

Análise da Precisão de Estimativas de Projetos de Software Utilizando Redes Bayesianas

Kelly Bettio, Andreia Malucelli, Giovanna Tiboni, Renato Ferraz Machado

Programa de Pós-Graduação em Informática (PPGIA)
Pontifícia Universidade Católica do Paraná (PUCPR)
Curitiba – PR – Brasil

kelly.bettio@pucpr.br, malu@ppgia.pucpr.br, giovanna.tiboni@gmail.com,
rfmachado@gmail.com

Abstract. *Software development companies have some difficulties in estimating the time for project development. One of these difficulties is the lack of resources to invest in process improvement and consequently to build a consistent software project measures historical basis. This paper aims to evaluate the use of Bayesian Networks to verify the accuracy and to support the decision to perform a refinement of software projects estimation in their early stages. This is an exploratory research using a software development projects database. It was possible to calculate the estimates correctness probability, indicating the need for a refinement.*

Resumo. *Empresas de desenvolvimento de software encontram dificuldades em estimar o tempo para o desenvolvimento de seus projetos. Uma dessas dificuldades é a falta de recursos para investir em melhoria de processos e conseqüentemente na formação de uma base histórica consistente de medidas de projetos. Este artigo tem como objetivo avaliar o uso de Redes Bayesianas para verificar a precisão e apoiar a decisão de executar um refinamento de estimativas de projetos de software em suas fases iniciais. Trata-se de uma pesquisa exploratória utilizando base de dados de projetos de desenvolvimento. Foi possível calcular a probabilidade de correteude das estimativas, indicando a necessidade de um refinamento.*

1. Introdução

Uma das principais razões para se realizar estimativas em projetos de software é auxiliar o planejamento do seu desenvolvimento. A qualidade da estimativa geralmente impacta na qualidade do planejamento e conseqüentemente na alocação de recursos e cálculo de custos envolvidos na realização do projeto [Humphrey 1995].

Segundo Barcellos et al (2003), a quantidade de projetos sem sucesso é um dos pontos mais frustrantes do desenvolvimento de software, resultando em oportunidades perdidas e insatisfação do cliente. Os autores ressaltam ainda que o sucesso de um projeto depende da habilidade do gerente estimar os custos e prazos no início do desenvolvimento e controlá-los durante o processo.

A estimativa de duração de uma atividade é o processo de aproximação do número de períodos de trabalho necessários para completar as atividades com os recursos estimados [PMI 2008]. No entanto, existe dificuldade na realização de estimativas de software devido à incerteza do produto a ser produzido, ou seja, quanto

maior a precisão na definição dos requisitos maior será a precisão na estimativa. Machado (2002) afirma que as práticas da indústria em relação à estimativa estão aquém do esperado, pois quando se estima, o resultado, algumas vezes, se mostra incompatível.

Em projetos de software os requisitos são coletados em fases iniciais e muitas vezes utilizando analogia com produtos previamente desenvolvidos [Humphrey 1995]. Passada a fase de proposta, os requisitos são conhecidos mais detalhadamente e então um refinamento das estimativas pode ser feito de forma a identificar melhor o tamanho do trabalho a ser desenvolvido.

De acordo com Gomes et al (2001), para alcançar níveis mais altos de qualidade é necessário obter e analisar devidamente os dados quantitativos que descrevam a realidade do processo. Os autores afirmam que medições de software são o fator chave para que isso seja possível, pois facilitam a visualização das oportunidades de melhoria.

Além da seleção adequada do método de estimativa e de uma base de dados consistente, Reinehr, Burnett e Pessôa (2007) complementam que um bom processo de estimativas é a base para um bom processo de planejamento e que seu amadurecimento propicia melhoria organizacional, que é um dos principais objetivos de modelos de maturidade de software como MPS.BR [SOFTEX 2011] e CMMI [SEI 2010]. Porém, mesmo com uma base histórica é difícil avaliar rapidamente se uma estimativa foi realizada corretamente, o que poderá influenciar o andamento do projeto.

Considerando este contexto, sistemas informatizados com Inteligência Artificial (IA) poderiam auxiliar na manipulação destes dados e na análise da precisão de estimativas de projetos de software, apoiando o gestor na tomada de decisão sobre o ajuste necessário na estimativa estabelecida.

Existem diversas técnicas de IA, sendo que cada uma possui objetivos específicos e uma delas, aplicada neste trabalho, é a Rede Bayesiana (RB). A RB, segundo Russell e Norvig (2004), é uma forma de representar o conhecimento em um sistema especialista probabilístico, sendo um grafo orientado acíclico, onde cada nó é identificado com informações de probabilidade quantitativa.

Segundo Pendharkar, Subramanian e Rodger (2005), modelos baseados em RBs têm sido usados em estudos recentes para estimar esforço em projetos de software. Eles desenvolveram um estudo com 33 projetos de software comparando o desempenho da RB com modelos de Redes Neurais e Regressão e descobriram que a RB é um modelo competitivo na previsão de esforço em projetos de desenvolvimento de software.

Diante deste cenário, este artigo tem como objetivo avaliar o uso de RB para verificar a precisão e apoiar a decisão de executar um refinamento de estimativas na fase inicial de projetos de desenvolvimento de software. Para isso, foi feita uma pesquisa exploratória utilizando a base de dados de projetos de desenvolvimento de uma empresa de médio porte da área médica. Foi possível calcular a probabilidade de correteude das estimativas e a partir destas probabilidades, indicar a necessidade de refinamentos.

Este artigo está organizado apresentando uma fundamentação teórica sobre Estimativas de Software e Redes Bayesianas na Seção 2. A Seção 3 descreve o método de pesquisa utilizado na realização do experimento. Os resultados e discussões a cerca

das descobertas estão descritos na Seção 4, e a Seção 5 conclui este artigo e apresenta os trabalhos futuros.

2. Fundamentação Teórica

Nesta seção são apresentados os conceitos necessários para um melhor entendimento deste trabalho.

2.1 Estimativas de Software

O processo de aproximação do número de períodos de trabalho necessários para completar as atividades de um projeto é chamado de estimativa de duração [PMI 2008].

Geralmente as estimativas de esforço e custo de software são realizadas utilizando técnicas adequadas e/ou bases históricas de projetos já finalizados que servem como parâmetros de planejamento. Estas estimativas consideram principalmente fatores como escopo, produtos de trabalho e tarefas previstas para o projeto associados à produtividade [SOFTEX 2011].

As medidas coletadas em projetos previamente desenvolvidos e armazenadas em uma base histórica são usadas para determinar o trabalho atual do software. As estimativas de software podem ser baseadas em diversos parâmetros dependendo do contexto nas quais estão inseridas. Alguns destes parâmetros, segundo Royce (1998), são:

- Tamanho funcional, que indica a complexidade do software, ou seja, a quantidade de software a ser produzida. Tem-se como exemplo o número de linhas de código, número de pontos de função e número de pontos de casos de uso.
- Produtividade, que indica um conjunto de componentes obtidos num certo período de tempo. Tem-se como exemplo o número de pontos de função por programador por mês.
- Esforço, que indica o esforço a ser empregado dada a estimativa de tamanho e a produtividade. Tem-se como exemplo o número de pontos de função multiplicado pela produtividade por ponto de função.

Os planejamentos de projetos de software necessitam de uma maior precisão para evitar o mau uso de recursos, impactando assim diretamente em seus custos. É possível prever o comportamento futuro baseado no comportamento passado e assim, com o passar do tempo, a precisão das estimativas aumenta, melhorando o planejamento [Humphrey 1995].

Nas últimas duas décadas a grande preocupação da comunidade de Engenharia de Software, de acordo com Chulani, Boehm e Steece (1999), tem sido por estimar mais precisamente os custos envolvidos no desenvolvimento de seus projetos. Muitos modelos têm sido desenvolvidos com esse objetivo e a maioria deles é calibrado, se tornando mais preciso por meio do uso de informações de projetos já finalizados.

A precisão das estimativas nem sempre é possível no início dos projetos, pois muitas vezes as informações ainda são prematuras e necessitam de um refinamento ao longo do seu desenvolvimento, podendo ocasionar mudanças e conseqüentemente afetar os parâmetros do projeto. Segundo Murthi (2002), uma maior precisão destas

estimativas iniciais é necessária, pois minimiza problemas com prazos e custos, como também problemas de monitoramento do progresso do projeto.

De acordo com Boehm, Abts e Chulani (2000), as técnicas para estimar software podem ser classificadas em categorias, algumas destas, entre outras, são: técnicas baseadas em modelo, como por exemplo *Constructive Cost Model* (COCOMO) e SLIM; técnicas orientadas a aprendizagem, como por exemplo Redes Neurais e Raciocínio Baseado em Casos; técnicas baseadas em regressão, como por exemplo Robust; e técnicas compostas (combinação de duas ou mais técnicas), como por exemplo Redes Bayesianas combinadas com COCOMO II.

Marçal et al (2009), propuseram um método de estimativa de projetos de software utilizando a base de dados de uma organização de alta maturidade onde integraram “*Story Points*”, provenientes de metodologias ágeis e “*Use Case Points*”, provenientes de metodologias tradicionais e conseguiram constatar que com a combinação destas técnicas é possível realizar estimativas mais rápidas e com uma base de dados consistente é possível realizar estimativas mais precisas.

Barcellos et al (2003), em seu trabalho relaciona algumas técnicas de estimativas de projetos de software e sua associação com práticas de gestão do conhecimento citando que a combinação de mais de uma técnica tem se mostrado eficiente para melhorar a precisão das estimativas iniciais.

Chulani, Boehm e Steece (1999), executaram um estudo comparativo do uso de técnicas baseadas em regressão múltipla e RB combinados com o modelo COCOMO II e os resultados concluíram que a abordagem bayesiana é mais robusta. Em outro estudo realizado por Boehm, Abts e Chulani (2000), os autores identificam que uma das mais atrativas técnicas utilizadas em composição com o COCOMO II são as RBs.

2.2 Redes Bayesianas

Redes Bayesianas, de acordo com Heckerman (1995), tornaram-se uma importante maneira de representar o conhecimento incerto em sistemas especialistas. A capacidade de realizar inferências bidirecionais, combinada com um fundamento probabilístico rigoroso, levaram as RBs a um rápido aparecimento como o método de escolha para raciocínio de incertezas em IA e sistemas especialistas [Pearl e Russel 2001].

Pesquisadores têm desenvolvido métodos para o aprendizado de RBs a partir de dados. As técnicas que foram desenvolvidas têm se mostrado efetivas para alguns problemas de análise de dados [Heckerman 1995].

Ainda de acordo com Heckerman (1995), existem inúmeras representações disponíveis para análise de dados, como bases de regras, árvores de decisão e redes neurais artificiais e existem também muitas técnicas, tais como estimativa da densidade, classificação, regressão e agrupamento. Porém, as RBs se destacam porque podem prontamente manipular conjuntos de dados incompletos; permitem aprender a partir de relações causais; em conjunto com técnicas estatísticas facilitam a combinação do conhecimento do domínio e de dados; em conjunto com métodos Bayesianos e outros tipos de modelos oferecem uma abordagem eficiente para evitar o *overfitting* dos dados.

A RB, segundo Russel e Norvig (1995), é um modelo de representação do conhecimento que trabalha com o conhecimento incerto e incompleto por meio da

Teoria da Probabilidade Bayesiana. A RB utiliza o conhecimento do especialista para simulá-lo de forma computacional e representa a incerteza no conhecimento com grafos acíclicos e direcionados que representam as dependências probabilísticas entre diversas variáveis.

Em particular, cada nó no grafo representa uma variável aleatória, enquanto que as arestas entre os nós representam dependências probabilísticas entre as variáveis correspondentes. Essas dependências condicionais são muitas vezes estimadas utilizando conhecidos métodos estatísticos e computacionais. Assim, RBs combinam princípios da teoria dos grafos, teoria da probabilidade, ciência da computação e estatística [Ben-Gal 2007].

As variáveis que constituem os nós da RB podem ser discretas ou contínuas, unidas aos pares por vínculos orientados, sendo que o nó de origem é o pai do nó localizado na extremidade da seta. Cada nó está relacionado com uma tabela de probabilidade condicional cujo tamanho está vinculado aos condicionamentos existentes para aquela variável [Russell e Norvig 2004].

As RBs estão fundamentadas no teorema de Bayes, que calcula a probabilidade de um evento acontecer, dado que um outro evento já ocorreu. Trata-se de uma partição do espaço amostral em diversos subconjuntos, os quais apresentam probabilidades conhecidas. As probabilidades previamente conhecidas para um evento específico podem ser ajustadas por meio do teorema de Bayes cada vez que se apresentem novas evidências envolvendo um determinado evento que seja dependente do primeiro. É um formalismo que mistura a teoria dos grafos e a teoria da probabilidade e é composta de uma parte qualitativa e outra quantitativa [IATROS 2010].

A parte qualitativa da RB representa com nós as dependências que correspondem às variáveis do sistema e seus valores. Estes nós (variáveis) estão ligados por arcos direcionados. Assim, um arco ligando as variáveis A e B indica que a variável B é a consequência e a variável A é a causa, numa relação de dependência. A parte quantitativa da RB representa os coeficientes das probabilidades condicionais estimadas de cada valor, *a priori*, das hipóteses e pode ser obtida pela inserção direta das probabilidades nas Tabelas de Probabilidades Condicionais pelos especialistas, ou pela inserção dos casos diretamente de uma base de dados [Ben-Gal 2007].

3. Método de Pesquisa

Trata-se de uma pesquisa exploratória com abordagem quantitativa. O método de trabalho foi dividido em quatro fases: (i) analisar a base de dados (BD), (ii) pré-processar os dados, (iii) construir a RB, e (iv) realizar experimentos e analisar os resultados.

A BD estudada pertence a área de Tecnologia da Informação (TI) de uma empresa de médio porte da área médica, cujos projetos envolvem a evolução de produtos para atendimento de demandas internas. A empresa possui processos de desenvolvimento de software definidos e institucionalizados apesar de não ter se submetido a avaliações formais de modelos de qualidade de software como MPS.BR [SOFTEX 2011] e CMMI [SEI 2010].

3.1 Analisar a Base de Dados

Na primeira etapa foi necessária a análise dos dados para entendimento e avaliação da possibilidade de aplicação das RBs sobre os dados fornecidos pela empresa. Foram realizadas conversas informais com a gerente de projetos, responsável pela equipe de desenvolvimento dos projetos, para entendimento do conteúdo das informações da BD. A BD, fornecida pela empresa, continha os seguintes campos:

- Tipo: classificação das tarefas, podendo assumir valores como: reincidência – falha de desenvolvimento, reincidência – falha de negócios, reincidência – falha de testes ou sem classificação.
- Grupo Executor: área da empresa onde a tarefa foi realizada;
- Executor: nome do profissional responsável pela fase/atividade;
- Prioridade da Tarefa: categoria de tamanho a que se refere à tarefa. É composto por duas letras, onde a primeira indica a complexidade do produto ao qual a tarefa pertence (Grande, Médio ou Pequeno) e a segunda letra indica a complexidade da tarefa em si (Grande, Médio ou Pequeno). As categorias são: GG (Produto Grande e Tarefa Grande); GM (Produto Grande e Tarefa Médio); GP (Produto Grande e Tarefa Pequeno); MG (Produto Médio e Tarefa Grande); MM (Produto Médio e Tarefa Médio); MP (Produto Médio e Tarefa Pequeno); PG (Produto Pequeno e Tarefa Grande); PM (Produto Pequeno e Tarefa Médio); PP (Produto Pequeno e Tarefa Pequeno);
- Tempo Previsto: esforço em homens/hora estimado para a realização da tarefa, atribuído conforme o campo Prioridade da Tarefa. Podendo assumir valores entre 1 e 10 homens/hora;
- Tempo Executado: esforço em homens/hora que foi realmente consumido na realização da tarefa. As faixas de valores de esforço utilizados para estruturação dos dados foram: abaixo de 1; entre 1 e 5; entre 5,1 e 10; entre 10,1 e 15, e acima de 15.
- Estimativa: desvio negativo ou positivo entre o tempo previsto e o tempo executado na realização da tarefa. As faixas de valores de desvio foram utilizadas em porcentagem e distribuídas da seguinte forma: desvio negativo $\leq 10\%$; desvio negativo entre 11% e 50%; desvio negativo entre 51% e 80%; desvio negativo acima de 80%; desvio positivo $\leq 10\%$; desvio positivo entre 11% e 50%; desvio positivo entre 51% e 80%; desvio positivo acima de 80%.

Os dados estudados foram coletados a partir de 190 projetos ocorridos nos anos de 2009 e 2010. São dados de tarefas realizadas para a correção de reincidências, ou seja, falhas ocorridas após o produto ter sido homologado.

3.2 Pré-Processar os Dados

A empresa disponibilizou uma base de dados em formato de planilha Microsoft Excel representando casos de projetos de software. Para que fosse possível utilizar a base de dados, os seguintes procedimentos foram necessários:

- De forma a preservar a confidencialidade acerca das informações dos profissionais, seus nomes foram substituídos por nomes genéricos como, Executor 1, Executor 2 até Executor 14.
- Foram estratificados os dados referentes ao tempo executado e cálculo do desvio entre previsto e executado, porque a quantidade de valores não era gerenciável pelo software utilizado para construção da RB.
- Foi incluído um campo para cálculo do desvio entre tempo previsto e executado, para que fosse possível constatar o desvio das estimativas já realizadas pelas equipes (tempo previsto em relação ao realizado efetivamente).
- Foram estratificados os dados referentes ao cálculo do desvio entre tempo previsto e executado porque a quantidade de valores não era gerenciável pelo software.
- Foram selecionados e retirados da BD casos aleatórios para serem utilizados na avaliação da RB.

3.3 Construir a Rede Bayesiana

Para a construção da RB foi utilizado o software *freeware* Netica [Netica 2010], tendo como entrada os casos de projetos reais que passaram pelo pré-processamento na etapa anterior. A amostra se constituiu de 190 casos de projetos, organizados conforme descrito no passo anterior. Para a realização desta fase, as seguintes etapas foram realizadas:

- Construção da parte qualitativa da rede bayesiana: composta por sete variáveis essenciais, denominadas de nós de entrada (Tipo, Grupo Executor, Tempo Previsto, Tempo Executado, Executor e Prioridade da Tarefa) e nó de saída (Estimativa), com as faixas de valores correspondentes. Os nós de entrada são os valores que serão modificados durante a inclusão de cada novo caso. Neste estudo as variáveis de entrada referem-se aos valores que indicam as características de cada projeto e que irão interferir no nó de saída. O nó de saída refere-se aos valores da estimativa que serão apresentados pela RB, de acordo com os valores de entrada.
- Construção da parte quantitativa: foi utilizada a BD composta pelos 190 casos de projetos da empresa, não sendo considerados os casos retirados para avaliação.

3.4 Realizar Experimentos e Analisar os Resultados

Para avaliar a aplicabilidade da RB foram realizados experimentos com os casos separados inicialmente da base de dados, conforme descrito anteriormente. Os casos foram simulados na RB e para a análise foi comparado se a probabilidade indicada pela RB correspondia ao que havia acontecido no caso real.

4. Resultados e Discussão

Esta seção apresenta alguns experimentos realizados com casos extraídos previamente da BD, para os quais se conhecia previamente o resultado da estimativa. Com estes

experimentos foi possível avaliar a aplicabilidade das RBs como apoio para a análise da precisão das estimativas de projetos de software.

Como explicado anteriormente, a RB, apresentada na Figura 1, foi construída utilizando-se a BD com o histórico de projetos de software, fornecida pela empresa. Analisando a RB construída é possível entender o cenário pesquisado em relação a suas estimativas de tempo de desenvolvimento.

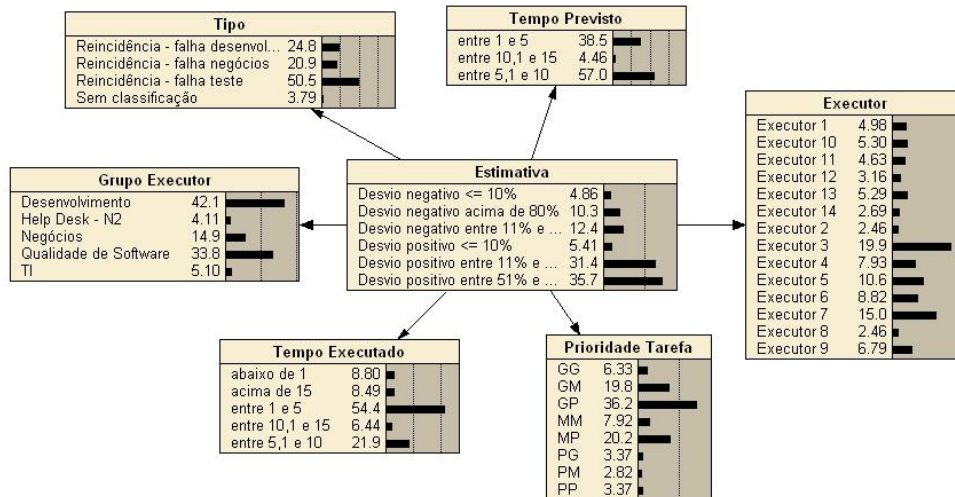


Figura 1 - Rede Bayesiana com casos da BD fornecida pela empresa

A RB mostra que 50,5% das reincidências registradas (nó Tipo) na BD são provenientes de falhas encontradas na fase de testes e 42,1% dos casos apresentados foram originários na equipe de desenvolvimento (nó Grupo Executor). Ainda é possível observar que 36,2% dos projetos têm prioridade GP (Produto Grande e Tarefa Pequena) e 20,2% têm prioridade MP (Produto Médio e Tarefa Pequena) concentrando essas duas categorias mais de 50% dos casos estudados (nó Prioridade Tarefa). Com relação às estimativas de tempo (nó Tempo Previsto), 57% dos casos tiveram tempo previsto entre 5,1 e 10 horas, mas o tempo executado (nó Tempo Executado) apresentou que 54,4% realizaram as tarefas em um tempo entre 1 e 5 horas, sendo possível notar uma falta de precisão nas estimativas iniciais.

Observando os valores das probabilidades estimadas pela RB, apresentadas no nó Estimativa, é possível notar que a categoria mais representativa é a que define o desvio positivo entre o tempo previsto e executado na faixa de valores entre 51% e 80%, representando 35,7% dos casos, se analisado em conjunto com a categoria de desvio positivo entre 11% e 50% que apresentou 31,4% dos casos, nota-se que 67,1% dos casos apresentaram estimativas de tempo em média 50% maiores do que o necessário para sua realização, comprovando assim a falta de precisão nas estimativas iniciais.

A partir desta RB, foram realizados alguns experimentos e quatro casos foram selecionados para serem demonstrados, os quais são apresentados a seguir.

Caso 1: Este caso trata de uma reincidência proveniente da falha nos testes, tendo como grupo executor a equipe de negócios e a atividade foi realizada pelo Executor 1. É uma tarefa considerada com prioridade GP (produto de tamanho Grande e tarefa de tamanho Pequeno), em que o executor estimou 4 horas de desenvolvimento,

mas a tarefa teve a duração efetiva de 16,43 horas. O caso foi simulado na RB e é apresentado na Figura 2.

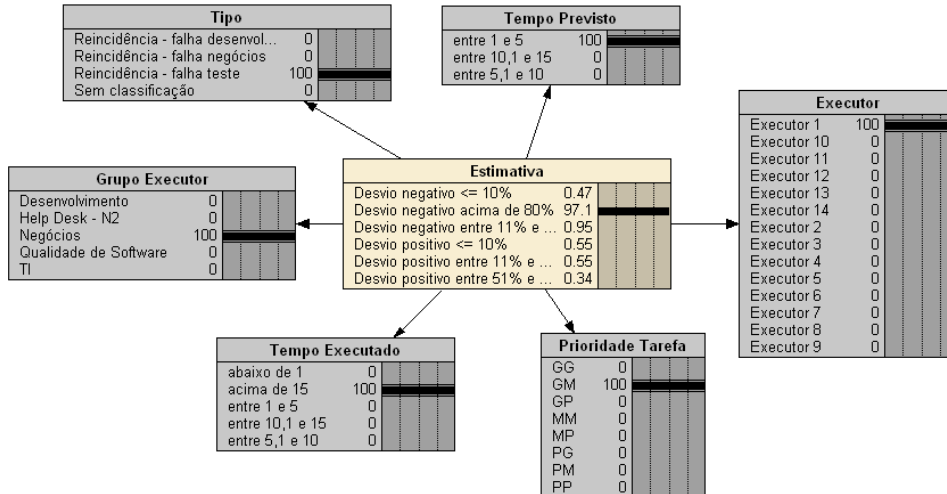


Figura 2 - Rede Bayesiana para o Caso 1

Pode-se observar na Figura 2, nó Estimativa, que a RB indica uma probabilidade de 97.1% da estimativa apresentar um desvio negativo acima de 80%. Caso o executor estivesse utilizando a RB como apoio no momento do refinamento da estimativa, ele teria descoberto que havia necessidade de refinamento. A RB sugere uma estimativa maior de horas para a atividade, uma vez que a variação foi acima de 80%, se comparadas as horas realizadas com as horas inicialmente estimadas.

Caso 2: Este caso se refere a uma reincidência proveniente da falha nos testes, tendo como grupo executor a equipe de *help desk* e a atividade foi realizada pelo Executor 12. É uma tarefa considerada com prioridade GP, em que o executor estimou 5 horas de desenvolvimento, mas a tarefa teve a duração efetiva de 7,68 horas. O caso foi simulado na RB, conforme mostrado na Figura 3.

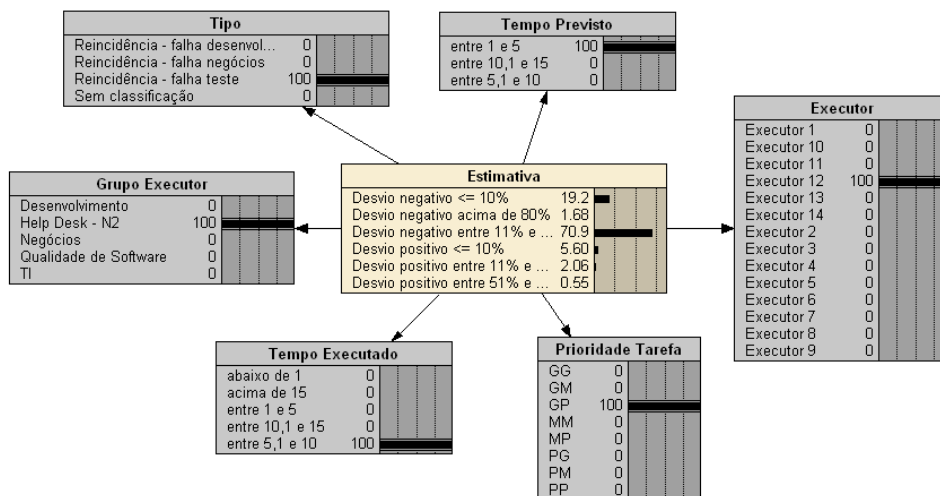


Figura 3 - Rede Bayesiana para o Caso 2

Na simulação do Caso 2, apresentada na Figura 3, o nó Estimativa mostra uma probabilidade de 70,9% da estimativa apresentar um desvio negativo entre 11% e 50%. A RB poderia auxiliar o executor, indicando a necessidade de estimar mais horas para a atividade, considerando que a estimativa foi de 5 horas e a atividade foi executada em 7,68 horas.

Caso 3: Este caso trata de uma reincidência proveniente da falha nos testes, tendo como grupo executor a equipe de desenvolvimento e a atividade foi realizada pelo Executor 5. É uma tarefa considerada com prioridade GP, em que o executor estimou 6 horas de desenvolvimento, mas a tarefa teve a duração efetiva de 3,67 horas, conforme mostrado na Figura 4.

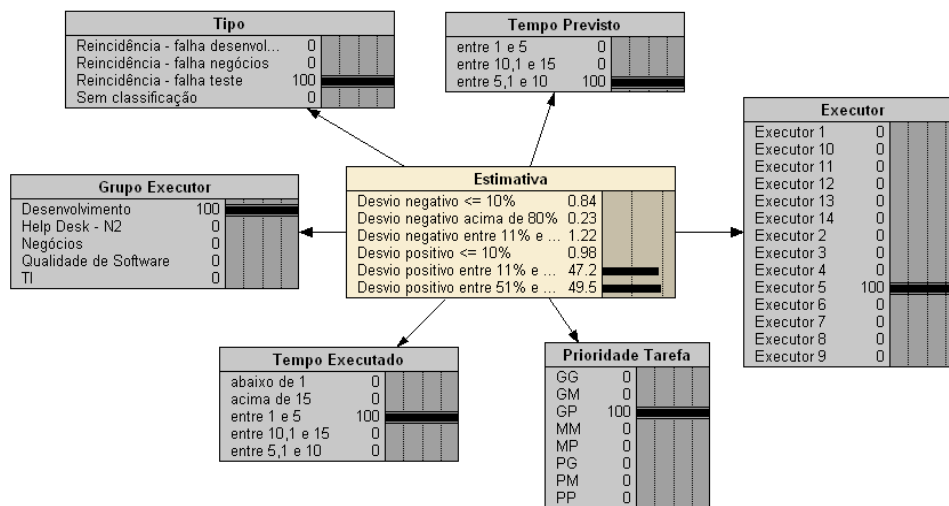


Figura 4 - Rede Bayesiana para o Caso 3

O nó Estimativa, na simulação do Caso 3 (Figura 4), apresenta uma probabilidade de 49,5% da estimativa apresentar um desvio positivo entre 51% e 80%. Neste caso, a RB estimou corretamente que houve um desvio positivo, porém indicou a probabilidade de 49,5% de ocorrer um desvio entre 51% e 80% e a probabilidade de 47,2% do desvio estar entre 11% e 50%, resultando em uma variação pequena entre os dois valores. O desvio calculado não foi maior que 50% se considerado que a estimativa foi de 6 horas e a atividade executada em 3,67 horas. Isso pode ter ocorrido devido a quantidade de casos correspondentes que fazem parte da BD.

Caso 4: Este caso trata de uma reincidência proveniente da falha nos testes, tendo como grupo executor a equipe de desenvolvimento e a atividade foi realizada pelo Executor 12. É uma tarefa considerada com prioridade GP, em que o executor estimou 6 horas de desenvolvimento, mas o projeto teve a duração efetiva de 5,17 horas, conforme detalhado na Figura 5.

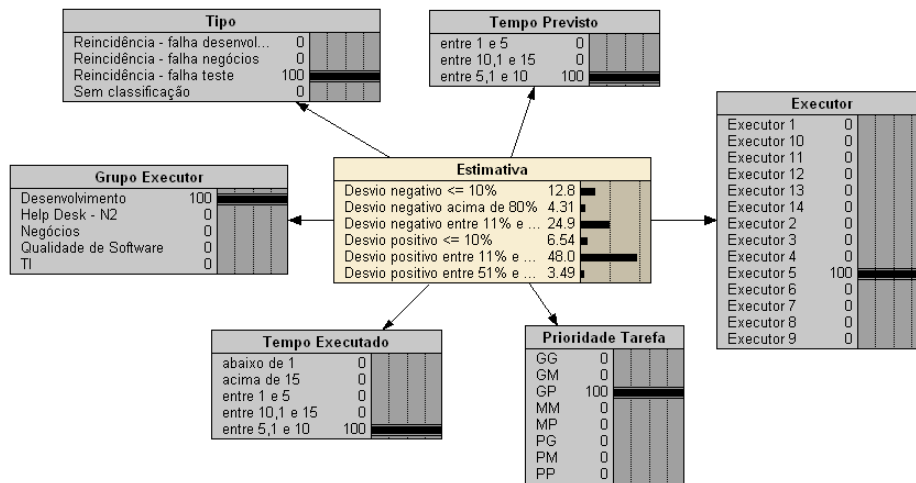


Figura 5 - Rede Bayesiana para o Caso 4

É possível observar na Figura 5 que a RB indicou uma probabilidade de 48% da estimativa apresentar um desvio positivo entre 11% e 50%. Verifica-se assim que a RB auxiliaria o executor indicando que há necessidade de estimar menos horas para a atividade, pois a estimativa foi de 6 horas, sendo necessárias apenas 5,17 horas para sua execução.

Em estimativas de projetos de desenvolvimento de software, como nos casos utilizados neste trabalho, existe uma grande quantidade de variáveis e implicações que demandam um árduo trabalho para se chegar a um valor mais próximo da realidade, tornando difícil que este seja um processo rápido e preciso. Isto ocorre porque existem inúmeros fatores de incerteza que permeiam a estimativa de software, como, por exemplo, a correta compreensão dos requisitos, a experiência de quem está estimando, o nível de experiência de quem executará a tarefa, entre outros fatores. Desta maneira, um sistema de apoio à tomada de decisão utilizando RBs, pode ser de extrema importância, tornando-se mais um apoio para o grupo de desenvolvimento analisar e decidir sobre o refinamento das estimativas em projetos de software. Segundo Murthi (2002), estimativas mais precisas levam a cronogramas e orçamentos mais realistas, melhorando o índice de sucesso em projetos de software.

Mesmo que os resultados apresentados neste trabalho, em função do número limitado de informações, possam apresentar distorções, a utilização de RBs como instrumento de apoio nas estimativas de tempo em projetos de software, mostra-se promissora. De acordo com Meneses (2001), é importante reaproveitar o conhecimento para que o gerente produza estimativas cada vez mais precisas e próximas da realidade.

A carência de técnicas que possam apoiar a identificação da necessidade de refinamento nas estimativas de projetos de software também pode ser identificada a partir da análise do trabalho de Machado (2002). Machado (2002), realizou uma pesquisa de campo que analisou os fatores de risco da indústria nacional de software privada e pública. A pesquisa apontou que o maior índice de não atendimento as estimativas nas saídas do projeto (custo, esforço e prazo) foi quanto a prazo. Em 88 projetos, dos 144 analisados, a estimativa chegou a ultrapassar em 1000%.

5. Conclusões

Segundo Meneses (2001), propor uma estimativa correta e precisa é fator crítico para o sucesso do projeto a ser realizado. Na tentativa de minimizar este fator crítico, este trabalho explorou a aplicação de Redes Bayesianas, tendo sua construção a partir da inserção de casos de projetos de desenvolvimento de software disponibilizados por uma empresa.

A RB se mostrou uma técnica promissora no auxílio durante a análise da precisão das estimativas de software e conseqüentemente no seu refinamento. Com os resultados obtidos pode-se concluir que o grau de incerteza que envolve os dados das estimativas pode ser amortizado pela análise do comportamento da população estudada.

Os resultados dos experimentos realizados mostram que a RB poderia indicar a necessidade de um refinamento da estimativa, reduzindo assim problemas com prazos e custos no projeto. Um exemplo seria o Caso 2 desta pesquisa, em que a RB indicou que existia uma probabilidade de 70,9% da estimativa realizada estar com um desvio negativo entre 11% e 50%, ou seja, havia a necessidade de estimar mais horas para a realização da atividade. Neste caso, havia uma estimativa de 5 horas para a realização da atividade e esta foi realizada em 7,68h, ocorrendo efetivamente um desvio negativo de 54%. A RB poderia auxiliar em uma estimativa mais precisa.

Uma limitação deste trabalho refere-se à quantidade de casos testados e de variáveis utilizadas. Uma quantidade maior de casos e variáveis se faz necessária para que seja possível uma melhor avaliação da efetividade das RBs para estimativas em projetos de desenvolvimento de software. Como trabalhos futuros pode-se citar a realização de um estudo sobre os demais anos que compõem a base histórica da empresa; aplicação das RBs em outras empresas, visando comparar os resultados; criação de diretrizes mais detalhadas para definir as categorias de complexidade de programas e de alteração, visando melhor possibilidade de separação do espaço amostral; inclusão de uma maior quantidade de variáveis, para verificar se os resultados apresentarão melhorias; e integração da RB proposta em um sistema de apoio a estimativas de software.

Referências

- Barcellos, M. P., Figueiredo, S. M., Rocha, A. R. C., Travassos, G. H. (2003) “Utilização de Métodos Paramétricos, Analogias, Julgamento de Especialistas e Conhecimento Organizacional no Planejamento de Tempo e Custos de Projetos de Software”, In: II Simpósio Brasileiro de Qualidade de Software, Fortaleza, Brasil.
- Boehm, B., Abts, C., Chulani, S. (2000) “Software Development Cost Estimation Approaches – A Survey”, In: Annals of Software Engineering 10, CA, USA, pg 177 – 205.
- Ben-Gal, I. (2007) “Bayesian Networks”, In: Ruggeri F., Faltin F. e Kenett R. (Eds.), Encyclopedia of Statistics in Quality and Reliability, John Wiley & Sons, disponível em <http://www.eng.tau.ac.il/~bengal/BN.pdf>, acessado em Agosto de 2011.

- Chulani, S., Boehm, B., Steece, B. (1999) “Bayesian Analysis of Empirical Software Engineering Cost Models”, In: IEEE Transactions on Software Engineering, Volume: 25 Issue:4, 573 – 583.
- Gomes, A., Oliveira, K., Rocha, A. R. (2001) “Avaliação de Processos de Software Baseada em Medições”, In: XV Simpósio Brasileiro de Engenharia de Software (SBES), Rio de Janeiro, Brasil.
- Heckerman, D. (1995) “A Tutorial on Learning With Bayesian Networks”. TechReport. MSR-TR-95-06, disponível em <http://research.microsoft.com/pubs/69588/tr-95-06.pdf>, acessado em Julho de 2011.
- Humphrey, W. S. (1995), A Discipline for Software Engineering, Addison – Wesley, 790.
- IATROS (2010) “Estatística e Pesquisa Científica para Profissionais da Saúde (softwares)”, disponível em <http://www.vadmecum.com.br/iatros/index.htm>, acessado em Janeiro de 2010.
- Machado, C. A. F. (2002) “A-RISK : Um Método para Identificar e Quantificar Risco de Prazo em Projetos de Desenvolvimento de Software”, Dissertação de Mestrado, Pontifícia Universidade Católica do Paraná, Curitiba, Brasil.
- Marçal, A. S. C., Bezerra, C. I. M., Pires, C. G. S.; Coelho, C., Souza, G. T., Barbosa, J. (2009) “Integração de Story Points e Use Case Points em Projetos Baseados em SCRUM e CMMI”, In: VIII Simpósio Brasileiro de Qualidade de Software, Ouro Preto, Brasil.
- Meneses, J. B. (2001) “Inspector: Um processo de Avaliação de Progresso para Projetos de Software”, Dissertação de Mestrado, Universidade Federal de Pernambuco, Recife, Brasil.
- Murthi, S. (2002) “Preventive Risk Management for Software Projects”, In: IEEE – IT Pro, Sep./Oct, 9-15.
- Netica (2010) “Ferramenta Netica”, disponível em <http://www.norsys.com/netica.html>, acessado em Novembro de 2010.
- Pearl, J. , Russel, S. (2001). “Bayesian networks”. Report (R-277), November 2000, In: Handbook of Brain Theory and Neural Networks, Arbib, M. (Ed), MIT Press, Cambridge, 157–160.
- Pendharkar, P.C., Subramanian, G.H. e Rodger, J.A. (2005) “A Probabilistic Model for Predicting Software Development Effort”, Sch. Of Bus. Adm, In: IEEE Transaction on Software Engineering, Pennsylvania, PA, July.
- PMI – Project Management Institute (2008) “A Guide to the Project Management Body of Knowledge: PMBOK Guide – 4rd Edition”, Newton Square: PMI Publications.
- Reinehr, S., Burnett, R., Pessôa, M. (2007) “PSPi – Melhorando Estimativas no Ambiente Corporativo”, In: VI Simpósio Brasileiro de Qualidade de Software (SBQS), Porto de Galinhas, Brasil.
- Royce, W. (1998), Software Project Management: a Unified Framework, Addison Wesley, 406.

Russell, S., Norvig, P. (2004), *Inteligência Artificial*, 2ª ed. Ed. Campus, Rio de Janeiro, 1021.

SEI - Software Engineering Institute (2010) "CMMI for Development", Version 1.3, Pittsburgh, PA.

SOFTEX (2011) "MPS.BR – Guia Geral", v. MR-MPS: 2011. disponível em http://www.softex.br/mpsbr/_guias/guias/MPS.BR_Guia_Geral_2011.pdf, acessado em Fevereiro de 2012.