foi introduzida por Zadeh para lidar com problemas no qual imprecisão era presente, valores linguísticos têm sido amplamente usados para o raciocínio aproximado. Numerosos estudos de TCF na avaliação de risco tem surgido em diferentes áreas, e são sumarizados em [NW05]. TCF tem sido efetivo numa variedade de áreas porque ele pode lidar com informações inexatas, porém úteis.

#### 4.3.7 Fuzzy Weight Average

Uma operação comumente usada em análise de risco e decisão é a operação de média ponderada, que tem a seguinte forma [NW05]:

$$\bar{W} = \frac{\sum_{i=1}^n W_i \times R_i}{\sum_{i=1}^n W_i} \quad (1)$$

onde  $\bar{W}$  é a média ponderada dos valores,  $R_i$  é o valor de acordo com o critério  $i$ , e  $W_i$  é o peso atribuído ao critério  $i$ . Quando os termos  $R_i$  e  $W_i$  são representados por conjuntos fuzzy ou números fuzzy, a operação acima é referenciada como uma Média Ponderada Fuzzy (MPF).

Schmucker [Sch84] usou a MPF para propor um método numérico aproximado conhecido como o "Analisador de Riscos Fuzzy". Além disso, muitas aplicações seguem o procedimento de Schmucker, e isso é amplamente aplicado em análise de risco, particularmente com relação a projetos de construção [NW05].

#### 4.3.8 Conceitos de Confiabilidade

Se  $X$  é uma variável aleatória que indica o tempo de vida ou tempo até a falha de um componente ou sistema e  $X$  tem função de distribuição de probabilidade [GT90].

$$F_X(t) = P(X \leq t) \quad (2)$$

em seguida, a confiabilidade do componente ou sistema  $R_X(t)$  representa a probabilidade de que o sistema sobreviva até ao tempo  $t$ , isto é,

$$R_X(t) = P(X < t) = 1 - F_X(t) \quad (3)$$

Se  $R_X(t)$  é diferenciável, então a taxa de perigo ou taxa de falha do componente ou sistema é dado por

$$h(t) = -R'_X(t)/R_X(t) = \frac{\partial F_X/\partial t}{1 - F_X(t)} \quad (4)$$

a qual podemos interpretar vagamente como a taxa de falha condicional no próximo instante dada a sobrevivência até  $t$ . Neste caso

$$F_X(t) = 1 - e^{-\int_0^t h(x)dx} \quad (5)$$

e, portanto, uma taxa de falha constante implica uma distribuição de tempo de vida exponencial.

Demandas para maior precisão na estimativa de confiabilidade rapidamente forçou o desenvolvimento de modelos de sistema mais elaborados sem os pressupostos de independência. A segunda geração de ferramentas de estimativa, surgiu com base em métodos Markovianos, permitindo assim uma importante dependência de primeira ordem.

Um modelo de Markov de estado discreto e tempo contínuo é um conjunto de estados, juntamente com as taxas de transição não negativas entre esses estados. Deixe  $a_{ij}$ , indicar a taxa de transição do estado  $i$  para o estado  $j$ . O modelo de Markoviano representa então um sistema em evolução o tempo que muda de estado de acordo com as seguintes regras [GT90]:

1. O tempo de espera em cada estado  $i$  é exponencialmente distribuído com média  $1/\sum_{j \neq i} a_{ij}$
2. Na saída do estado  $i$ , o estado  $j$  é escolhido com probabilidade  $a_{ij}/\sum_{j \neq i} a_{ij}$

Esses modelos Markovianos são equivalentes a sistemas lineares de equações diferenciais ordinárias especiais. Se  $P_i(t)$  determina a probabilidade do sistema estar no estado  $i$  no tempo  $t$ ,  $P(t) = (P_1(t), \dots, P_n(t))$ , e  $A$  é uma matriz  $n \times n$  cujas entradas fora da diagonal são taxas de transição  $a_{ij}$  e entradas diagonais  $a_{ii} = -\sum_{j \neq i} a_{ij}$ , então o modelo markoviano é equivalente ao sistema linear homogêneo

$$P'(t) = P(t)A \quad (6)$$

Entre as ferramentas que utilizam um framework markoviano para construção e análise de modelos tolerante a falhas ou erros há a rede de Petri [GT90].

Por fim, Bonini e Caivano [BC13] utilizaram os conceitos de sobrevivência para desenvolver uma técnica de análise de risco de crédito. Essa abordagem mostrou-se bastante promissora no contexto de previsão da quitação de empréstimos.

## 5 Metodologia

Esse projeto de pesquisa é realizado baseado nas seguintes atividades:

1. Pesquisa Bibliográfica das Áreas de Conhecimento:
  - Gerenciamento de Projetos
  - Gerenciamento de Risco de Projetos
  - Análise de Risco de Projetos de Software
  - Probabilidade e Estatística
  - Confiabilidade de Sistemas
  - Métodos Estatísticos para Análise de Risco
  - Métodos Formais para Análise de Risco
  - Redes Neurais Artificiais
  - Inteligência de Enxames
2. Levantamento de Bases de Dados com informações de Riscos de Gerenciamento de Projetos de Software



3. Levantamento de Ferramentas necessárias para o Tratamento dos Dados contidos nas Bases
4. Definição de Métricas Quantitativas e os Parâmetros para Análise de Risco de Projetos de Software
5. Avaliação dos Resultados das Métricas
6. Definição do Grau de Exposição ao Risco de um Projeto
7. Suporte a Tomada de Decisões Estratégicas de Mitigação de Riscos

## 5.1 Base de Dados

### 5.1.1 Base de Dados PERIL

Um melhor gerenciamento de riscos começa com a identificação de problemas potenciais. A adoção das ferramentas disponíveis como: revisão das lições aprendidas, *brainstorming*, entrevistas, opinião especializada; são alternativas relativamente eficientes, porém muitas vezes apresentam alto custo. Uma proposta acessível, de baixo custo e extensível é utilizar a base de dados do PERIL [Ken03a].

Durante alguns anos, em *Workshops* de Gerenciamento de Riscos, foram realizadas entrevistas com centenas de líderes de projetos para conhecer problemas de projetos típicos já enfrentados, definindo o que havia ocorrido e o impacto disso no projeto. Esses dados foram agrupados no banco de dados *Project Experience Risk Information Library* (PERIL) [Ken03b].

Em projetos, os riscos encontrados podem ser classificados como "conhecidos", aqueles antecipados no planejamento, ou "desconhecidos", encontrados durante a execução do projeto. O objetivo da análise dessa base de dados é prover um *framework* para identificação de riscos de modo a aumentar o número de riscos conhecidos, e diminuir o número de riscos desconhecidos.

Algumas características dessa base de dados:

1. Os dados são descorrelacionados: eles representam uma pequena fração de dezenas de milhares de projetos realizados pelo gerente de projeto, por quem eles foram coletados;
2. PERIL apresenta viés (*bias*): a informação não foi coletada aleatoriamente.
3. PERIL representa somente os riscos mais significantes.

A maioria dos projetos teve duração entre 6 meses e 1 ano, e a equipe continha entre 10 e 25 pessoas. No total, 649 riscos foram identificados e categorizados nos seguintes tipos: escopo, cronograma e recurso. A categoria escopo é composta das seguintes subcategorias: mudança e defeito. A categoria cronograma é composta das subcategorias: dependência, estimativa e atraso. Já, recursos é composta das seguintes subcategorias: dinheiro, terceirização e pessoas.

Uma desvantagem dessa base de dados, é que ela somente contabiliza riscos que tiveram impacto negativo no projeto. As oportunidades não foram identificadas e maximizadas com esse estudo.

No entanto, um dos grandes benefícios é que o autor apresenta alguns riscos como *black swans* [Ken03a]: representando a idéia de riscos com amplo impacto, difíceis de prever e raros de ocorrer. Se o risco tiver impacto negativo, é conhecido como catástrofe, ao passo que, se tiver impacto positivo, é conhecido como recompensa.

### 5.1.2 Bases de Dados de Riscos de Software

Outras bases de dados utilizadas para previsão de custo de software e de defeito de software também serão consideradas nesse estudo. Nos manuais dessas bases de dados existe a definição indireta da métrica de risco a ser extraída. Essas bases de dados públicas estão disponíveis em <http://promise.site.uottawa.ca/SERepository/datasets-page.html>.

Durante o estudo as bases de dados que se adaptarem aos objetivos do estudo serão selecionadas.

## 5.2 Ferramentas

### 5.2.1 TimeNET

TimeNET (*Timed Net Evaluation Tool*) é um pacote de software para a modelagem e avaliação de redes de Petri estocásticas com tempos de disparo não exponencialmente distribuídos. TimeNET foi desenvolvido na Technische Universität Berlin em vários projectos de investigação. A interface gráfica é fornecida para a especificação do modelo e várias análises especializadas além de simulação para a avaliação do modelo automatizado. A implementação dos componentes de análise e simulação é baseada em resultados de pesquisas recentes [GKZH95].

### 5.2.2 SPNP

SPNP é um pacote de GSPN poderoso desenvolvido na Universidade de Duke. SPNP permite a modelagem de comportamentos de sistemas complexos. Construções avançadas estão disponíveis, tais como a marcação de multiplicidades de arco dependentes, permitindo as funções, vetores de lugares ou transições e sub-redes; além disso, o poder expressivo completo da linguagem de programação C está disponível para aumentar a flexibilidade da descrição da rede.

Solucionadores de estado estacionário e transiente sofisticados estão disponíveis. Finalmente, o utilizador não está limitada a um conjunto pré-definido de medidas: expressões detalhadas refletindo exatamente as medidas requeridas podem ser facilmente especificadas [CMT89].

### 5.2.3 Weka

O projeto WEKA tem como objetivo fornecer um conjunto abrangente de algoritmos de aprendizado de máquina e ferramentas para pré-processamento de dados para pesquisadores e profissionais afins. Ele permite ao usuário tentar comparar rapidamente diferentes métodos de aprendizado de máquina a partir de um conjunto de dados. Sua arquitetura modular e extensível permite que processos sofisticados de mineração de dados possam ser construído a partir da vasta coleção de algoritmos de aprendizagem de base e ferramentas fornecidas. Estender esse kit de ferramentas é fácil graças a existência de: uma API, plugins e a instalação que automatiza a integração de novos algoritmos de aprendizado com o WEKA.

Ele inclui algoritmos de regressão, classificação, agrupamento, mineração de regras de associação e seleção de atributos. A exploração preliminar dos dados é realizada através de visualização de dados e muitas ferramentas de pré-processamento [HFH<sup>+</sup>09].

## 5.3 Estudos de Caso

Alguns estudos de casos precisam ser definidos para avaliar qual das alternativas utilizadas apresentam melhores desempenhos.

O primeiro estudo de caso é comparar o desempenho da adoção de Redes de Petri, Árvores de Falha, Redes Bayesianas e Lógica Fuzzy. Nesse estudo de caso, o objetivo é verificar qual a melhor alternativa para realizar a análise qualitativa dos riscos do projeto.

O segundo estudo de caso é comparar o desempenho da adoção de Redes Neurais MLP, Máquinas de Vetor de Suporte, Simulação de Monte Carlo e um método estatístico convencional de Regressão Linear. Nesse caso, o objetivo é verificar qual alternativa oferece menor erro na estimativa do impacto dos riscos.

### 5.3.1 Atividades Realizadas

Durante um ano de pesquisa algumas atividades foram realizadas, as principais atividades são listadas e descritas abaixo:

1. Estudo do Guia de Boas Práticas do PMBOK 2008 [Ins08]: Foi realizada a inscrição num Curso Preparatório para obtenção do Certificado CAPM. As aulas eram semanais, contabilizando 4h/semana. Dois meses depois, após submissão ao exame, o Certificado de Gerenciamento de Projetos CAPM foi obtido.
2. Estudo do conceito de Risco e Perigo: Lendo sobre os conceitos de Risco, observou-se importância no estudo dos perigos e catástrofes. A teoria de Confiabilidade de Sistemas aborda os conceitos de tempo de sobrevivência, confiabilidade, disponibilidade, função perigo. Apresentando esses conceitos ao orientador foi observada uma relação desses conceitos com a definição

proposta por Boehm [Boe91]. A representação gráfica da função de sobrevivência é similar ao do risco. Essa relação inspira esse estudo e a publicação dessa relação, visto que não encontramos nenhum autor que tivesse observado esse fato.

3. Estudo dos conceitos de Probabilidade e Estatística: Observou-se que as técnicas convencionais para Análise Quantitativa de Riscos [Ins09a] utilizam técnicas estatísticas como diagrama PERT e CPM, Simulação de Monte Carlo, Árvore de Falha. Portanto, os conceitos de Probabilidade e Estatística básica foram adquiridos e aprofundou-se em Modelos de Box & Jenkins e Modelos de Markov, que é a base para Redes de Petri Estocásticas.
4. Levantamento de Base de Dados: Foi localizado no motor de busca do site do PMI.org (<http://www.pmi.org>) a base de dados PERIL publicada em alguns artigos. Em seguida, contactou-se o autor dos artigos por email e a base de dados foi solicitada.
5. Estudo de Redes Neurais Artificiais: Analisando o PERIL, observou-se que fosse interessante extrair uma previsão dos dados contidos nessa base de dados. Para realizar essa atividade, estudou-se Redes Neurais Artificiais e foi implementada uma rede neural MLP e um programa que realizava a previsão de alguns dados de testes. A partir desse estudo, algumas métricas e parâmetros de análise de riscos foram definidos. Por fim, foi realizado um experimento empírico para a determinação dos melhores parâmetros da MLP, no contexto da base de dados PERIL. Mais detalhes sobre os resultados, será descrito na seção de Resultados Preliminares.
6. Estudo de técnicas de Computação Inteligente: Após a implementação do programa para previsão do impacto de riscos a partir da MLP, surge uma necessidade conhecida da implementação de um algoritmo de otimização dos parâmetros da rede neural artificial. Nesse caso, observou-se que uma implementação do híbrida do PSO+MLP é suficiente para que o algoritmo de previsão se adapte aos dados que estão sendo tratados.
7. Revisão Bibliográfica: Diante dos resultados obtidos e do conhecimento obtido em Análise de Risco, foi necessário realizar uma revisão bibliográfica para identificar o estado da arte e validar o trabalho já realizado.

### 5.3.2 Resultados Preliminares

O método utilizado para realizar a estimativa do impacto de riscos num projeto de software a partir da base de dados do PERIL é o seguinte:

1. Obtenção da base de dados do PERIL
2. Pré-processamento dos dados: Binarização dos campos textuais e normalização gamma dos impactos, para todas as colunas da base de dados. O in-

intervalo de normalização é entre 0.15 e 0.85, conforme sugere Valença [Val05]. Além disso, os dados foram embaralhados aleatoriamente.

3. Rede Neural MLP: Os parâmetros da rede neural após otimização empírica foram:
  - $\alpha = 0.5; \beta = 0.1$
  - Quantidade de neurônios nas camadas: Entrada = 12; Escondida = 10; Saída = 1 (Impacto)
  - Aprendizado: Algoritmo *Back Propagation*
  - Condição de Parada: Erro Médio Absoluto (EMA) de Testes e Validação Cruzada convergirem.
4. Cálculo do EMA e Erro Percentual Médio Absoluto (EPMA) para os dados de Teste.
5. Validação Estatística dos Erros: 30 experimentos.

Para a definição dos parâmetros ótimos, foi realizado um estudo estatístico empírico com os seguintes passos:

1. Determinação das variáveis: número de neurônio da camada escondida,  $\alpha$  e  $\beta$
2. Execução de 30 experimentos
3. Análise Gráfica e/ou Teste Estatístico Wilcoxon para seleção dos melhores valores

O intervalo de valores aceitável para as variáveis número de neurônios da camada escondida ( $n$ ),  $\alpha$  e  $\beta$  são apresentados na Tabela ??.

Intervalo de Valores			
	$n$	$\alpha$	$\beta$
Mínimo	10	0.5	0.1
Máximo	100	0.9	0.5

Os resultados indicam que a configuração que obtinha os menores EMA para o PERIL são:  $n = 10$ ,  $\alpha = 0.5$  e  $\beta = 0.1$ . A Figura 7 mostram os resultados para as dez primeiras configurações avaliadas fixando a quantidade de neurônios, em que  $n = 10$ . Observa-se que os trinta experimentos para as dez primeiras configurações são apresentados por *boxplots*. Nesse caso, o *boxplot* mais a esquerda é o que apresenta os menores erros de previsão. Para essa configuração foi obtido um Erro Percentual Médio Absoluto de 15%.

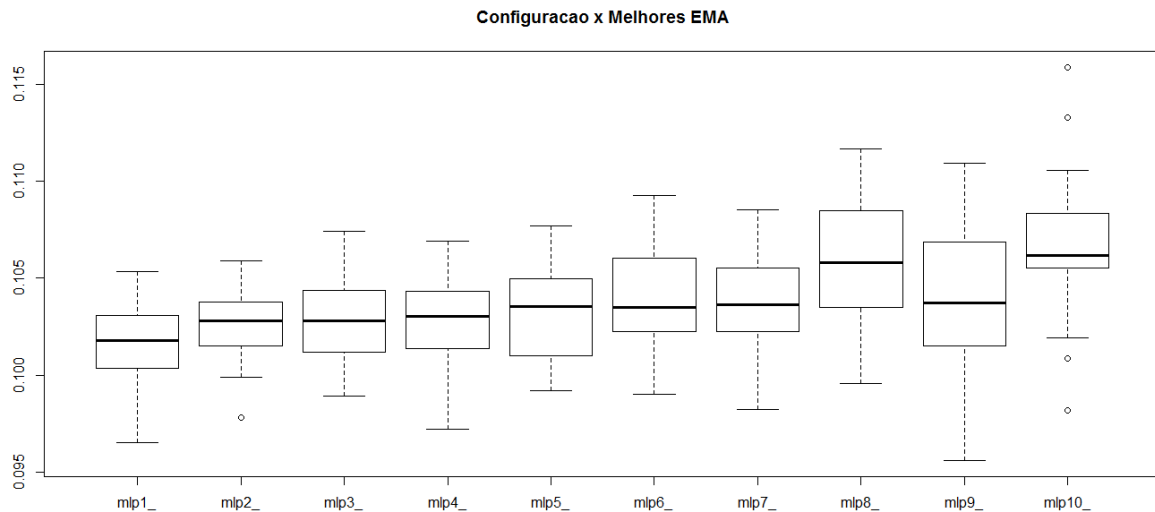


Figura 7: *Boxplots* das configurações do MLP

## Referências

- [AM95] Balbo G. Conte G. et al. Ajmone Marsan, M. Modelling with generalized stochastic petri nets. *Wiley series in parallel computing*, 1995.
- [Ban08] Paul L Bannerman. Risk and risk management in software projects: A reassessment. *Journal of Systems and Software*, 81(12):2118–2133, 2008.
- [BC13] Stephano Bonini and Giuliana Caivano. The survival analysis approach in Basel II credit risk management: modeling default rates in the loss given default parameter. *Journal of Credit Risk*, 9(1):101–118, 2013.
- [BMM99] Kakoli Bandyopadhyay, Peter P Mykytyn, and Kathleen Mykytyn. A framework for integrated risk management in information technology. *Management Decision*, 37(5):437–445, 1999.
- [Boe91] B. W. Boehm. Software risk management: principle and practices. *IEEE Software*, 8:32–41, 1991.
- [Car12] Tasinaffo P. M. Montini D. A. Fernandes D. D. da Cunha A. M. Dias L. A V. Cardoso, F. R. M. A formal control model for risks management within software projects. In *Proceedings of the 2012 Ninth International Conference on Information Technology - New Generations, ITNG '12*, pages 452–457, Washington, DC, USA, 2012. IEEE Computer Society.

- [CMT89] Gianfranco Ciardo, Jogesh Muppala, and Kishor Trivedi. Spnp: stochastic petri net package. In *Petri Nets and Performance Models, 1989. PNPM89., Proceedings of the Third International Workshop on*, pages 142–151. IEEE, 1989.
- [CW96] Chris Chapman and Stephen Ward. *Project risk management: processes, techniques and insights*. John Wiley, 1996.
- [Fai94] Richard Fairley. Risk management for software projects. *Software, IEEE*, 11(3):57–67, 1994.
- [Fer07] S. M. M. Fernandes. Avaliação de dependabilidade de sistemas com mecanismos tolerantes a falha: Desenvolvimento de um método híbrido baseado em edspn e diagrama de blocos. PhD Thesis, 2007.
- [FNM<sup>+</sup>07] Norman Fenton, Martin Neil, William Marsh, Peter Hearty, David Marquez, Paul Krause, and Rajat Mishra. Predicting software defects in varying development lifecycles using bayesian nets. *Information and Software Technology*, 49(1):32–43, 2007.
- [GKZH95] Reinhard German, Christian Kelling, Armin Zimmermann, and Günter Hommel. Timenet: a toolkit for evaluating non-markovian stochastic petri nets. *Performance Evaluation*, 24(1):69–87, 1995.
- [Gro09] The Standish Group. Chaos report. 2009.
- [GT90] Robert Geist and Kishor S. Trivedi. Reliability estimation of fault-tolerant systems: Tools and techniques. *Computer*, 23(7):52–61, 1990.
- [Hai09] Yacov Y(University of Virginia) Haimes. *Risk Modeling, Assesment and Management*. John Wiley & Sons, Inc., 3rd edition edition, 2009.
- [Hay07] S. Haykin. *Redes Neurais: Princípios e Práticas*. 2007.
- [HFH<sup>+</sup>09] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.
- [Hig96] Y. Y. Higuera, R. P. e Haimes. Software risk management, 1996.
- [HS11] Vered Holzmann and Israel Spiegler. Developing risk breakdown structure for information technology organizations. *International Journal of Project Management*, 29(5):537–546, 2011.
- [HTT00] Christophe Hirel, Bruno Tuffin, and Kishor S. Trivedi. Spnp: Stochastic petri nets. version 6.0. In Boudewijn R. Haverkort, Henrik C. Bohnenkamp, and Connie U. Smith, editors, *Computer Performance*



*Evaluation: Modelling Techniques and Tools, 11th International Conference, TOOLS 2000, Schaumburg, IL, USA, March 27-31, 2000, Proceedings*, volume 1786 of *Lecture Notes in Computer Science*, pages 354–357. Springer, 2000.

- [Hu07] Huang J. Chen J. Liu M. Xie K. Hu, Y. Software project risk management modeling with neural network and support vector machine approaches. *Third International Conference on Natural Computation (ICNC)*, 2007.
- [Ins07] SEI Software Engineering Institute. *CMMI - Capability Maturity Model Integration*. 2007.
- [Ins08] Project Management Institute. *A Guide to the Project Management Body of Knowledge*. 2008.
- [Ins09a] Project Management Institute. Practice standard for project risk management. 2009.
- [Ins09b] Project Management Institute. Relatório principal perspectiva geral estudo de benchmarking em gerenciamento de projetos. 2009.
- [Isl09] S. Islam. Software development risk management model - a goal driven approach. *Proc. Of ESEC FSE Doctoral Symposium 09 Amsterdam The Netherlands*, pages 5–8, 2009.
- [Jen96] Finn V Jensen. *An introduction to Bayesian networks*, volume 210. UCL press London, 1996.
- [Jia04] Chen Y. Jiang, G. Coordinate metrics and process model to manage software project risk. *International Engineering Management Conference*, 2004.
- [JN07] Finn Verner Jensen and Thomas Dyhre Nielsen. *Bayesian networks and decision graphs*. Springer, 2007.
- [Ken03a] T. Kendrick. *Identifying and Managing Project Risk: Essential Tools for Failure-Proofing your Project*. 2003.
- [Ken03b] T. Kendrick. Overcoming project risk: Lessons from the peril database. *Proceedings of PMI Global Congress*, 2003.
- [Kes10] Riddle S. Keshlaf, A. Risk management for web and distributed software development projects. *The Fifth International Conference on Internet Monitoring and Protection*, 2010.
- [KI07] Young Hoon Kwak and Lisa Ingall. Exploring monte carlo simulation applications for project management. *Risk Management*, 9(1):44–57, 2007.

- [KPM05] KPMG. Global it project management survey. 2005.
- [LLWS12] Jianping Li, Minglu Li, Dengsheng Wu, and Hao Song. An integrated risk measurement and optimization model for trustworthy software process management. *Information Sciences*, 191:47–60, 2012.
- [MCC43] Pitts W. MCCulloch, W. A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, pages 115–133, 1943.
- [Nea93] Radford M Neal. Probabilistic inference using markov chain monte carlo methods. 1993.
- [NW05] Eric WT Ngai and FKT Wat. Fuzzy decision support system for risk analysis in e-commerce development. *Decision support systems*, 40(2):235–255, 2005.
- [Osu12] A. Osundahunsi. Effective project risk management using the concept of risk velocity, agility and resiliency. *CAMERON International, Houston, TX, USA*, page 13, 2012.
- [Pet62] C. A. Petri. Kommunikation mut automaten. *PhD thesis, Schriften des IIM Nr. 2, Bonn*, 1962.
- [Sch84] Kurt J Schmucker. *Fuzzy sets, natural language computations, and risk analysis*. Computer Science Press Rockville, MD, 1984.
- [Val05] M. J. S. Valença. Aplicando redes neurais: um guia completo. *Livro Rápido, Olinda-PE*, 2005.
- [Vir09] L. Virine. Project risk analysis: How to make better choices in the uncertain times. *Proceedings of PMI Global Congress*, 2009.
- [Wil95] T. M. Williams. A classified bibliography of recent research relating to project risk management. *European Journal of Operational Research*, pages 18–38, 1995.
- [WPB99] Ray C Williams, George J Pandelios, and Sandra G Behrens. *Software Risk Evaluation (SRE) Method Description: Version 2.0*. Carnegie Mellon University, Software Engineering Institute, 1999.
- [YZ<sup>+</sup>09] Benhai Yu, Jinlong Zhang, et al. The application of fault tree analysis in software project risk management. In *Management and Service Science, 2009. MASS'09. International Conference on*, pages 1–4. IEEE, 2009.