UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

BRUNO LOPES DALMAZO

Uma Abordagem Bayesiana para Previsão de Custos de Suporte de Projetos de Gerenciamento de TI

Dissertação apresentada como requisito parcial
para a obtenção do grau de
Mestre em Ciência da Computação

Prof. Dr. Luciano Paschoal Gaspary
Orientador

Porto Alegre, Novembro de 2011

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Dalmazo, Bruno Lopes

Uma Abordagem Bayesiana para Previsão de Custos de Suporte de Projetos de Gerenciamento de TI / Bruno Lopes Dalmazo. – Porto Alegre: PPGC da UFRGS, 2011.

68 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2011. Orientador: Luciano Paschoal Gaspary.

1. Tecnologia da Informação. 2. Modelo de informação. 3. Gerenciamento de projetos. 4. Estimativas de custos. 5. Redes Bayesianas. I. Gaspary, Luciano Paschoal. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Reitor: Prof. Carlos Alexandre Netto
Vice-Reitor: Prof. Pedro Cezar Dutra Fonseca
Pró-Reitora de Pós-Graduação: Prof. Aldo Bolten Lucion
Diretor do Instituto de Informática: Prof. Flávio Rech Wagner
Coordenador do PPGC: Prof. Álvaro Freitas Moreira
Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

"Hakuna Matata "
— Jambo Bwana

AGRADECIMENTOS

Agradeço aos meus pais, Ricardo e Fátima, pelo apoio que eu sempre recebi em todos os momentos da minha vida. Quero que vocês saibam que todas as minhas conquistas são resultado da educação e dos ensinamentos que vocês me transmitiram. Meus agradecimentos também à minha irmã Aline. Minha lembrança especial ao resto da família (que é muito grande) que também foi muito importante para eu chegar até aqui.

É um grande prazer também lembrar aqui todas as pessoas que me facilitaram e tornaram a minha estadia em Porto Alegre muito boa: Zita, Elton, Luidi, Flávia e todos os amigos que fiz por aí.

Meu agradecimento todo especial aos Profs. Luciano Gaspary, Lisandro Granville e Marinho Barcellos, pelos ensinamentos e pela santa paciência em terem conseguido me aturar durante o mestrado. A boa notícia para vocês é que esse período finalmente acabou e estou indo para Portugal! Ráá́á!

Não posso esquecer também os integrantes do Grupo de Redes do Inf/UFRGS. Espero que vocês tenham aprendido alguma coisa de fundamento comigo, assim como aprendi lições importantes com vocês: Weverton, Roben, Juliano, Jéferson, Flávio Baratinaicos, Pedro Jedi, Jair, Rafael Esteves, Rodolfo, Adler e o grande Luís Armando (Nautilus)! Agradeço também à dupla Paleta (Rodrigo Mansilha e Ricardo) pelas sempre divertidas férias, viagens, festas e levitadas!

Valeu galera!

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
AP	Access Point
BN	Bayesian Network
CIM	Common Information Model
CPT	Conditional Probability Table
COCOMO	Constructive Cost Model
DB	Database
DNS	Domain Name System
IC	Item de Configuração
ISBSG	International Software Benchmarking Standards Group
ITIL	Information Technology Infrastructure Library
ITSM	Information Technology Service Management
PF	Pontos de Função
PMBOK	Project Management Body of Knowledge
TI	Tecnologia da Informação
XML	Extended Markup Language
W3C	World Wide Web Consortium
WfMC	Workflow Management Coalition Specification
WPA2	Wi-Fi Protected Access II
WWW	World Wide Web

LISTA DE FIGURAS

LISTA DE TABELAS

RESUMO

Existe uma noção intuitiva de que os custos associados a ações de suporte de projetos de gerenciamento de Tecnologia da Informação (TI), muitas vezes considerados já muito elevados e em crescimento, possuem forte vinculação com esforços empreendidos nas fases de desenvolvimento/implantação e teste. Apesar da importância de caracterizar e compreender a sistemática dessa relação, pouco tem sido feito neste domínio, principalmente devido à falta de mecanismos adequados tanto para o compartilhamento de informações entre as fases de um projeto de TI, quanto para aprender com experiências passadas.

Para lidar com essa problemática, propõe-se nesta dissertação uma abordagem para estimar dinamicamente os custos de suporte de projetos de gerenciamento de TI à luz de informações provenientes das fases de desenvolvimento/implantação e teste. As estimativas de custos são calculadas a partir da integração de informações produzidas ao longo do ciclo de vida de projetos (passados). O núcleo da solução presente neste trabalho conta com um modelo Bayesiano para realizar previsão de custos de suporte, apoiado em um modelo de informação usado para persistir informações históricas. Para provar conceito e viabilidade técnica da solução proposta considerou-se, como estudo de caso, a predição de custos associados com projetos de implantação de infraestrutura de redes sem fio. Durante a avaliação é demonstrada a eficácia e eficiência do modelo, bem como discutido suas potencialidades e limitações para auxiliar no entendimento do compromisso entre custos de desenvolvimento/implantação, teste e suporte. A avaliação conduzida fez uso de dados reais/sintéticos produzidos a partir de projetos do ISBSG e apresenta resultados próximos dos encontrados em cenários reais. Nossa abordagem obteve cerca de 80% de acerto na estimativa dos custos de suporte para os cenários avaliados.

Palavras-chave: Tecnologia da Informação, modelo de informação, gerenciamento de projetos, estimativas de custos, redes Bayesianas.

A Bayesian Approach to Predict
Support Costs of IT Management Projects

ABSTRACT

There is an intuitive notion that the costs associated with IT management project support actions, often deemed extremely high and increasing, are directly related to the effort spent during their development/deployment and test phases. Despite the importance of systematically characterizing and understanding this relationship, little has been done in this realm mainly due to the lack of proper mechanisms for both sharing information between IT project phases and learning from past experiences.

To tackle this issue, in this dissertation we proposed an approach for dynamically predicting IT management project support costs taking into account information gathered from the development/deployment and test phases. Support cost estimates are computed by integrating existing information from the lifecycle of (past) projects. The core of the solution in this work relies on a Bayesian model to perform support cost predictions, supported by an information model employed to persist historical information gathered from past projects. To prove the concept and technical feasibility of our solution we consider as a case study the prediction of costs (either development/test/support) associated with projects for the deployment of wireless network infrastructures. During the evaluation is demonstrated the effectiveness and efficiency of the model and discussed its potential and limitations in order to help understanding the trade-offs between development/deployment, test, and support costs. Our solution has been evaluated based on real/synthetics data gathered from the ISBSG dataset, and presents results similar to those found in real-life scenarios. Our solution has provided correct estimates for around 80% of the support costs for the scenarios evaluated.

Keywords: Information Technology, information model, project management, cost estimation, Bayesian networks.

1 INTRODUÇÃO

O gerenciamento do ciclo de vida de projetos de Tecnologia de Informação (TI) consiste em uma abordagem sistemática para organizar a execução de projetos e tem por finalidade contribuir para aumentar produtividade da equipe, aprimorar qualidade dos produtos e reduzir custos dos projetos (PMBOK, 2008). Projetos de TI geralmente consistem na implantação e/ou manutenção de uma infraestrutura de *software* e *hardware*. Não raro, são materializados através de uma sucessão de fases, como análise, planejamento, desenvolvimento/implantação[1], testes e suporte. Ao estabelecer o uso de boas práticas e preconizar a monitoração e controle de cada fase (que, em conjunto, configuram um processo) deseja-se facilitar e padronizar a execução de projetos dessa natureza.

Três fases importantes do gerenciamento do ciclo de vida de projetos recebem atenção especial neste trabalho: desenvolvimento/implantação, teste e suporte. O relacionamento entre essas três fases ocorre, tipicamente, da seguinte forma. Uma vez que um projeto é aprovado e os seus requisitos de negócio são capturados e entendidos, o mesmo pode, então, ser executado. Paralelamente à execução, ou em um próximo momento, o produto (ou serviço em desenvolvimento/implantação) pode ser testado, com o propósito de assegurar que satisfaz os requisitos funcionais e não funcionais previamente elicitados.

Durante a fase de teste, erros podem ser encontrados, levando assim à criação de relatórios. Enquanto alguns desses erros são corrigidos, outros, devido a restrições de tempo e custo, são apenas documentados. Como tais erros se manifestam após a implantação e entrega do projeto, eles são tratados como incidentes e/ou alçados para problemas (ITIL, 2010). A partir desse momento, passam a demandar, do setor da organização responsável pelo suporte, o consumo de recursos (materiais e humanos) para atenuar os efeitos negativos produzidos. A partir deste ponto, há duas situações possíveis: a equipe de suporte pode dar origem a um novo projeto de TI para lidar com o problema relatado, ou pode apenas indicar ao usuário qual é o procedimento de solução a ser adotado.

O esforço despendido para executar e apoiar as ações de suporte, naturalmente, tem um custo associado. Acredita-se que exista uma forte relação entre tal esforço e

[1]No contexto desta dissertação, a palavra *desenvolvimento* refere-se à fase em que as atividades do projeto - seja ele de configuração, de mudança, etc. - são conduzidas. Por esta razão, e para *evitar* sua associação com desenvolvimento de *software*, aparecerá ao longo do texto acompanhada da palavra *implantação*.

o realizado durante as fases de desenvolvimento/implantação e teste de um projeto de TI. Caracterizar e compreender de maneira sistemática esta relação está longe de representar uma tarefa trivial e, consequentemente, é pouco estudada por duas razões principais. Em primeiro lugar, o compartilhamento de informação entre as várias fases que compõem o ciclo de vida de projetos de TI é dificultado pela falta de suporte apropriado por parte das ferramentas existentes, com poucos (ou nenhum) relacionamentos estabelecidos entre essas fases. O segundo problema, por sua vez, é que pouco conhecimento é extraído a partir dessas ferramentas de modo a propiciar o aprendizado com experiências passadas.

A motivação para abordar os problemas recém mencionados e, mais especificamente, determinar a relação entre as fases de desenvolvimento/implantação, teste e suporte reside na possibilidade de apoiar gerentes a responderem perguntas como: *quanto tempo e esforço um projeto exigirá da equipe de suporte após sua implantação?* Ou: *como planejar desenvolvimento/implantação e teste dado um limite máximo de custo de suporte?* Respostas para essas perguntas podem oferecer às organizações a oportunidade de aumentar a produtividade da equipe e a qualidade dos produtos/serviços, bem como melhorar o planejamento e a implantação de projetos futuros.

Para suprir esta lacuna, nessa dissertação propõe-se uma solução que permite estimar custos de suporte à luz de informações oriundas das fases de desenvolvimento/implantação e teste. Ao contrário de outros trabalhos conduzidos na área, nossa abordagem relaciona informações produzidas em diferentes fases do ciclo de vida de projetos. Mais importante, por empregar uma rede de aprendizagem com alimentação dinâmica, permite não somente responder questões no sentido direto (desenvolvimento $\rightarrow$ teste $\rightarrow$ suporte), como também no sentido inverso (suporte $\rightarrow$ teste $\rightarrow$ desenvolvimento), obtendo previsões em consonância com o histórico recente dos projetos conduzidos pela organização. Para provar conceito e viabilidade técnica, a solução é avaliada com o uso de dados reais e sintéticos produzidos a partir de projetos do ISBSG (ISBSG, 2007), referentes a instâncias de projetos de implantação de infraestrutura de redes sem fio.

O restante da dissertação está organizado como segue. No Capítulo 2 aborda-se fundamentos e boas práticas de gerência de projetos de TI e alguns dos principais trabalhos relacionados. No Capítulo 3 apresenta-se a solução proposta para estimar custos de suporte, com destaque para o modelo de informação que permite persistir, de forma integrada, dados produzidos em diferentes fases do ciclo de vida de projetos e para o modelo Bayesiano que embasa o processo de predição. No Capítulo 4 é detalhado o protótipo do sistema \$UPPORT desenvolvido no contexto dessa dissertação, enquanto no Capítulo 5 descreve-se a avaliação experimental conduzida utilizando esse sistema e discute-se os resultados obtidos. Por fim, o Capítulo 6 encerra a dissertação com síntese das contribuições, considerações finais e perspectivas de trabalhos futuros.

2 REVISÃO BIBLIOGRÁFICA

Neste capítulo são sintetizados alguns conceitos-chave da área de projetos de TI. A Seção 2.1 apresenta, resumidamente, alguns dos principais fundamentos do gerenciamento de projetos de TI. Para isso, são consideradas as boas práticas empregadas no gerenciamento dessa natureza de projetos e as principais ferramentas e sistemas utilizados nesse contexto. Na Seção 2.2 é apresentado um formalismo matemático baseado em gráficos de dependência probabilística para representar redes de aprendizagem, conhecido por Rede Bayesiana (*Bayesian Network* - BN). Essa rede de aprendizagem é uma das mais relevantes e adotadas para raciocínio (conclusões) baseado na incerteza, ou seja, produzir estimativas em ambientes que possuem deficiência de dados. A Seção 2.3, por sua vez, encerra o capítulo com alguns dos trabalhos mais proeminentes relacionados ao tema de estimativas de custos de suporte em projetos de TI.

2.1 Fundamentos em Gerenciamento de Projetos de TI

Esta seção tem por objetivo definir o escopo dos projetos abordados na dissertação, bem como apresentar as boas práticas recomendadas para executá-los de maneira eficaz e eficiente. Nessa seção também são apresentadas ferramentas e sistemas de apoio às fases de análise, planejamento, desenvolvimento/implantação, testes e suporte. Essas ferramentas de auxílio têm como objetivo facilitar o trabalho do gerente auxiliando na coordenação de projetos.

2.1.1 Projetos de Tecnologia da Informação

Considerando o escopo desta dissertação, projetos de Tecnologia da Informação são aqueles que têm por objetivo a implantação, a automatização, a modernização e/ou a expansão da infraestrutura de *hardware* e *software* de uma organização. Com o intuito de caracterizar o tipo de projeto abordado nesse trabalho, a seguir é descrito um exemplo de projeto de TI que é tipicamente instanciado em situações reais.

Suponha que uma organização deseje expandir e melhorar a atual infraestrutura de rede cabeada para uma rede de computadores sem fio. Após definido esse objetivo (*upgrade* da infraestrutura de rede), um projeto de TI com esse propósito, denominado WNet ao longo dessa dissertação, poderia ser instanciado (Figura 2.1 nível A). Desde o levantamento de requisitos, passando pelo planejamento e desen-

volvimento/implantação até seus planos de testes e suporte, o projeto pode ser visto como um conjunto de ações independentes, porém complementares.

Fazendo uma análise *top-down*, o projeto poderia ser dividido em macro etapas. Cada etapa seria representada pela execução de um subprojeto em cada um dos departamentos (*e.g.*, almoxarifado, *marketing*, administração, relações públicas, etc.) da organização. Como pode ser observado na Figura 2.1 nível B.

Dessa forma, cada um desses departamentos receberia atenção individual, desde a instalação de equipamentos, até a utilização dos serviços pelo usuário. Para tal, rodadas de desenvolvimento são definidas e divididas em novas etapas, desta vez menores e com escopo mais específico (Figura 2.1 nível C). Por exemplo: o departamento de *marketing* contaria com duas etapas, uma primeira com a implantação da infraestrutura de rede sem a preocupação com questões de segurança, com o objetivo apenas de garantir conectividade e, posteriormente, uma segunda com suporte às questões de segurança. Da mesma forma, os outros departamentos da organização seriam tratados de forma análoga.

Detalhando um pouco mais, neste caso específico (implantação da infraestrutura), cada uma das etapas seria composta por duas funcionalidades distintas. Uma comprometida com a infraestrutura física da rede, que é transparente ao usuário (*back-end*), e outra preocupada com as interfaces gráficas que permitem ao usuário interagir com a infraestrutura (*front-end*) como está ilustrado na Figura 2.1 nível D.

Note que cada uma dessas funcionalidades (*back-end* e *front-end*) possui um processo sistemático para sua condução. Esse processo, por sua vez, é composto por um conjunto de fases ordenadas que ajudam na elaboração das funcionalidades (*e.g.*, elicitação de requisitos, planejamento da implantação, instalação de equipamentos, realização de testes, suporte, entre outras). E finalmente, cada uma dessas fases corresponde à porção mínima divisível do projeto em questão (Figura 2.1 nível E).

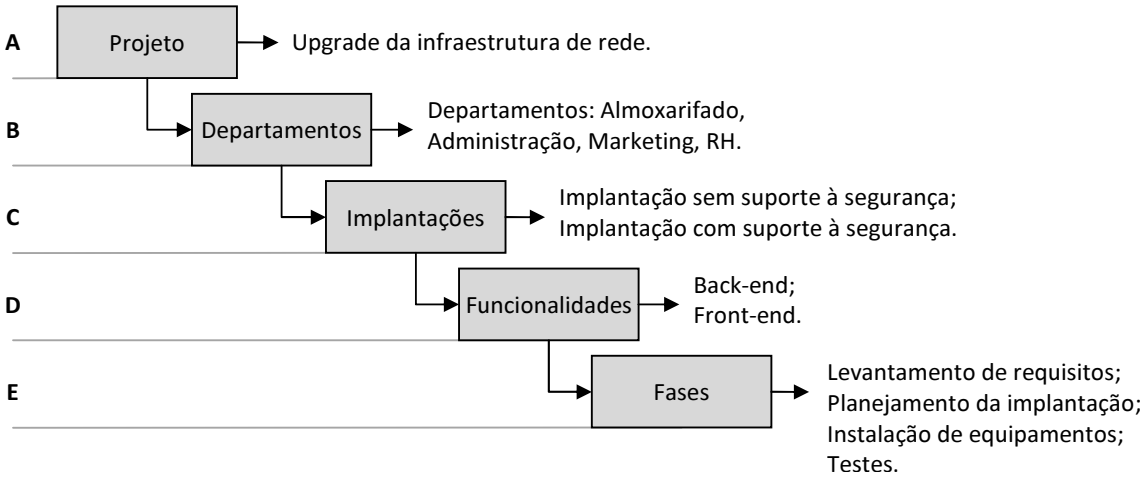


Figura 2.1: Organização de um projeto WNet

2.1.2 Boas Práticas no Gerenciamento de Projetos de TI

Um projeto de TI, como o exemplo citado na subseção anterior, possui metodologias para padronizar e facilitar a sua condução. Entre essas metodologias, duas são as mais adotadas e utilizadas na execução de projetos de TI. Nesse contexto, esse trabalho se propõe a discutir duas dessas metodologias que são as mais conhecidas e amplamente utilizadas: o Modelo em Cascata (ROYCE, 1970) e o Desenvolvimento Iterativo e Incremental (SOARES, 2009).

Primeiramente, no Modelo em Cascata os projetos são organizados para percorrer cada **fase** em sequência e uma única vez, conforme ilustra a Figura 2.2. O conjunto dessas fases (levantamento de requisitos, análise e planejamento, desenvolvimento, testes, implantação) é chamado de **ciclo**. A fase de levantamento de requisitos consiste em definir e enumerar todas as funcionalidades que o projeto deverá atender após sua conclusão. A fase de análise e planejamento é dedicada para a preparação de como se dará a condução do projeto, como por exemplo: tipo e tamanho do projeto, pessoas envolvidas no desenvolvimento, cobertura de testes, etc. Desenvolvimento é a fase do ciclo em que o projeto é desenvolvido/implementado de modo que esteja alinhado com os objetivos definidos na fase de levantamento de requisitos. A fase de testes é dedicada à validação dos artefatos produzidos na fase desenvolvimento. Por fim, a fase de implantação diz respeito ao momento que o projeto será instanciado na organização.

Essa abordagem é eficiente quando trata-se de projetos pequenos, os quais não demandam muito esforço e tempo da equipe de desenvolvimento para serem executados. Se por um lado sua simplicidade implica em um aspecto favorável, por outro, a medida que o projeto torna-se maior, frequentemente essa abordagem é sujeita a falhas. Imagine a seguinte situação: é despendido tempo com a fase de levantamento de requisitos, depois com a análise, e somente em um momento posterior o projeto é desenvolvido, testado e implantado. Seguindo esse modelo de desenvolvimento, não raro, após sua implantação dá-se conta que existem necessidades que não foram contempladas. Essas limitações são decorrentes de um único ciclo de desenvolvimento, característico do Modelo em Cascata, ou também por fases de levantamento de requisitos e análise mal formuladas, visto que essas fases são executadas por humanos e estão sujeitas a falhas (AMBLER; NALBONE; VIZDOS, 2005).

Figura 2.2: Fases de execução do projeto

Em contrapartida, para superar este tipo de limitação, pode-se usar outra abordagem alternativa largamente utilizada nas organizações, o modelo de Desenvolvimento Iterativo e Incremental. Esse modelo se propõe em dividir o projeto em ciclos, conforme ilustra a Figura 2.3. Ao final de cada ciclo tem-se como resultado um entregável. Via de regra, esse entregável possui um porte menor e permite que

sejam feitas observações, as quais serão usadas como entrada no próximo ciclo de desenvolvimento. Ao reusar e avaliar informações coletadas ao final de cada ciclo, pode-se entender melhor os requisitos e, se for o caso, redefinir algum deles. A partir do melhor entendimento dos requisitos do projeto, ele pode ser planejado de forma mais eficiente fazendo com que o produto/serviço final seja mais fidedigno com as necessidades da organização.

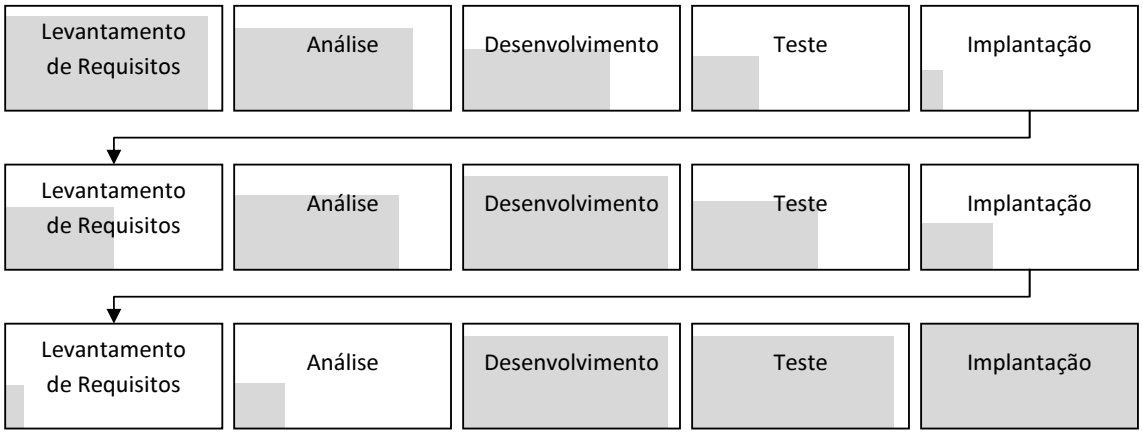


Figura 2.3: Desenvolvimento Iterativo e Incremental

Quando o entregável passar diversas vezes pelas fases de execução do projeto e atingir um nível satisfatório de qualidade (caracterizando uma funcionalidade do projeto), pode-se agrupar esses ciclos em uma **iteração**. Portanto, do ponto de vista do desenvolvimento/implantação, o ciclo de vida de um projeto é uma sucessão de iterações, por meio das quais o produto final se desenvolve de maneira incremental. Cada iteração (ou conjunto delas) termina com a liberação de um subproduto executável e pronto para o uso, também chamada de ***release***. Esse subproduto pode ser apenas um subconjunto da visão completa do projeto, mas mesmo assim ser útil do ponto de vista da organização ou do usuário. Cada *release* é acompanhado de artefatos de apoio, como por exemplo: descrições, documentação do usuário, planos, etc., bem como modelos atualizados do sistema.

Nesse momento o leitor pode observar que o modelo de Desenvolvimento Iterativo e Incremental se presta muito bem como forma de conduzir o projeto que foi apresentado na Subseção 2.1.2. Onde as fases (compostas por atividades) são representadas na Figura 2.1 Nível E. Os ciclos podem ser representados pelas funcionalidades implantadas em *Back-end* e *Front-end* (Figura 2.1 Nível D). As implantações de segurança da Figura 2.1 Nível C se encaixam como iterações. As *releases* são representadas pelos departamentos da organização (Figura 2.1 Nível B). E, por fim, o projeto poder ser visto como o agregador de todas essas etapas no Nível A, topo da Figura 2.1.

2.1.3 Ferramentas e Sistemas

Existe um grande número de ferramentas de apoio ao planejamento, desenvolvimento e suporte de projetos. O objetivo dessas ferramentas e sistemas é facilitar

o trabalho do gerente auxiliando na coordenação e distribuição de tarefas, e promovendo a integração da equipe que desenvolverá o projeto. Duas ferramentas de acompanhamento de projetos foram largamente estudadas e utilizadas neste trabalho, *HP Quality Center* (HEWLETT-PACKARD, 2009) e *Bugzilla* (BARNSON et al., 2004). A partir de agora, algumas das principais características dessas ferramentas serão apresentadas.

O *software HP Quality Center* permite que um projeto possa ser especificado através do modelo de Desenvolvimento Iterativo e Incremental apresentado na subseção anterior. Através desse *software* pode-se organizar o projeto em *releases*, iterações, ciclos e fases. Além disso, ele também facilita, de diferentes maneiras, o acompanhamento da execução de um projeto de tecnologia de informação. Seu uso é baseado em três pilares centrais: levantamento de requisitos, desenvolvimento/implantação e planos de testes.

Através dessa ferramenta é possível planejar todas as fases que serão conduzidas durante a execução do projeto de tecnologia da informação. Com isso pode-se associar humanos às atividades de desenvolvimento, mais especificamente falando, conectar planos de testes com requisitos e pessoas responsáveis por sua realização. A partir desse *software*, o gerente do projeto pode obter várias informações importantes sobre o projeto em questão. Além disso, ele permite o controle sobre planos de testes que cobrem todos os requisitos de uma determinada *release* do projeto. Por exemplo: pode-se contabilizar quantos humanos serão necessários para executar as atividades de testes propostas, consultar qual a cobertura de testes que já foi executada no projeto, ou então, quantas horas já foram gastas com uma determinada fase do projeto.

A ferramenta *Bugzilla*, por outro lado, trata do momento posterior ao da implantação do projeto. Através desse sistema de *troubletickets*, o cliente pode fazer chamadas ao sistema (abrir um bilhete) para registrar alguma deficiência ou dificuldade de manuseio do produto/serviço entregue. Esse sistema permite a associação entre as chamadas e humanos, delegando responsáveis (ou grupo deles) para mitigar as limitações encontradas. Nem todos os bilhetes exigem uma reimplementação do produto/serviço entregue, alguns são identificados como restrições de uso, nesses casos a indicação de um procedimento corretivo é suficiente para fechar a chamada ao sistema.

Nessa seção foram vistas duas ferramentas muito consolidadas no mercado que são amplamente utilizadas durante o ciclo de vida de um projeto de tecnologia de informação. Note que esse tipo de ferramenta pode gerar fontes de dados com ricas informações como saída. Esses dados representam um histórico detalhado a respeito de projetos passados que poderiam ser reaproveitados como estatísticas ou até mesmo ajudando no planejamento de novos projetos futuros. Via de regra, esses dados não são utilizados e nem ao menos existe qualquer forma de conexão entre eles e as fases do projeto. No próximo capítulo, é proposta uma solução que faz uso desse histórico a favor de obter-se uma estimativa qualificada a respeito da fase de suporte baseado em dados de desenvolvimento/implantação e teste de projetos passados.

2.2 Redes de Aprendizagem

Informações produzidas ao longo do ciclo de vida de projetos de TI podem ser aproveitadas e utilizadas para melhor planejar projetos futuros. Mesmo seguindo boas práticas e processos para o gerenciamento de projetos de TI, há um momento em que o volume de informação gerado é grande demais para ser administrado manualmente por gerentes. Neste ponto, faz-se necessária uma metodologia eficiente e eficaz para manusear e organizar os dados. Uma boa solução para esta problemática é a adoção de um formalismo matemático que permita representar dados, consequentemente, facilitando a obtenção de estimativas a partir do processamento de dados brutos.

Redes de aprendizagem estão recebendo considerável atenção dos cientistas e engenheiros nas últimas décadas em vários campos, incluindo ciência da computação, ciência cognitiva, estatística, etc. Inseridas no contexto dessas redes, existem duas principais abordagens que podem ser utilizadas: raciocínio lógico e raciocínio probabilístico. O raciocínio lógico pondera sobre o conhecimento prévio adquirido a respeito do problema e, sobre essa base de conhecimento, infere suas conclusões. Apesar de poderosa, essa abordagem pode não ser útil em situações de ignorância teórica (não se conhece previamente todo o escopo do problema) e/ou impossibilidade (quando é muito oneroso inserir informações na base de dados e capturar as inferências a respeito do domínio do problema).

Lidar em ambientes com falta de informação significa lidar com incertezas. Para suprir essa lacuna, o uso de conectivos que manipulem níveis de certeza, e não apenas valores booleanos (verdadeiro e falso), são indispensáveis. Rede Bayesiana (*Bayesian Network* - BN) se destaca sendo uma das redes de aprendizagem mais utilizadas em pesquisas no meio acadêmico. BNs oferecem uma excelente solução em que conclusões não podem ser obtidas apenas no domínio do problema, necessitando o uso de probabilidade e experimentação (MENDES; MOSLEY, 2008). BNs proporcionam uma abordagem para o raciocínio probabilístico que engloba teoria de grafos para estabelecer relações entre sentenças e, ainda, teoria de probabilidades, para a atribuição de níveis de confiabilidade. Em ciência da computação, o desenvolvimento de BN está dirigido por pesquisas em inteligência artificial que visam à produção de *frameworks* práticos para raciocínio do senso comum (DARWICHE, 2010). A estatística também contribuiu muito para a difusão das BNs, as quais são suportadas e fundamentadas a partir de modelos gráficos que fazem o uso de probabilidades.

Curiosamente, outros campos vêm ganhando força na pesquisa através de BN, tais como análise genética, reconhecimento de fala, teoria da informação, análise de confiabilidade. Esses exemplos podem ser pensados como instâncias concretas de casos restritos de BNs. Por exemplo, *pedigrees* associados com informações de fenótipo e genótipo, diagramas de blocos de confiabilidade e modelos de *Markov* ocultos (usados em reconhecimento de fala e bioinformática) (DUDA; HART; STORK, 2001) também podem ser vistos como instâncias de BNs. Da mesma forma, instâncias canônicas de BNs existem e têm sido utilizadas para resolver problemas clássicos

que abrangem vários domínios, tais como visão computacional e diagnóstico médico (OLIVER; ROSARIO; PENTLAND, 2002).

Uma BN provê um método sistemático para estruturar coerentemente informações probabilísticas a respeito de um evento. BNs também oferecem um conjunto de algoritmos que permitem derivar automaticamente muitas implicações a respeito destas informações, que vem a formar a base das conclusões e decisões inferidas sobre a situação correspondente. Tecnicamente falando, uma rede Bayesiana é uma representação compacta de uma distribuição de probabilidade que normalmente é muito grande para ser tratada por meio de especificações tradicionais de probabilidade e estatística, tais como tabelas e equações. Redes Bayesianas com milhares de variáveis têm sido construídas e utilizadas com sucesso, permitindo uma forma eficiente para representar e raciocinar sobre distribuições de probabilidades.

Uma BN consiste em um conjunto de variáveis ligadas através de arcos e um conjunto de tabelas de probabilidades condicionais (*Conditional Probability Tables* - CPTs). Cada variável possui um conjunto limitado de estados mutuamente exclusivos. As variáveis e arcos formam um grafo dirigido e sem ciclos. Variáveis incondicionais são aquelas que não dependem de outras variáveis; também são conhecidas por nós raízes ou nós pais. Variáveis condicionais são as influenciadas por outras variáveis; também são conhecidas como nós folhas ou nós filhos. Uma rede Bayesiana deve incluir uma CPT para cada variável, que quantifica a relação entre essa variável e todas as suas variáveis pais. Através do uso de redes Bayesianas é possível realizar simples consultas a partir de informações históricas, ou até mesmo, a partir de evidências, ajudar o gerente de projeto na tomada de decisões baseadas em probabilidades.

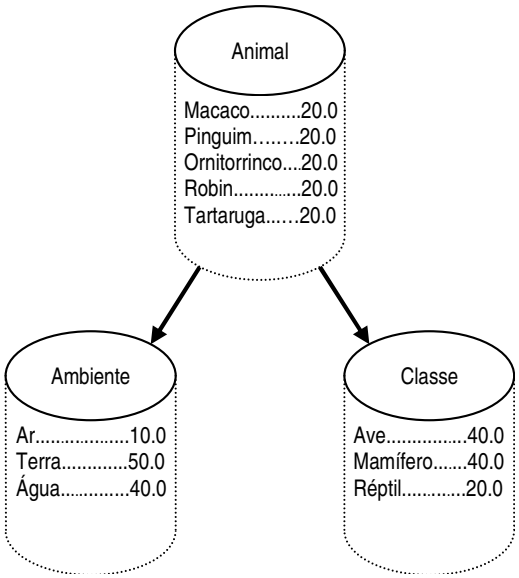


Figura 2.4: Exemplo simples de uso de Rede Bayesiana

Na Figura 2.4 pode-se observar um exemplo simples de Rede Bayesiana. A figura ilustra algumas características de animais, entre elas o ambiente em que eles vivem e a classe a que pertencem. No contexto desse exemplo, *Animal* representa uma

variável incondicional do modelo Bayesiano e macaco, pinguim, ornitorrinco, robin e tartaruga representam os possíveis estados dessa variável. A descrição dos estados das outras duas variáveis do exemplo (*Ambiente* e *Classe*) se dá de forma análoga.

Como mencionado anteriormente, para cada uma das três variáveis, existe uma CPT. Essas tabelas representam o relacionamento entre os estados de uma ou mais variáveis. Quanto maior o número de variáveis pais, maior será a CPT, pois a quantidade de linhas da tabela é proporcional ao número de combinações possíveis entre todos os estados (da própria variável e das variáveis pais). Por exemplo, como a variável *Animal*, representada pela Tabela 2.1, não possui nenhuma variável pai, sua tabela é simples e seus estados não dependem de outros. Isso não ocorre nas outras duas variáveis do exemplo (representadas pelas Tabelas 2.2 e 2.3), já que elas são variáveis condicionais desse modelo Bayesiano.

Tabela 2.1: Tabela de probabilidade condicional da variável Animal

Macaco	**Pinguim**	**Ornitorrinco**	**Robin**	**Tartaruga**
20	20	20	20	20

Detalhando um pouco mais o exemplo, a probabilidade da variável *Animal* pertencer a alguma das suas categorias é dada por $Pr(Animal) = 0.20$. Assim, a Tabela 2.1 representa uma variável do modelo Bayesiano que possui um conjunto de 100 animais distribuídos igualmente em 5 categorias. A construção dessa tabela se dá de forma simples e direta, pois a variável *Animal* não sofre influência de nenhuma outra variável e as categorias são uniformemente distribuídas. Dessa forma, podemos dizer que a probabilidade de um animal pertencer a uma categoria é 20%. Os valores mostrados nos estados da variável *Ambiente*, assim como os valores dos estados da variável *Classe*, ilustradas no exemplo da Figura 2.4, refletem o conhecimento *a priori* extraído das suas respectivas CTPs. Por exemplo: sabe-se que pinguins podem viver tanto na terra quanto na água, como está representado na terceira linha da Tabela 2.2. Do mesmo modo, a segunda linha da Tabela 2.3 está dizendo que todos os macacos existentes são mamíferos.

Tabela 2.2: Tabela de probabilidade condicional da variável Ambiente

Animal	**Ar**	**Terra**	**Água**
Macaco	0	100	0
Pinguim	0	50	50
Ornitorrinco	0	0	100
Robin	50	50	0
Tartaruga	0	50	50

Através do modelo apresentado na Figura 2.4, é possível demostrar como atribuir inferências através de redes Bayesianas. Desse modo, um exemplo de seu uso pode ser conduzido na forma de pergunta da seguinte maneira: *Alguém, durante um mergulho, observa um animal dentro da água, qual a probabilidade desse animal ser um Pinguim?* De acordo com o conhecimento *a priori*, sabe-se que: o observador estava

Tabela 2.3: Tabela de probabilidade condicional da variável Classe

Animal	**Ave**	**Mamífero**	**Réptil**
Macaco	0	100	0
Pinguim	100	0	0
Ornitorrinco	0	100	0
Robin	100	0	0
Tartaruga	0	0	100

na água ($Pr(Ambiente|Agua)$); pinguim e tartaruga vivem tanto na água quanto na terra ($Pr(Pinguim|Agua) = Pr(Tartaruga|Agua) = 0.5$) e; o ornitorrinco vive somente na água ($Pr(Ornitorrinco|Agua) = 1.0$).

O fato do mergulhador observar um animal dentro da água torna as informações, referentes à Terra e Ar, irrelevantes. Portanto, somente os valores da coluna *Água* da Tabela 2.2 são observados. Essas informações são buscadas diretamente na CTP da variável *Ambiente*, dada pela combinação de seus estados com todos os estados de suas variáveis pais (neste caso, *Animal*). O próximo passo depois dessas buscas, é realizar uma normalização dos valores, chegando a resposta que a probabilidade do animal observado ser um pinguim é 25% ($Pr(Animal|Pinguim) = 0.25$).

2.3 Trabalhos Relacionados

A área de estimativas de custos, no contexto de projetos de gerenciamento de TI, tem recebido grande atenção da comunidade científica nos últimos anos. Observa-se, contudo, que os esforços de pesquisa concentram-se em propor métodos para prever custos (recursos humanos/materiais e tempo) de projetos de *software*. Até onde sabemos, não apenas o escopo das investigações reside em projetos de natureza específica, como também relações entre fases do ciclo de vida de diferentes projetos são pouco exploradas. Diante deste panorama, discute-se a seguir trabalhos que foram identificados como mais correlatos, apesar de apresentarem objetivos distintos dos abordados nesta dissertação.

O *Constructive Cost Model* (COCOMO) (BOEHM, 1981) é um dos modelos de estimativas de custo mais citados em trabalhos científicos da área. Seu objetivo é estimar o tempo e o esforço que um projeto de *software* despenderá ao ser implementado. O COCOMO funciona a partir de um modelo de regressão simples, sendo baseado em atributos como, por exemplo, Pontos de Função (PF) e linhas de código. Sua formulação foi fundamentada no estudo de 63 projetos de *software* produzidos em torno do ano de 1981, estabelecendo estimativas estáticas (conforme o modelo de regressão proposto) e colocando sob suspeita a eficácia delas em projetos atuais. Outro ponto fraco desse modelo é que suas estimativas focam em complexidade de desenvolvimento, não realizando relação com outras fases de projetos como, por exemplo, testes e suporte. Mais recentemente, alguns modelos derivados, a exemplo de COCOMO II e COINCOMO (BOEHM; VALERDI, 2008), foram propostos com propósito de atualizar o modelo original, sem, contudo, violar suas características

nativas.

Mendes e Mosley (MENDES; MOSLEY, 2008) introduzem oito modelos Bayesianos, quatro criados dinamicamente (através das ferramentas *Hugin* e *PowerSoft*) e quatro criados por especialistas. A finalidade do trabalho foi realizar um estudo comparativo entre esses modelos para estimar o esforço de desenvolvimento em projetos *Web*. Embora os autores explorem vários modelos diferentes, a inexistência de variáveis de outras fases do ciclo de vida dos projetos como, por exemplo, variáveis das fases de teste e suporte, impossibilitam a estimativa de esforço de manutenção dos projetos.

O modelo ED^3M proposto por (HAIDER et al., 2008) permite, durante a fase de testes, estimar quantos defeitos um projeto de *software* apresentará depois de concluído. Esse modelo é baseado na Teoria das Probabilidades (*Estimation Theory*) e não necessita de conhecimento prévio como entrada, ou seja, não depende de dados históricos de projetos passados para realizar suas estimativas. Se por um lado não analisar históricos pode facilitar o processo de estimativas, por outro, os resultados obtidos podem não refletir precisamente situações reais, visto que o desempenho atual da organização pode ser visto como uma projeção de seus projetos passados. Além disso, as estimativas são calculadas somente com base em dados de testes, desconsiderando informações fundamentais provenientes de fases como, por exemplo, desenvolvimento e suporte.

Existem, também, trabalhos que, apesar de não tratarem diretamente da questão de estimativas de custos de suporte, devem receber atenção por representarem esforços nacionais na busca pela resolução de problemas ligados à grande área de gerenciamento de TI. Por exemplo, Cordeiro (CORDEIRO et al., 2009) propõe o uso de *templates* como um mecanismo para formalizar, preservar e reusar o conhecimento adquirido com mudanças em infraestruturas de TI, para reuso em requisições futuras. Lunardi (LUNARDI et al., 2009) aborda a problemática de alinhamento de planos de mudanças em infraestruturas de TI a objetivos/restrições de negócio, propondo algoritmos e estratégias para realizar associações adequadas de humanos a atividades dos referidos planos. Ainda nesse contexto, Wickboldt (WICKBOLDT et al., 2009) propõe um método de estimativa e análise automatizada de riscos em processos de gerenciamento de mudanças em infraestruturas de TI.

Outro conjunto de trabalhos merece breve discussão, seja por explorar técnicas alternativas de prever quanto um projeto de TI despenderá de esforço e tempo da equipe de desenvolvimento ou por apresentar outras abordagens para aplicação de redes Bayesianas. Verhoef (VERHOEF, 2002) apresenta em seu trabalho formas simples de estimar tempo de execução de um projeto de *software* e tamanho de uma equipe de desenvolvimento em organizações com baixo grau de maturidade. No que se refere ao emprego de redes Bayesianas, Fung *et al.* (FUNG et al., 2010) emprega essa técnica para modelar o compartilhamento de experiências coletivas sobre os alarmes disparados por diferentes sistemas de detecção de intrusão. Em outro trabalho, Adolfson *et al.* (ADOLFSON et al., 2007) utilizam redes Bayesianas para estimar dinamicamente vários índices econômicos como, por exemplo, taxa de desemprego, inflação e balança comercial de um país.

Em resumo, mesmo que o tópico a respeito de estimativas de custo tenha sido explorado em algumas investigações recentes, nenhum dos trabalhos citados permite prever custos de suporte em projetos de gerenciamento de TI a partir de informações produzidas nas fases de desenvolvimento/implantação e testes. Além disso, os trabalhos apresentados nessa seção usam abordagens simplistas (baseadas em um número limitado de variáveis) ou estáticas, abrindo mão da aprendizagem a partir de experiências passadas. Para tratar essas deficiências, no próximo capítulo é apresentada uma abordagem Bayesiana para gerar previsões de custos de suporte.

3 SOLUÇÃO PROPOSTA

Como mencionado no capítulo anterior, pouco tem sido feito para facilitar o processo de predição de custos de suporte em projetos de TI conduzidos pelas organizações. Para resolver esta questão, propomos uma solução conceitual para permitir a produção de estimativas de custos de suporte. Em contraste com pesquisas anteriores, nossa solução concentra-se no fornecimento de uma forma sistemática para projetar estimativas a partir de dados históricos sobre a fase de desenvolvimento/implantação e testes de projetos de TI.

A Seção 3.1 descreve a solução proposta. Os componentes da solução são detalhados e descritos a partir da Seção 3.2, com o modelo de informação utilizado. Na Seção 3.3 a metodologia de agregação e clusterização dos dados é detalhada e, por fim, na Seção 3.4 é apresentado o modelo Bayesiano empregado nessa dissertação.

3.1 Solução Conceitual

Nossa solução concentra-se em fornecer uma maneira sistemática para realizar estimativas de custo de suporte (ou, no sentido inverso, de desenvolvimento/implantação e/ou teste) a partir de dados históricos de projetos de gerenciamento de TI. A Figura 3.1 ilustra a base da nossa solução, destacando seus principais componentes conceituais, pessoal envolvido e interações.

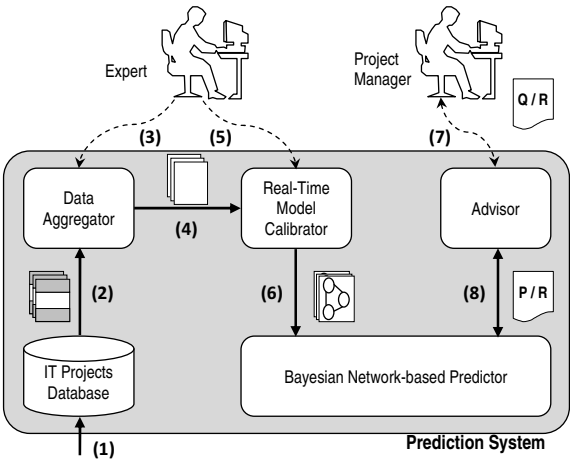


Figura 3.1: Elementos da solução proposta e suas interações

Dados brutos acerca de projetos de TI conduzidos na organização são continu-

amente coletados (fluxo 1 na Figura 3.1) e persistidos em uma base de projetos (*IT Projects Database*), seguindo modelo de informação a ser apresentado na Seção 3.2. Esses dados brutos de cada projeto são, então, processados pelo componente *Data Aggregator* visando à extração de informações-macro por projeto (ex: *Development/Test/SupportTime*), que são armazenadas em variáveis (fluxo 2). Esse processo de agregação pode ser apoiado pela figura de um especialista (fluxo 3), que pode determinar como popular as variáveis (privilegiando acurácia, eficiência, etc.) a partir dos dados brutos.

Como próximo passo da solução, os valores (das variáveis) associados aos projetos são repassados ao componente *Real-Time Model Calibrator* (fluxo 4 na Figura 3.1), este responsável por normalizar valores observados para cada variável e clusterizá-los em estados mutuamente exclusivos. A clusterização pode ser realizada de maneira automática ou semi-automática, nesse caso com intervenção do especialista para apoiar o processo (fluxo 5). Tanto o processo de agregação quanto o de calibragem são explicados com mais detalhes na Seção 3.3.

O resultado da referida clusterização é empregado como parâmetro de entrada do componente *Bayesian Network-based Predictor* (fluxo 6 na Figura 1). Esse componente modela as variáveis (nós) e suas relações causais (arcos) por meio de um grafo acíclico. Ademais, para cada variável Y que possui como antecessores X_1, ..., X_n, existe uma tabela P(Y | X_1, ..., X_n). Caso Y não possua um nó antecessor, a tabela de probabilidades é reduzida para uma probabilidade incondicional P(Y). Esse modelo Bayesiano, descrito com detalhe na Seção 3.4, permite que, fixadas as probabilidades de alguns nós (hipóteses), sejam computadas as demais probabilidades que se deseja (estimativas). Um gerente de projeto interessado em utilizar a solução para realizar estimativas interagirá com o componente *Advisor* (fluxo 7), *front-end* gráfico que contatará o motor de inferência (fluxo 8) repassando-lhe consultas do gerente de projetos (*i.e.*, conjunto de hipóteses) e retornando ao gerente estimativas computadas.

Tendo apresentado uma visão geral da solução proposta, as próximas seções têm por objetivo detalhar (*i*) o modelo de informação de projetos de TI; (*ii*) o processo de agregação e normalização de dados de projetos; e (*iii*) o modelo Bayesiano para gerar estimativas de custos de suporte.

3.2 Modelo de Informação de Projetos de TI

Conforme apresentado na solução conceitual, a previsão de custos de suporte requer acesso a informações detalhadas sobre o ciclo de vida (ex: composição das atividades, recursos envolvidos, tempo consumido) de um conjunto de projetos de TI realizados dentro da organização. Com o objetivo de representar essas informações, propõe-se um modelo que agrega classes oriundas do *Common Information Model* (CIM) (CIM, 2007) e da *Workflow Management Coalition Specification* (WfMC, 2007). O modelo também incorpora classes que materializam e mantém informações oriundas da observação de sistemas de acompanhamento de projetos de TI, como o *HP Quality Center* (HEWLETT-PACKARD, 2009). A Figura 3.2 ilustra uma visão

parcial do modelo proposto.

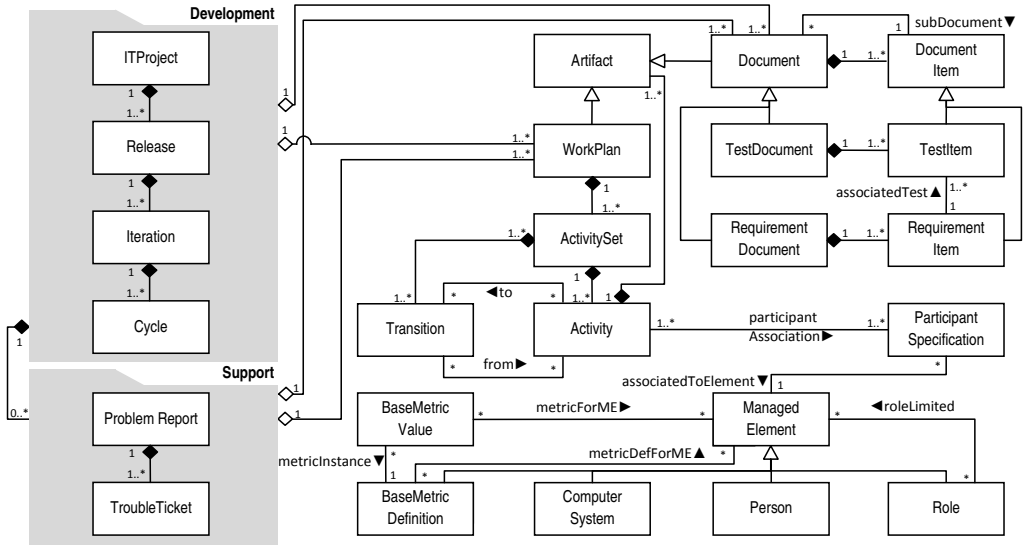


Figura 3.2: Visão parcial do modelo de informação para persistência de dados do ciclo de vida de projetos de gerenciamento de TI

Uma instância da classe *ITProject*, ponto de partida do modelo, representa um projeto, *i.e.*, um esforço temporário empregado para criar e/ou implantar um produto ou serviço encomendado a partir das necessidades do usuário (*i.e.*, cliente ou usuário final). Todo projeto de gerenciamento de TI pode ser entregue ao usuário por meio de uma ou mais *releases* (instâncias da classe *Release*). Cada *release* contém um conjunto de funcionalidades completamente desenvolvidas e testadas que quando validada pelo cliente, pode ser colocada em produção. Cada uma dessas funcionalidades pode ser concluída por meio de uma ou mais iterações (*Iteration*), e uma iteração é composta por um ou mais ciclos (*Cycle*). Os ciclos variam de acordo com a metodologia de desenvolvimento adotada, tais como levantamento de requisitos, análise e planejamento, desenvolvimento/implantação e teste. As classes recém descritas fazem parte do pacote *Development*, ilustrado na Figura 3.2.

Com o objetivo de organizar as atividades realizadas em cada ciclo, um ou mais planos (instâncias da classe *WorkPlan*) são definidos. Um plano é um *workflow* de atividades seguindo a definição proposta pela WfMC. Tomando como exemplo o ciclo de testes associado a uma dada iteração/*release* de um projeto de implantação de uma infraestrutura de rede sem fio, atividades (*Activity*) e transições (*Transition*) entre elas modelariam um plano (*WorkPlan*) ordenado de testes. Instâncias da classe *Activity* sempre possuem participantes associados. Esses são representados pela classe *ParticipantSpecification* e podem ser recursos humanos e/ou materiais empregados nas atividades. Tais recursos são mapeados a partir de classes do CIM. Ao mesmo tempo, atividades podem tanto produzir quanto consumir artefatos (*Artifact*). Exemplos de artefatos são documentos de requisitos (*RequirementDocument*) e de teste (*TestDocument*). Além de permitirem a documentação sistemática do projeto, esses artefatos também permitem o compartilhamento de informações entre fases do ciclo de vida. O modelo proposto também possui um conjunto de classes de

documentos que descrevem e organizam todo o projeto de TI. Lá são encontrados documentos que são consumidos e produzidos nas diversas etapas do ciclo de vida do desenvolvimento do projeto. No exemplo específico, itens de teste (documentados por meio de instâncias da classe *TestItem*) estão associados à validação de um ou mais requisitos (*RequirementItem*) a fim de registrar e descrever a cobertura dos testes envolvidos no projeto de TI.

Após o desenvolvimento e entrega do projeto, a fase de suporte, representada pelo pacote *Support* poderá ser instanciada em duas situações: a primeira ocorre quando algum tipo de falha é descoberto em algum dos itens de teste, e a segunda é quando um usuário encontra algum tipo de dificuldade em instalar ou utilizar a aplicação, serviço ou infraestrutura. O primeiro caso sugere a identificação de um erro no desenvolvimento de algum dos requisitos previamente elicitados na fase de planejamento e levantamento de requisitos. Essa natureza de problema leva a equipe de manutenção à criação de algum procedimento de contorno. O desenvolvimento de *patches* de correção pode ser visto como um novo projeto, pois possui seus próprios ciclos de análise, desenvolvimento e testes.

Já o segundo caso contemplado pela fase de suporte diz respeito à situação em que o usuário encontra algum problema na instalação ou manuseio do *software* ou infraestrutura de TI. Nestes casos o problema é tratado com a indicação do procedimento adequado pela equipe de suporte sem a necessidade da produção de nenhum artefato de *software* ou implantação de novos recursos materiais. Um documento indicando uma falha em algum teste de requisito dá entrada na fase de suporte. Esta requisição de suporte é direcionada à *ProblemReport* que analisa a chamada e faz a documentação. Se o suporte for solucionado com apenas a indicação de algum procedimento correto, a documentação é suficiente para dar fim à chamada. Se houver a necessidade do desenvolvimento e implementação de uma solução, então é instanciado um ou mais *TroubleTickets*, conforme o grau de dificuldade e complexidade da solução adotada para sanar a falha.

3.3 Agregação e Clusterização de Dados de Projetos

Como já mencionado, a sumarização de dados de projetos em variáveis é realizada pelo componente *Data Aggregator*. Ainda que a solução conceitual não tenha por objetivo fixar um conjunto único de variáveis, algumas são intuitivamente importantes para as estimativas propostas neste trabalho. Cita-se, como exemplo, tamanho do projeto e tamanho das equipes de desenvolvimento/implantação, teste e suporte. O processo de agregação consiste em percorrer cada instância de projeto e calcular, observando diversos objetos do modelo, valores a essas variáveis. No caso específico da variável tamanho da equipe de suporte, por exemplo, é necessário percorrer todas as atividades (instâncias da classe *Activity*) dos planos (*WorkPlan*) vinculados a bilhetes de cada projeto (*TroubleTicket*) e contabilizar todos os humanos envolvidos (*Person*). Já no caso da variável tempo despendido na fase de testes, por exemplo, é preciso percorrer todas as atividades dos planos de testes vinculados aos documentos de testes e seus respectivos requisitos e, então, contabilizar todo

o tempo gasto em cada uma dessas atividades. Demais variáveis são valoradas de forma análoga.

Em complementação ao processo de agregação, o componente *Real Time Model Calibrator*, após os valores já estarem associados às variáveis de cada projeto, executa (para cada variável encontrada) um procedimento de clusterização desses valores. Como resultado, cada variável passa a ser representada por um conjunto de estados (modelando diferentes naturezas ou grandezas). Voltando ao exemplo da variável tamanho da equipe de suporte, o procedimento poderia gerar, como resultado, os estados *pequeno*, *médio* e *grande*. Vale lembrar que o procedimento de clusterização pode ser conduzido usando algoritmo para esse fim, como *k-means*, *fuzzy c-means* e classificação *naive Bayes* (DUDA; HART; STORK, 2001) ou valendo-se da experiência de um especialista. No contexto desse trabalho, o processo de clusterização deu-se através do algoritmo *k-means*, apoiado por meio de opiniões de especialisas na área de projetos de TI durante a escolha do centríolo de cada cluster.

Como resultado do procedimento de clusterização, tem-se as variáveis do modelo de estimativa devidamente alimentadas (representando o conhecimento *a priori*). Ainda como parte do processo de calibragem, analisa-se relações existentes entre as variáveis e estabelece-se um grafo de influências, podendo este ser entendido como o esqueleto do modelo Bayesiano de estimativa. Esse modelo é explicado na próxima seção.

3.4 Modelo Bayesiano

O modelo Bayesiano para estimativas de custos de suporte (ou desenvolvimento/ implantação e teste) proposto neste trabalho é composto por dezesseis variáveis, organizadas em três grupos: *desenvolvimento/implantação*, *teste* e *suporte*. A Figura 3.3(a) ilustra o modelo, destacando esses grupos em cinza claro, cinza escuro e branco, respectivamente. Variáveis como *StaffProd*, *StaffSize*, *ProjectSize* e *Time* são representadas nos três grupos, enquanto outras, dada sua especificidade, aparecem apenas em um ou outro grupo (ex: *TestCoverage*). As ligações entre as variáveis, representadas na figura por meio de arcos orientados, expressam relações causais entre elas. Lembrando, tais ligações são estabelecidas no processo de calibragem.

A configuração do modelo recém mencionado foi delineada tendo como base (*i*) opiniões de especialistas, (*ii*) análise de uma base com mais de 5.000 projetos disponibilizados pelo *International Software Benchmarking Standards Group* (ISBSG) (ISBSG, 2007) e (*iii*) observação de variáveis empregadas em investigações correlatas (MENDES; MOSLEY, 2008). Ressalta-se, ainda, que o modelo é aberto, podendo ser *simplificado*, priorizando reduzir processamento, ou *estendido*, visando à captura de aspectos (variáveis e relações) não contemplados.

Voltando ao modelo, as variáveis incondicionais, populadas no processo de calibragem, influenciam direta ou indiretamente o cálculo de probabilidades dos estados das variáveis condicionais. Este é o caso, por exemplo, da variável *SupProjectSize* em relação à variável *SupStaffSize*. Como pode ser observado na Figura 3.3(b),

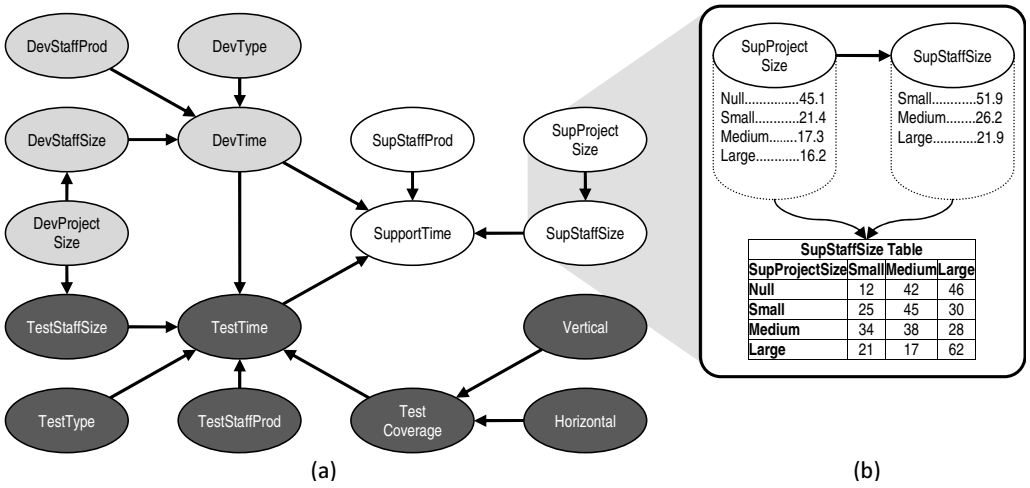


Figura 3.3: Modelo Bayesiano para estimativa de custos de desenvolvimento / implantação, teste e suporte

SupStaffSize modela uma tabela de probabilidades que associa os seus estados com os de *SupProjectSize* (variável antecessora). Uma célula da tabela representa a probabilidade de *SupStaffSize* (X=pequeno, médio, grande) ocorrer em *SupProjectSize* (Y=*null*, pequeno, médio, grande). De forma análoga, as demais variáveis condicionais possuem suas próprias tabelas.

Com o modelo devidamente populado, é possível realizar consultas *what-if* (fixando estados de uma ou mais variáveis) e obter estimativas (observando o efeito provocado nas demais). No contexto do trabalho, passa a ser possível responder perguntas como: *conhecido tempo/esforço despendidos em desenvolvimento/implantação de um projeto, qual a expectativa de custos de suporte?* Ou, ainda, *quanto investir em desenvolvimento/implantação e teste dado um limite máximo tolerável de custos de suporte?* As estimativas produzidas pelo modelo serão probabilidades associadas aos estados das variáveis, e essas respostas refletem a probabilidade da incidência de certo evento ocorrer em relação ao número total de eventos disponíveis na base de dados, levando em consideração o cenário (relações causais) em que este evento está inserido.

4 SISTEMA $UPPORT

A solução para a estimativa de custos de suporte em projetos de TI, apresentada no capítulo anterior, é materializada pelo protótipo de um sistema de predição de custos, tempo e esforço de desenvolvimento/implantação, de testes e de suporte, denominado $UPPORT. A seguir serão detalhados alguns dos principais aspectos da solução proposta com a implementação dos componentes conceituais apresentados no Capítulo 3, seguido do exemplo de uso do sistema.

O sistema $UPPORT materializa as funcionalidades dos componentes *Data Aggregator*, *Real-Time Model Calibrator*, *Bayesian Network-based Predictor* e *Advisor* (destacados na Figura 3.1). A partir das especificações recebidas, o sistema gera como saída estimativas detalhando as variáveis que o usuário escolheu deixar em aberto. Neste capítulo será descrito o sistema $UPPORT, mais especificamente (*i*) as tecnologias envolvidas durante seu desenvolvimento e implementação, e (*ii*) a interface gráfica implementada para facilitar a utilização e a interação com o usuário.

4.1 Tecnologias Envolvidas

O sistema $UPPORT consiste em uma implementação prototípica da solução apresentada na capítulo anterior. Desenvolvido em Java com auxílio de uma API chamada *JavaBayes*, o sistema permite realizar estimativas acerca de custos, esforço e/ou tempo de uma dada fase do ciclo de vida de um projeto de TI.

JavaBayes (COZMAN, 2001) possui um conjunto de ferramentas e algoritmos que tornam possível a realização de rotinas com redes Bayesianas, entre elas: cálculo de probabilidades e expectativas, análise de robustez, testes *what-if*. Além disso, *JavaBayes* também permite ao usuário criar, modificar, importar e exportar modelos de redes.

4.2 Interface Gráfica do Sistema $upport

Com o objetivo de facilitar a condução do processo de estimativas de custos, tempo e esforço de projetos de TI, foi implementada uma interface gráfica amigável ao usuário utilizando a biblioteca Java/Swing GUI Builder (LOY; ECKSTEIN, 2002). A Figura 4.1 ilustra a aba *Profiles*, na qual pode-se observar a tela inicial do Sistema $UPPORT. Neste momento, o programa encontra-se sem nenhum

profile carregado. No contexto dessa dissertação, um *profile* representa um conjunto de projetos de TI semelhantes, ou seja, um grupo de projetos da mesma natureza e executados por uma única organização. Vale lembrar que todos os *profiles* de projetos carregados são descritos de acordo com o modelo de informação apresentado na Seção 3.2, sendo organizado em *releases*, *iterações* e *ciclos*.

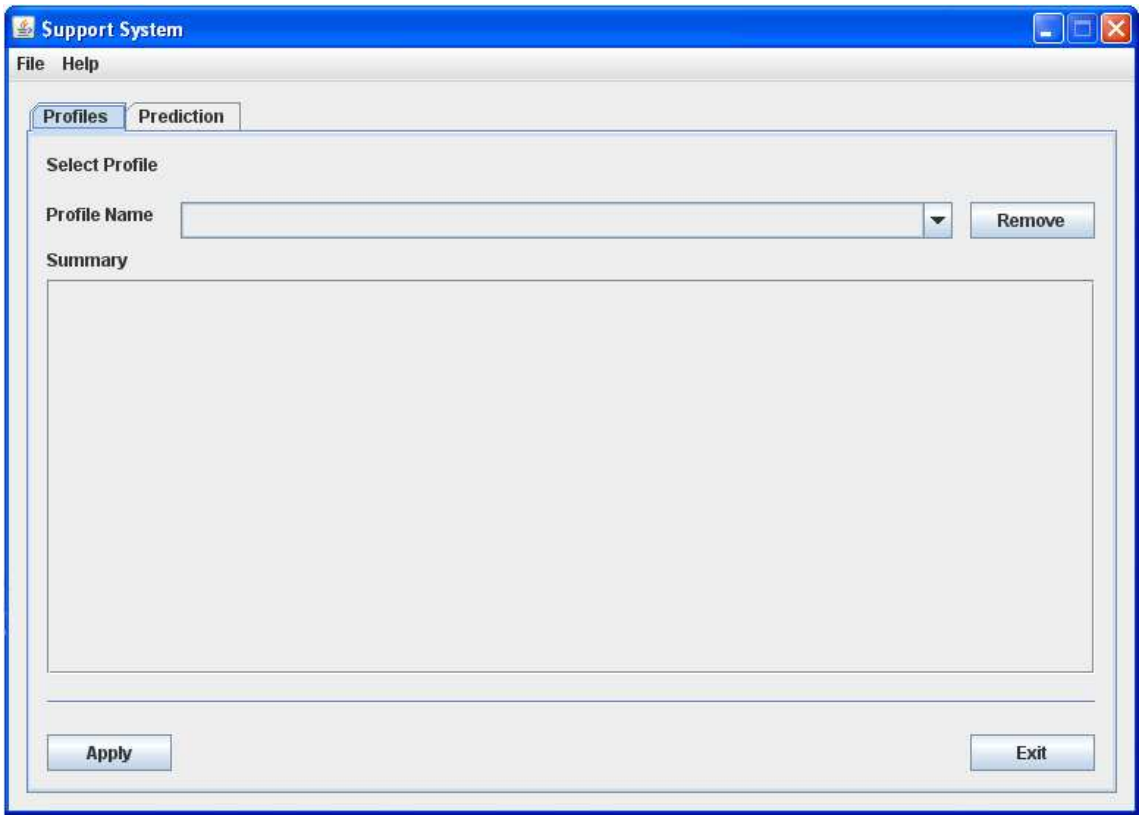


Figura 4.1: Interface gráfica do sistema $UPPORT

A Figura 4.2 descreve o procedimento correto que deve ser seguido para importar um *profile*. O *profile* desejado é selecionado e todas as variáveis com seus detalhes de ciclos de desenvolvimento/implantação, teste e suporte são automaticamente carregadas.

Um resumo das variáveis e seus estados podem ser observados na Figura 4.3. Essa figura ilustra o comportamento esperado quando um *profile* é carregado no sistema. A ideia é que todos esses dados possam ser importados de sistemas como *HP Quality Center*, sem (ou com pouca) intervenção do usuário. De posse dessas informações, $UPPORT tem condições de extrair, do projeto em análise, valores (no caso, informações já computadas) para estados de algumas das variáveis do modelo. Porém, nem todas as variáveis presentes no projeto de TI são conhecidas ou podem ser deduzidas facilmente. Alternativamente (ou em complemento), o usuário pode informar, diretamente na aba *Prediction*, valores arbitrários (nesse caso, hipóteses) a estados das variáveis do projeto. Tal é ilustrado na Figura 4.4, onde, para o projeto em análise, cria-se a hipótese de que o tempo de desenvolvimento/implantação é *médio*. Como resultado, o sistema $UPPORT recalcula as probabilidades dos estados das variáveis (do modelo) que estão em aberto e estima uma faixa de valor em que

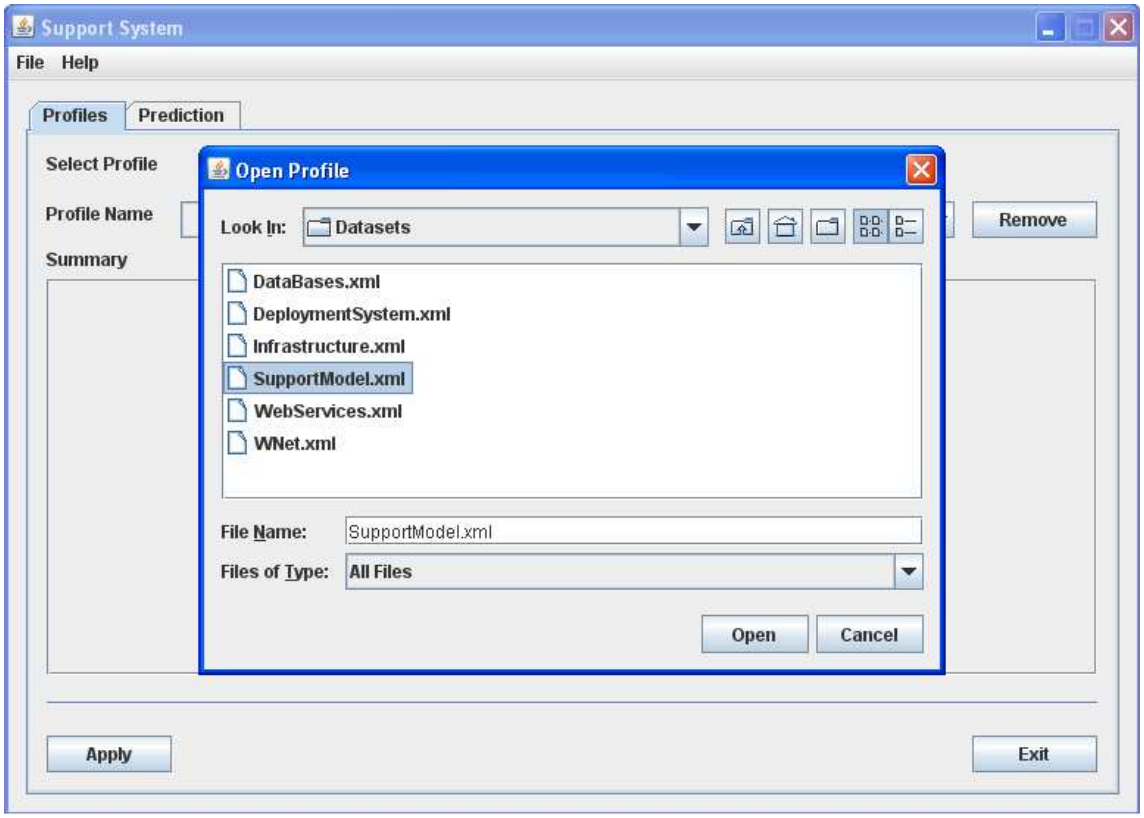


Figura 4.2: Importação de um perfil

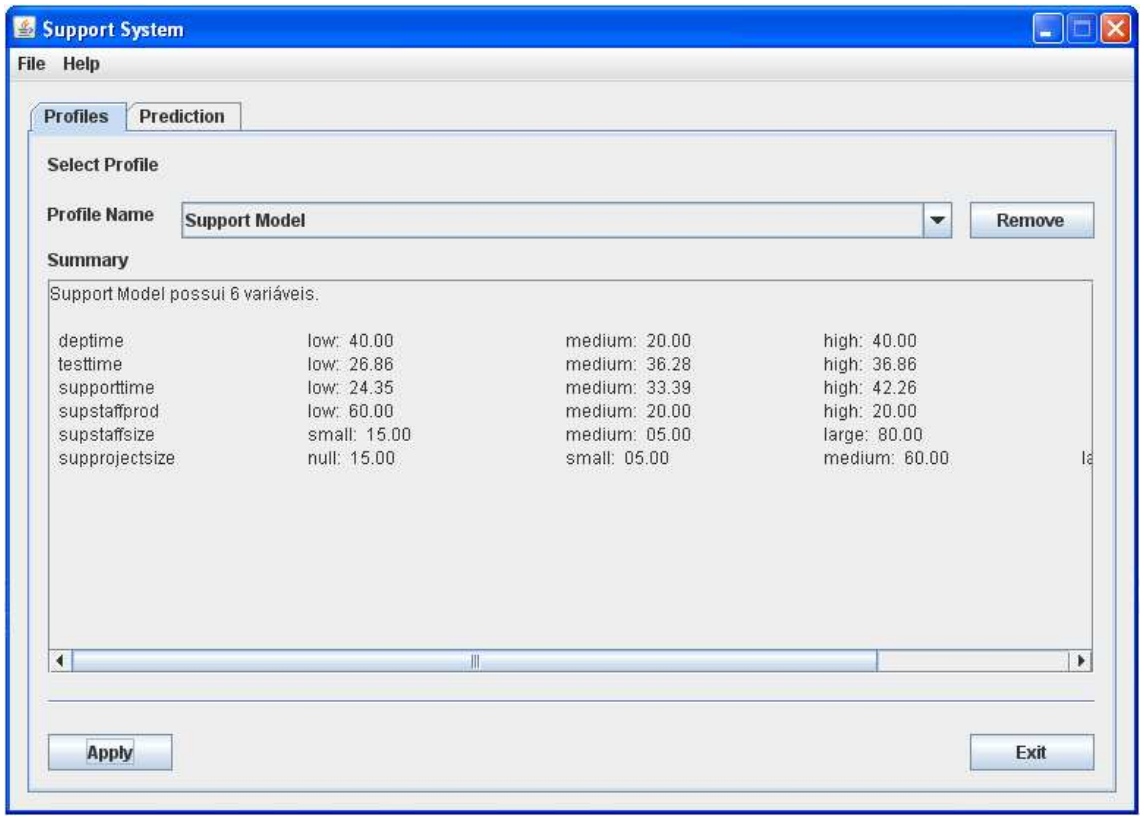


Figura 4.3: Aplicação de um perfil e visualização de suas variáveis

essa variável está enquadrada.

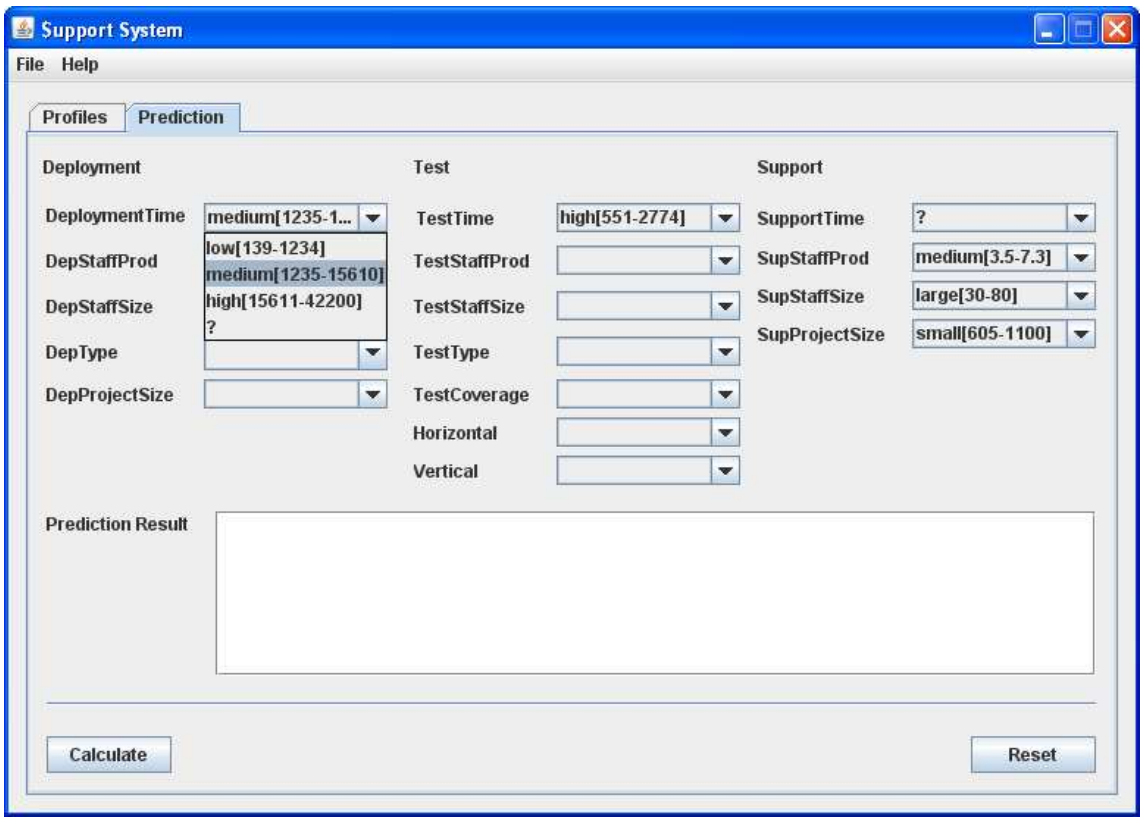


Figura 4.4: Configuração de questões *what-if*

Conforme pode ser visto na Figura 4.4, algumas variáveis foram deixadas em branco (*e.g. DepProjectSize*, *TestType*, *Horizontal*, *Vertical*, etc.). Isso ocorre pelo fato delas não estarem presentes no perfil de projetos escolhido pelo usuário. Por exemplo, considere a variável *TestType* de um perfil de projetos de implantação de redes sem fio executado por uma organização. Imagine que todos os testes de funcionalidade das redes sem fio foram realizados através de *scripts*, ou seja, de forma automática. Dessa maneira, considerando a base de dados histórica de projetos de TI dessa organização, a variável *TestType* não possui instâncias de diferentes tipos e, portanto, seu uso é dispensável para o cálculo de estimativas para um novo projeto. Em outras palavras, essas variáveis deixadas em branco não vão influenciar em nada no processo de estimativa executado pelo sistema \$UPPORT, pois elas carregam uma informação constante que não varia entre as diferentes instâncias de projetos proveniente do histórico.

O sistema \$UPPORT apresenta, como resultado, estimativas baseadas no modelo Bayesiano devidamente alimentado com informações provenientes de uma base de dados de projetos de TI executados no passado (como explicado no Capítulo 3). Um exemplo da resposta produzida pelo sistema é apresentado na Figura 4.5, onde é informado que, com 92,31% de chance, o tempo a ser despendido em suporte no projeto (WNet) será *pequeno*. Em complemento, o sistema também oferece uma faixa de valores associada ao estado de maior probabilidade. Como pode ser visto nesse caso, a resposta da predição é interpretada da seguinte maneira: o provável

tempo que a equipe de suporte despenderá na manutenção desse projeto de TI está dentro do intervalo de 110 - 276 horas.

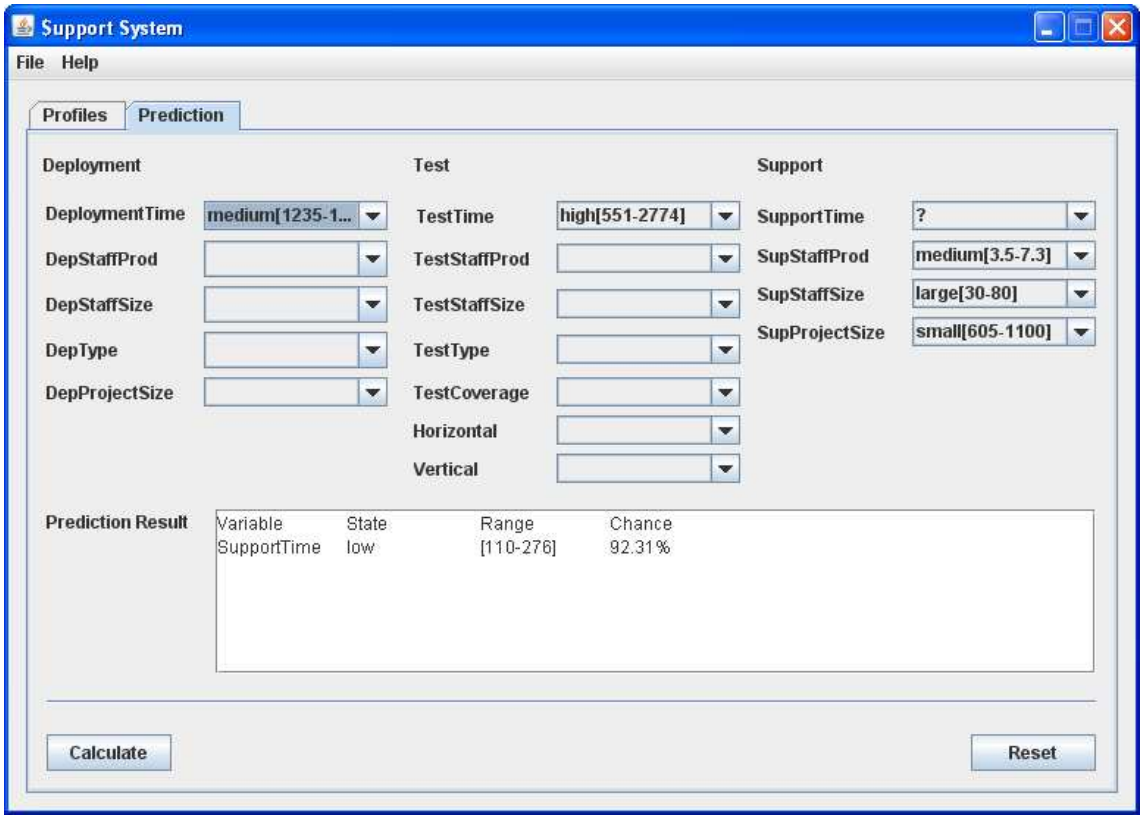


Figura 4.5: Resultado da estimativa

5 AVALIAÇÃO DA SOLUÇÃO PROPOSTA

Para provar conceito e viabilidade técnica da solução proposta considerou-se, como estudo de caso, a predição de custos associados com projetos de implantação de infraestrutura de redes sem fio (também referenciados como WNet ao longo deste capítulo). Esse estudo de caso foi escolhido uma vez que este tipo de projeto de TI é largamente instanciado e possui forte relação com o tema de gerenciamento de redes e serviços. Durante a avaliação o estudo de caso foi conduzido utilizando o sistema $UPPORT apresentado no capítulo anterior. Esse sistema foi utilizado tanto para gerar uma avaliação qualitativa quanto quantitativa da nossa solução.

A seguir, na Seção 5.1 é apresentada a metodologia utilizada no estudo de caso. O estudo de caso presente na avaliação é observado sobre duas perspectivas. Na primeira, os resultados obtidos através do sistema $UPPORT são confrontados com a opinião de especialistas em projetos de implantação de redes de computadores. Na segunda perspectiva, esses mesmos resultados foram comparados com os produzidos por outro modelo Bayesiano, criado de forma automática pelo *software* Genie 2.0 (Genie and Smile Systems, 2010). A avaliação descrita neste capítulo também conta com uma análise qualitativa, uma análise quantitativa e uma análise de sensibilidade, que serão detalhadas nas Subseções 5.2.1, 5.2.2 e 5.2.3 respectivamente.

5.1 Metodologia

A avaliação foi realizada por meio de duas abordagens distintas, mas complementares. Em ambas as abordagens, um conjunto de perguntas *what-if* sobre a instanciação de projetos WNet foi elaborado e submetido ao sistema $UPPORT, a fim de obter como resultado a estimativa de uma variável deixada como indefinida. Na primeira abordagem, os resultados obtidos foram comparados com a opinião de um segundo grupo de especialistas em projetos de TI (diferente do grupo que auxiliou na classificação dos projetos, mencionado na Subseção 3.4). Já na segunda abordagem, os resultados obtidos pela nossa solução foram comparados aos produzidos por um modelo Bayesiano criado automaticamente por meio do *software* Genie 2.0 (Genie and Smile Systems, 2010).

Os cenários pertencentes ao caso de estudo utilizado na avaliação ao longo do capítulo dizem respeito a projetos de redes de computadores sem fio. Vários outros ambientes poderiam ser explorados como, por exemplo, instalação de servidores,

migração entre sistemas ou desenvolvimento de *software*. Esta natureza de projeto foi escolhida pelo fato de ser um projeto tipicamente instanciado em organizações que fazem uso de uma infraestrutura de TI. Além disso, outro aspecto favorável foi a facilidade em encontrar especialistas dessa área que disponibilizassem informações úteis utilizadas durante a avaliação da solução.

Dessa forma, a obtenção e a preparação da base de dados de projetos tornou-se uma tarefa importante da avaliação. A partir dos dados do *dataset* do ISBSG, foram selecionados mais de 450 projetos de nosso interesse, para extrair parâmetros-chave, tais como faixas de valores reais para as variáveis do modelo (por exemplo, *DevTime*, *TestTime* e *SupportTime*). Além disso, olhando para esses valores (de cada projeto), foi possível clusterizá-los e consolidar os estados das variáveis. A Tabela 5.1 resume o resultado desta etapa. Usando *DevTime* (tempo de desenvolvimento/implementação) como exemplo, essa variável foi organizada em três estados: *baixo* [0 - 1.234 horas], *médio* [1.235 - 15.610 horas] e *alto* maior ou igual a 15.611 horas.

Tabela 5.1: Variáveis do modelo Bayesiano e seus estados correspondentes

Variável	**Estados***
DevTime	low [0-1.234], medium [1.235-15.610], high [≥ 15.611]
TestTime	low [0-300], medium [301-550], high [≥ 551]
SupStaffSize	small [1-8], medium [9-29], large [≥ 30]
SupStaffProd	low [0-3,4], medium [3,5-7,3], high [≥ 7,4]
SupportTime	low [0-276], medium [277-800], high [≥ 801]
DevStaffProd	low, medium, high
DevType	innovation, improvement, workaday
DevStaffSize	small, medium, large
DevProjectSize	small, medium, large
TestStaffProd	low, medium, high
TestType	manual, automatic
TestStaffSize	small, medium, large
TestCoverage	small, medium, large
Vertical	low, medium, high
Horizontal	low, medium, high
SupProjectSize	null(0), small, medium, large

* *DevTime*, *TestTime*, e *SupportTime* são medidos em horas. *SupStaffSize* é medido em número de humanos, e *SupStaffProd* em ManPower/hora.

Idealmente, o objetivo era usar os projetos disponíveis no *dataset* do ISBSG como entrada para o modelo Bayesiano. No entanto, tal não foi possível porque a maioria dos projetos tinha um conjunto incompleto de dados disponíveis (valores das variáveis) e também porque os projetos de TI não estavam relacionados com suas respectivas fases de suporte. Para superar essa limitação, foi gerada uma base de dados sintética de projetos agrupando os dados reais conhecidos com informações

fornecidas por um grupo de especialistas na gerência de projetos de TI, como explicado a seguir. Primeiro, foram selecionadas cinco variáveis (*DevTime*, *TestTime*, *SupStaffSize*, *SupStaffProd* e *SupportTime*) e gerada uma lista com todas as combinações possíveis entre seus estados, desde *baixo*, *baixo*, *pequena*, *baixo*, *baixo* até *alta*, *alta*, *grande*, *alta*, *alta*. Cada combinação representa uma instância de projeto de TI, também chamada de *cenário* ao longo deste trabalho. Na sequência, foi solicitado aos especialistas que, com base em sua experiência, classificassem cada um dos cenários como *provável*, *possível* ou *improvável* e, em seguida, foram criados três conjuntos de projetos distintos conforme essa classificação. Finalmente, foi escolhida uma proporção variada de cenários a partir desses conjuntos, dependendo do perfil desejado da base de dados e, então, povoadas tanto as variáveis incondicionais, quanto as variáveis condicionais do modelo Bayesiano.

A fim de analisar a precisão das predições, o modelo Bayesiano foi populado com informações coletadas a partir de 710 cenários de projetos, na proporção de 70% (provável), 20% (possível) e 10% (improvável). Esse perfil representa o caso em que 70% dos projetos conduzidos por uma organização apresenta um comportamento esperado, enquanto os outros 30% se comportam de maneira irregular e, por vezes, de forma anormal. Essas proporções foram propostas através do estudo analítico do *dataset* do ISBSG e confirmadas pelos especialistas.

A avaliação da solução proposta foi conduzida por meio de 50 perguntas submetidas ao sistema $UPPORT. Na primeira abordagem da avaliação, os resultados obtidos foram confrontados com a opinião de um grupo de especialistas. Na segunda abordagem, por outro lado, os resultados foram comparados com os produzidos por um novo modelo Bayesiano gerado automaticamente, de modo *bottom-up*, a partir da base de dados. O novo modelo, ilustrado na Figura 5.1, foi gerado com o *software* Genie 2.0 (Genie and Smile Systems, 2010), selecionando-se o algoritmo PC (SPIRTES; GLYMOUR; SCHEINES, 2000). Em comparação com o modelo que propomos, esse introduz uma relação causal entre as variáveis *DevTime* e *SupStaffProd*. Além disso, algumas relações, destacadas em cinza na figura, são redefinidas.

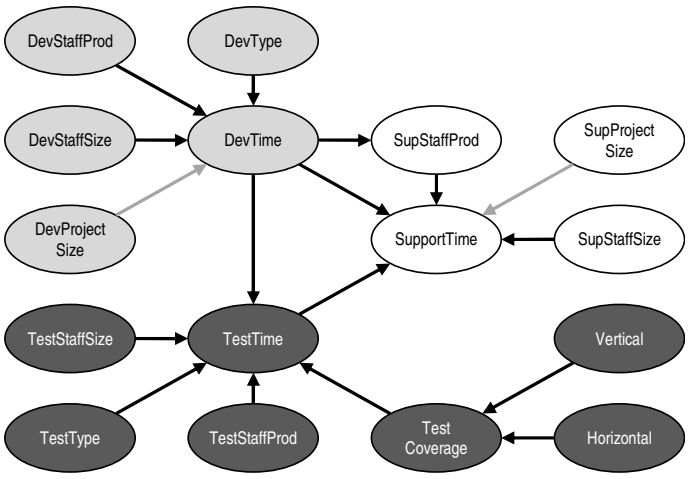


Figura 5.1: Modelo Bayesiano criado automaticamente através do Genie 2.0

5.2 Resultados e Discussão

A apresentação dos resultados obtidos é organizada da seguinte forma: primeiramente a precisão das predições de perguntas *what-if* estimadas pelo sistema $UPPORT são reportadas e comparadas, tanto com as realizadas pelos especialistas, como as realizadas pelo modelo Bayesiano criado através do Genie (considerado *baseline* para a avaliação), caracterizando uma análise qualitativa da solução. Posteriormente, é apresentada uma avaliação de caráter quantitativo de cenários hipotéticos de projetos quando uma de suas variáveis é modificada experimentalmente. Toda a avaliação deu-se através da apresentação de questões *what-if* (tanto para a previsão dos custos esperados, quanto para avaliar as condições necessárias para a execução de um novo projeto). Por fim, foi analisada a sensibilidade da solução proposta através de estimativas elaboradas a partir de diferentes perfis de base de dados.

5.2.1 Predição de Questões *What-if*

A Tabela 5.2 ilustra um conjunto de 10 (do total de 50 escolhidos aleatoriamente) questões submetidas aos sistemas $UPPORT e Genie, bem como a um grupo de especialistas. Em cada uma dessas questões foi marcado um estado para cada uma das quatro variáveis, e uma variável, que deveria ser prevista, foi deixada em aberto. Para ilustrar, considere o Cenário 1. Neste cenário, a questão colocada foi: *qual é o tempo necessário de suporte para a manutenção dos ativos entregues por um projeto WNet específico, se para isso for despendido grande tempo para desenvolvimento e testes, e a organização possuir um grande número de humanos na equipe de suporte, cuja produtividade seja considerada média?* Em outras palavras: dado *DevTime* = *alto*, *TestTime* = *alto*, *SupStaffSize* = *grande* e *SupStaffProd* = *médio* para a implantação de um projeto WNet específico, qual é o estado esperado para *SupportTime*?

Tabela 5.2: Questões *what-if* para predição de custos de projetos WNet.

#	Variáveis					Resultado do $UPPORT	Opinião dos Especialistas	Resultado do Genie
	DevTime	TestTime	SupStaffSize	SupStaffProd	SupportTime			
1	high	high	large	medium	?	low	Concordo	low
2	high	?	medium	high	low	high	Concordo	high
3	medium	medium	?	medium	low	*Any*	Discordo	high
4	?	medium	medium	medium	medium	low	Concordo	medium
5	low	medium	large	?	high	medium	Discordo	low
6	medium	high	small	?	high	low	Concordo	low
7	high	medium	?	low	high	medium	Concordo	medium
8	low	?	large	medium	medium	medium	Concordo	medium
9	low	high	large	low	?	high	Concordo	high
10	?	high	large	medium	high	low	Concordo	low

Para a pergunta apresentada o sistema $UPPORT retornou como resultado o estado *baixo*. Isso significa que, para projeto WNet em questão, é esperado um baixo tempo de suporte após a entrega final dos seus ativos. Os especialistas encontraram

na atribuição de *baixo* para *SupportTime* uma boa estimativa. De forma análoga, o mesmo cenário foi experimentado no modelo Bayesiano criado automaticamente pelo *software* Genie. O resultado obtido confirmou as respostas do sistema $UPPORT e dos especialistas.

Agora considerando o Cenário 3, cuja pergunta foi: *qual deve ser o tamanho da equipe de suporte, se houver uma quantidade média de tempo disponível para o desenvolvimento do projeto, quantidade média de tempo destinado para a fase de testes, uma equipe de suporte com produtividade considerada média, e deseja-se gastar tempo baixo com a fase de suporte?* Dito de outro modo: dado *DevTime = médio*, *TestTime = médio*, *SupStaffProd = médio* e *SupportTime = baixo* para a execução de um projeto WNet, qual o estado esperado para *SupStaffSize*?

O sistema $UPPORT retornou como resultado a mesma probabilidade para todos os estados. Esse resultado indica que, considerando as circunstâncias declaradas na pergunta apresentada e o histórico de projetos, cada estado possui 33,33% de chance de ocorrer. No entanto, os especialistas discordaram da estimativa fornecida pelo sistema, que, embora factível com o histórico de projetos, não representa uma boa resposta para a questão. Quando avaliado pelo modelo Bayesiano produzido pelo Genie, o cenário apontou *SupStaffSize = alto*, estimativa também discordante daquela produzida por $UPPORT.

Entre os cenários expostos na Tabela 5.2, pode-se notar que a nossa solução foi capaz de estimar precisamente os resultados de oito entre dez perguntas (1, 2, 4, 6, 7, 8, 9 e 10) pelo ponto de vista dos especialistas em projetos de TI. Já se comparada com os resultados obtidos com o Genie, nossa solução foi capaz de estimar corretamente os resultados de sete perguntas (1, 2, 6, 7, 8, 9 e 10). Observando os resultados por uma perspectiva mais ampla, para as 50 consultas *what-if* analisadas, o sistema $UPPORT foi capaz de estimar com precisão os custos previstos e as condições necessárias para a execução de projetos WNet em 82% dos cenários especificados, segundo a opinião dos especialistas. Pela visão do modelo Bayesiano gerado, $UPPORT apresentou um índice de acerto de 74%.

5.2.2 Avaliação de Cenários Hipotéticos

A Tabela 5.3 apresenta alguns cenários hipotéticos - em relação à execução de projetos WNet - submetidos ao sistema $UPPORT a fim de avaliar a sua viabilidade e probabilidade de ocorrência em instâncias reais. Para ilustrar, considere a hipótese do Cenário 6. O que se quer avaliar nesse caso é se é razoável que uma instância de um projeto WNet – executada considerando um valor médio de tempo para a fase de desenvolvimento, uma quantidade alta de tempo para a fase de teste, uma equipe de suporte de tamanho médio, e uma quantidade elevada de horas de suporte – exija que a produtividade de cada membro da equipe de suporte seja *alta*. Seguindo esta configuração, de acordo com o sistema $UPPORT, a probabilidade de tal cenário realmente ocorrer seria 55,56% – fato que foi considerado *possível* de acordo com o parecer dos especialistas consultados. A Tabela 5.4 contém as opiniões dos especialistas referentes aos outros cenários apresentados na Tabela 5.3. Ainda falando sobre o cenário 6, o mesmo também foi submetido ao modelo criado

automaticamente pelo Genie, a resposta dada por esse modelo foi *correto* e, da mesma forma, a Tabela 5.4 ilustra os demais resultados encontrados frente aos outros cenários apresentados na Tabela 5.3.

Tabela 5.3: Exemplo de predições obtidas a partir do sistema $UPPORT

#	Variáveis					Predição (%)*		
	DevTime	TestTime	SupStaffSize	SupStaffProd	SupportTime	low	medium	high
1	high	high	large	medium	low	83.33	04.76	11.90
2	high	high	medium	high	low	05.13	05.13	89.74
3	medium	medium	small	medium	low	33.33	33.33	33.33
4	low	medium	medium	medium	medium	48.61	48.61	02.78
5	low	medium	large	high	high	22.22	22.22	55.56
6	medium	high	medium	high	high	22.22	22.22	55.56
7	high	medium	medium	low	high	16.67	41.67	41.67
8	low	high	large	medium	medium	16.67	41.67	41.67
9	low	high	large	low	medium	41.67	16.67	41.67
10	high	high	large	medium	high	33.33	33.33	33.33

*Small/Medium/Large, se *SupStaffSize* está sendo considerada para a predição, e Low/Medium/High para os outras.

Também é possível observar na Tabela 5.3 as variações de probabilidades entre os estados das hipóteses do cenário 6. Considere, por exemplo, que a produtividade média de cada membro da equipe de suporte é *baixa*, em vez de *alta* e, sob essa nova configuração, sabemos que essa hipótese tem chance de 22,22% de ocorrer.

Tabela 5.4: Parecer dos especialistas sobre um conjunto de cenários

#	Variáveis					Opinião dos Especialistas	Resultado do Genie
	DevTime	TestTime	SupStaffSize	SupStaffProd	SupportTime		
1	high	high	large	medium	low	Provável	Correto
2	high	high	medium	high	low	Provável	Correto
3	medium	medium	small	medium	low	Provável	Incorreto
4	low	medium	medium	medium	medium	Provável	Correto
5	low	medium	large	high	high	Possível	Correto
6	medium	high	medium	high	high	Possível	Correto
7	high	medium	medium	low	high	Possível	Correto
8	low	high	large	medium	medium	Possível	Correto
9	low	high	large	low	medium	Improvável	Correto
10	high	high	large	medium	high	Improvável	Inconclusivo

Agora considere o caso do Cenário 10, cuja hipótese é: *um projeto WNet – com uma grande quantidade de tempo disponível para desenvolvimento e teste, e um tamanho grande para equipe de suporte com sua produtividade classificada como média – deverá exigir uma elevada quantidade de horas destinada a fase de suporte para a manutenção dos bens entregues (durante seu tempo de vida útil).* O sistema

$UPPORT informou que tal cenário hipotético tem chance de 33,33% de ocorrer, enquanto os especialistas constataram que o mais intuitivo seria que este cenário fosse considerado como *improvável* e, consequentemente, essa previsão é considerada inconclusiva (já que este cenário não é provável nem improvável).

Em resumo, o sistema $UPPORT foi capaz de apresentar avaliações precisas para 8 das 10 hipóteses descritas na Tabela 5.3. Em um dos casos (cenário 3), o sistema retornou uma avaliação errônea (em relação à opinião dos especialistas), enquanto no outro caso (Cenário 10), a avaliação não foi considerada conclusiva.

5.2.3 Análise de Sensibilidade

Quanto à avaliação da sensibilidade do modelo, foram testados outros perfis de base de dados com proporções variadas de cenários. Os experimentos discutidos anteriormente neste capítulo foram realizadas considerando o perfil do conjunto de dados 70% (provável), 20% (possível) e 10% (improvável). Essas porcentagens refletem o comportamento de um conjunto de dados reais (de acordo com a opinião de especialistas e confirmado por meio de um estudo aprofundado do conjunto de dados ISBSG). Uma vez efetuada a avaliação experimental considerando tal conjunto de dados, repetimos os mesmos experimentos com outros conjuntos, que eram essencialmente caracterizados por proporções variáveis de projetos prováveis, possíveis e improváveis. O objetivo dessa avaliação foi observar o quão sensível é a precisão das estimativas obtidas com o sistema $UPPORT frente a diferentes perfis de conjuntos de dados.

A Tabela 5.5 ilustra a taxa de sucesso que $UPPORT obteve quando alimentado com diferentes perfis de bases de dados. Como se pode notar, as avaliações dos cenários considerando um perfil de base de dados onde a maioria dos projetos apresentou padrão semelhante de desenvolvimento/implantação resultou em estimativas mais precisas. Este é o caso do perfil 70%-20%-10%, o que proporcionou a maior taxa de sucesso (de 82% dos 50 cenários avaliados).

Tabela 5.5: Precisão obtida com outros perfis de bases de dados

Perfil de dados	Correto (%)	Incorreto (%)	Inconclusivo (%)
33-33-33	40	24	36
50-30-20	60	24	16
70-20-10	82	02	16

Em contraste, as avaliações com base em perfis mais amorfos (ou seja, perfis que possuem projetos com alto nível de disparidade no valor de suas variáveis) levaram ao aumento do número de estimativas incorretas. Este é o caso dos perfis de 50%-30%-20% (60% de taxa de sucesso) e 33%-33%-33% (com somente 40% de taxa de sucesso). Essas observações provam que o número de estimativas inconclusivas (aquelas que não oferecem qualquer informação relevante para o gerente que utiliza o sistema de previsão) crescem proporcionalmente ao aumento de número de projetos fora de padrão. Por outro lado, resultados mais precisos podem ser alcançados por

perfis que exibem uma maior taxa de projetos semelhantes.

Como constatação geral da avaliação, os resultados produzidos pela solução proposta nesta dissertação são bastante satisfatórios. Mesmo com uma base contendo 30% de instâncias "ruidosas" de projetos, foi possível realizar predições com taxa de acerto entre 74 e 82%. Caso a base de projetos fosse mais uniforme, certamente os resultados seriam ainda mais favoráveis. Analisando o modelo materializado no sistema $UPPORT em comparação com o produzido automaticamente pelo *software* Genie (considerado o modelo *baseline*), vale ressaltar que ele (construído de forma *top-down* refletindo visão de especialistas) tem potencial para ser empregado em projetos de natureza distinta daquele explorado ao longo do texto. Essa maior generalidade, somada com a agregação do conhecimento de especialistas, explica os resultados com menor número de respostas diferente da correta (18%) em relação ao obtido de modo *bottom-up* (26%) a partir, exclusivamente, das instâncias de projeto WNet.

6 CONCLUSÕES

Ao longo desta dissertação foi proposta uma solução que, aproveitando-se de dados do ciclo de vida de projetos de gerenciamento de TI, é capaz de prever os custos atrelados à fase de suporte. Na verdade, como mostrado ao longo da dissertação, a nossa solução baseada em redes Bayesianas vai além, permitindo também previsões, no sentido inverso, associadas com as fases de desenvolvimento e testes. Uma análise comparativa entre as estimativas obtidas no Capítulo 5 com a opinião de especialistas em projetos de gerenciamento de TI constatou que a nossa solução é capaz de prever custos com elevado grau de confiança. Tal foi confirmado, também, pelo modelo Bayesiano criado automaticamente através do *software* Genie.

Outra contribuição desse trabalho implica no alinhamento da solução de estimativas de custos proposta com as sugestões indicadas pelas boas práticas encontradas na literatura, permitindo a integração de uma técnica confiável junto com uma base de históricos de projetos de TI. Outrossim, nossa proposta inclui uma solução que permite ser alimentada a todo momento com novos projetos, buscando a constante atualização dos valores das variáveis.

É importante mencionar que os bons resultados da solução consideraram uma base consistente (em termos de qualidade de perfis) e rica (em termos de instâncias) de projetos anteriores de TI. No entanto, acreditamos que esses requisitos não limitem a aplicabilidade de nossa solução por três razões. Primeiro, há uma tendência crescente de que organizações (especialmente as de médio e grande porte) empreguem boas práticas e processos propostos por *frameworks* de gerência de projetos. Segundo, organizações com um determinado nível de maturidade já adotam ferramentas computacionais de gerenciamento (de prateleira ou personalizadas) de projetos de TI. E terceiro, a abordagem Bayesiana viabiliza o uso da experiência de especialistas (caráter qualitativo), o que pode ser muito útil em ambientes com ausência de dados.

Uma discussão que merece atenção diz respeito aos resultados obtidos com os dois modelos Bayesianos apresentados no trabalho. Na comparação com a opinião de especialistas em projetos de TI, o modelo do sistema $UPPORT apresentou ligeira vantagem sobre o modelo Bayesiano do *software* Genie. Essa vantagem pode ser explicada pela sensível diferença na topologia das duas redes. No modelo $UPPORT o tamanho do projeto (*DevProjectSize* e *SupProjectSiza*) influencia no tempo, mas a propagação de sua influência é moderada pelo tamanho da equipe (*DevStaffSize*

e *SupStaffSize*). Isso significa que os resultados julgados pelos especialistas corroboram com nossa visão de que a relação de causa-efeito entre tamanho e tempo é subordinada a uma variável intermediária (no nosso caso *DevStaffSize* e *SupStaffSize*).

Por outro lado, no modelo proposto pela ferramenta Genie, o tempo do projeto (*DevTime* e *SupportTime*) possui relação de dependência direta com tamanho do projeto e tamanho da equipe. A partir disso, é possível chegar a uma constatação de relevância prática sobre os esforços em estimativas de projeto, na qual o tamanho do projeto é um fator importante e que deve ser considerado. Porém, a relação de causa-efeito entre tamanho e tempo deve ser mediada por outro fator, que no contexto deste estudo foi o tamanho da equipe. Observe também que, em relação à granularidade das previsões geradas, estimativas mais detalhadas podem ser obtidas pela agregação dos valores observados (para cada variável do modelo Bayesiano) em um número maior de estados mutuamente exclusivos.

Um aspecto-chave desse trabalho diz respeito ao uso de históricos de dados para suas previsões. Na literatura encontra-se trabalhos relacionados ao tema dessa dissertação que não fazem uso de históricos de dados para realizar suas estimativas. Se por um lado não analisar históricos pode facilitar o processo de estimativas, por outro, os resultados obtidos podem não refletir precisamente situações reais, visto que o desempenho atual da organização pode ser visto como uma projeção de seus projetos passados. Além disso, o uso de histórico pode garantir que informações preciosas não sejam diluídas com o passar do tempo, ou até mesmo, que estas informações não sejam exclusivas de um pequeno grupo.

A partir das investigações realizadas e dos resultados alcançados o trabalho apresentado nesta dissertação foi apresentado e discutido com a comunidade em eventos científicos nacionais e internacionais de grande relevância. As críticas, de modo geral, foram extremamente positivas, o que demonstra a importância e a qualidade do trabalho desenvolvido. Algumas das sugestões feitas pelos revisores estão consolidadas nessa dissertação. Abaixo segue a relação dos dois artigos submetidos e aceitos dentro do tema desse trabalho:

- DALMAZO, Bruno Lopes; CORDEIRO, Weverton Luis da Costa; SOUSA, Abraham Lincoln Rabelo de; WICKBOLDT, Juliano A; LUNARDI, Roben C; SANTOS, Ricardo L dos; GASPARY, Luciano Paschoal; GRANVILLE, Lisandro Zambenedetti; BARTOLINI, Claudio; HICKEY, Marianne. "**Leveraging IT Project Lifecycle Data to Predict Support Costs**". IM 2011 (12th IFIP/IEEE International Symposium on Integrated Network Management). Dublin - Irlanda. (DALMAZO et al. 2011a).

- DALMAZO, Bruno Lopes; CORDEIRO, Weverton Luis da Costa; SOUSA, Abraham Lincoln Rabelo de; WICKBOLDT, Juliano A; LUNARDI, Roben C; SANTOS, Ricardo L dos; GASPARY, Luciano Paschoal; GRANVILLE, Lisandro Zambenedetti; BARTOLINI, Claudio; HICKEY, Marianne. "**Variáveis de Projetos de TI "na Balança": Uma Abordagem Bayesiana para**

Previsão de Custos de Suporte". SBES 2011 (XXV Simpósio Brasileiro de Engenharia de Software). São Paulo - Brasil. (DALMAZO et al. 2011b).

Em investigações futuras pretende-se explorar formas de apresentar resultados mais detalhados das estimativas geradas. Também sugere-se a realização de experimentos com o sistema \$UPPORT aplicado a outros cenários envolvendo diferentes projetos de TI. Isso garantiria sua generalidade, caso os resultados alcançados fossem satisfatórios. Outras direções de pesquisa incluem: (*i*) avaliação da solução proposta considerando traços gerados por ferramentas de gerenciamento de projetos de TI; (*ii*) ampliação do escopo da solução para outras fases do ciclo de vida do projeto de TI e; (*iii*) investigação de outras variáveis relacionadas ao ciclo de vida de projetos de TI, a fim de analisar e prever outros aspectos.

REFERÊNCIAS

ADOLFSON, M.; LASEéN, S.; LINDé, J.; VILLANI, M. Bayesian Estimation of an Open Economy DSGE Model with Incomplete Pass-Through. **Journal of International Economics**, [S.l.], v.72, n.2, p.481 – 511, 2007.

AMBLER, S.; NALBONE, J.; VIZDOS, M. **Enterprise Unified Process, The**: extending the rational unified process. [S.l.]: Prentice Hall Press Upper Saddle River, NJ, USA, 2005.

BARNSON, M. et al. The Bugzilla Guide. **Estados Unidos**, [S.l.], 2004.

BOEHM, B. W. **Software Engineering Economics**. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1981.

BOEHM, B. W.; VALERDI, R. Achievements and Challenges in Cocomo-Based Software Resource Estimation. **IEEE Software**, Los Alamitos, CA, USA, v.25, p.74–83, 2008.

BRAY, T.; PAOLI, J.; SPERBERG-MCQUEEN, C.; MALER, E.; YERGEAU, F. Extensible markup language (XML) 1.0. **W3C recommendation**, [S.l.], v.6, 2000.

CANNON, D.; WHEELDON, D. **IT Infrastructure Library: itil service operation version 3.0**. London, UK: Office of Government Commerce (OGC), 2007. 393p.

CIM. **Distributed Management Task Force. Common Information Model**. Visited on: April, 2007. [Online]. Disponível em: http://www.dmtf.org/standards/cim.

CORDEIRO, W.; MACHADO, G.; ANDREIS, F.; SANTOS, A. dos; BOTH, C.; GASPARY, L.; GRANVILLE, L.; BARTOLINI, C.; TRASTOUR, D. ChangeLedge: change design and planning in networked systems based on reuse of knowledge and automation. **Computer Networks**, [S.l.], v.53, n.16, p.2782–2799, 2009.

COZMAN, F. Javabayes: bayesian networks in java. **http://www.cs.cmu.edu/javabayes**, [S.l.], 2001.

DARWICHE, A. Bayesian networks. **Commun. ACM**, New York, NY, USA, v.53, p.80–90, December 2010.

DUDA, R.; HART, P.; STORK, D. **Pattern Classification**. 2.ed. [S.l.]: Citeseer, 2001.

FUNG, C. J.; ZHU, Q.; BOUTABA, R.; BASAR, T. Bayesian Decision Aggregation in Collaborative Intrusion Detection Networks. **Network Operations and Management Symposium (NOMS 2010)**, [S.l.], p.349 –356, 2010.

Genie and Smile Systems. **The Decision Systems Laboratory of the University of Pittsburgh**. [Online]. Disponível em: http://genie.sis.pitt.edu/.

HAIDER, S. W.; CANGUSSU, J. W.; COOPER, K. M.; DANTU, R.; HAIDER, S. Estimation of Defects Based on Defect Decay Model: ed^3m. **IEEE Transactions on Software Engineering**, Los Alamitos, CA, USA, v.34, p.336–356, 2008.

HEWLETT-PACKARD. **Quality Center**. Available at: <http://h50281.www5.hp.com/software/index.html >. [Online]. Disponível em: http://h50281.www5.hp.com/software/index.html.

ISBSG. **International Software Benchmarking Standards Group Dataset. Release 10**. [Online]. Disponível em: http://www.isbsg.org/.

ITIL. **Office of Government Commerce (OGC). Information Technology Infrastructure Library.** [Online]. Disponível em: http://www.itil-officialsite.com/.

LOY, M.; ECKSTEIN, R. **Java Swing**. [S.l.]: O'Reilly Media, Inc., 2002.

LOYD, V.; RUUD, C. **IT Infrastructure Library: itil service design version 3.0**. London, UK: Office of Government Commerce (OGC), 2007. 449p.

LUNARDI, R.; COSTA CORDEIRO, W. da; WICKBOLDT, J.; MACHADO, G.; ANDREIS, F.; SANTOS, A. dos; BOTH, C.; GASPARY, L.; GRANVILLE, L. CHANGEADVISOR: alinhando o planejamento de mudanças em infra-estruturas de rede e serviços a propósitos de negócio. **Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC 2009)**, [S.l.], p.437–450, may 2009.

MENDES, E.; MOSLEY, N. Bayesian Network Models for Web Effort Prediction: a comparative study. **IEEE Transactions on Software Engineering**, Los Alamitos, CA, USA, v.34, p.723–737, 2008.

OLIVER, N.; ROSARIO, B.; PENTLAND, A. A Bayesian computer vision system for modeling human interactions. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, [S.l.], v.22, n.8, p.831–843, 2002.

PMBOK. **A guide to the project management body of knowledge** : (pmbok guide). 4.ed. Newtown Square, PA: Project Management Institute, Inc., 2008.

ROYCE, W. Managing the development of large software systems. In: IEEE WESCON. **Proceedings...** [S.l.: s.n.], 1970. v.26, n.8, p.1–9.

SOARES, M. Metodologias ágeis extreme programming e scrum para o desenvolvimento de software. **Revista Eletrônica de Sistemas de Informação ISSN 1677-3071 doi: 10.5329/RESI**, [S.l.], v.3, n.1, 2009.

SPIRTES, P.; GLYMOUR, C.; SCHEINES, R. **Causation, Prediction, and Search**. [S.l.]: The MIT Press, 2000.

TAYLOR, S.; CASE, G.; SPALDING, G. **IT Infrastructure Library: itil continual service improvement version 3.0**. London, UK: Office of Government Commerce (OGC), 2007. 308p.

TAYLOR, S.; IQBAL, M.; NIEVES, M. **IT Infrastructure Library: itil service strategy version 3.0**. London, UK: Office of Government Commerce (OGC), 2007. 373p.

TAYLOR, S.; LACY, S.; MACFARLANE, I. **IT Infrastructure Library: itil service transition version 3.0**. London, UK: Office of Government Commerce (OGC), 2007. 399p.

VERHOEF, C. Quantitative IT Portfolio Management. **Science of Computer Programming**, [S.l.], v.45, p.1–96, 2002.

WfMC. **Workflow Process Definition Interface - XML Process Definition Language**. Visited on: May, 2007. The Workflow Management Coalition Specification. [Online]. Disponível em: http://docs.oasis-open.org/wsbpel/2.0/.

WICKBOLDT, J.; MACHADO, G. S.; LUNARDI, R.; COSTA CORDEIRO, W. L. da; SANTOS, A.; BOTH, C. Automatizando a Estimativa de Riscos em Sistemas de Gerenciamento de Mudanças em TI. In: Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC 2009). **Proceedings...** [S.l.: s.n.], 2009. p.423–436.

ANEXO A ARTIGO PUBLICADO – IM 2011

Neste anexo, o artigo intitulado "Leveraging IT Project Lifecycle Data to Predict Support Costs" é apresentado. Essa foi a primeira publicação no tema desta dissertação em eventos científicos renomados. A solução para previsão de custos de suporte foi apresentada, bem como foi desenvolvido um primeiro protótipo chamado $UPPORT.

- **Título:**
  Leveraging IT Project Lifecycle Data to Predict Support Costs
- **Conferência:**
  12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011)
- **URL:**
  `http://www.ieee-im.org/`
- **Data:**
  23-27 Maio de 2011
- **Local:**
  Trinity College Dublin, Dublin, Irlanda

Leveraging IT Project Lifecycle Data to Predict Support Costs

Bruno L. Dalmazo*, Weverton L. Cordeiro*,
Lincoln Rabelo*, Juliano A. Wickboldt*,
Roben C. Lunardi*, Ricardo L. dos Santos*,
Luciano P. Gaspary* and Lisandro Z. Granville*
*Institute of Informatics
Federal University of Rio Grande do Sul, Brazil
{bldalmazo, wlccordeiro, rabelo, jwickboldt,
rclunardi, rlsantos, paschoal, granville}@inf.ufrgs.br

Claudio Bartolini‡ and Marianne Hickey†
‡Hewlett Packard Laboratories, Palo Alto, USA
†Hewlett Packard Laboratories, Bristol, UK
{claudio.bartolini, marianne.hickey}@hp.com

***Abstract*—There is an intuitive notion that the costs associated with project support actions, currently deemed too high and increasing, are directly related to the effort spent during their development and test phases. Despite the importance of systematically characterizing and understanding this relationship, little has been done in this realm mainly due to the lack of proper tooling for both sharing information between IT project phases and learning from past experiences. To tackle this issue, in this paper we propose a solution that, leveraging existing IT project lifecycle data, is able to predict support costs. The solution has been evaluated through a case study based on the ISBSG dataset, producing correct estimates for more than 80% of the assessed scenarios[1].**

I. Introduction

Information Technology (IT) project lifecycle management consists of a systematic approach for organizing the deployment of IT projects, and has as goals to contribute to increasing staff productivity, improving product quality, and reducing IT project costs [1]. IT projects typically refer to the deployment and/or maintenance of a software and hardware infrastructure, and they are generally materialized over a succession of phases, such as *analysis*, *planning*, *development*/*deployment*, *testing*, and *support*. By ensuring the use of project lifecycle management best practices and enforcing the monitoring and the control of each phase that compose a process, organizations may standardize and consequently ease the deployment of projects having such a nature.

Three important phases of project lifecycle management receive special attention in this paper: *development*, *test*, and *support*. The relationship between these phases typically occurs as follows. Once an IT project is approved and its business requirements are captured and understood, it may then be executed. In parallel or in a subsequent moment, the product (or service) under implementation may be tested, with the goal of ensuring that it satisfies the previously identified functional and non-functional requirements.

During the test phase, errors may be found, thus leading to the creation of reports. While some of these errors are corrected, others are documented only (for example, because of time and financial constraints). As such errors manifest themselves after the project deployment and delivery, they are treated as incidents and/or escalated to problems [2]. At this stage, they demand the consumption of resources (both human and material) – from the organization responsible for the support of the product in question – in order to mitigate their negative effects. From this point, there are two possible scenarios. The support team may either launch a new IT project in order to deal with the reported problem, or it may just indicate to the user what is the workaround procedure to be adopted, thus characterizing a support action without deployment or development of new IT assets.

The effort demanded to execute and assist support actions naturally has an associated cost. It is believed that there is a strong relationship between such effort and the one spent during the development and test phases of an IT project. Systematically characterizing and understanding this relationship is a non-trivial task, which has received marginal attention due to the following reasons. First, information sharing between the various phases that compose the lifecycle of IT projects is hampered by the lack of proper tooling, with few relationships (if any) being established between these information. Second, little knowledge is extracted from existing tools in order to enable learning from past experiences.

The motivation for approaching the aforementioned issues and, more specifically, determining the relationship between development/test and support phases, lies in the possibility of enabling managers to answer questions such as *how much time and effort will an IT project demand from the support staff after its deployment?* And *how to plan the development/test cycles given an upper bound for support costs?* Answers to these questions may provide organizations with the opportunity of a trustworthy learning from their past experiences. Furthermore, they have the potential to increase the productivity of these organizations and the quality of project deliverables, in addition to improve planning and deployment of future projects.

To fill in this gap, in this paper we propose a solution that enables the estimation of support costs by means of information obtained from development and test phases. In contrast to previous researches carried out in the field, our solution establishes a relationship between information produced in different phases of a project lifecycle. More importantly, by

[1]This result was achieved in cooperation with Hewlett-Packard Brasil Ltda. using incentives of Brazilian Informatics Law (Law no 8.2.48 of 1991).

employing a Bayesian model with dynamic feeding, it enables solving the posed questions in both directions (either from development → test → support or from support → test → development), obtaining cost predictions consonant with past IT projects carried out in the organization. To prove the concept and technical feasibility, we evaluated our solution using real (and synthetic) traces related to instances of IT projects for the deployment of wireless network infrastructures.

The remainder of this paper is organized as follows. Section II covers some of the most prominent related work. Section III introduces the proposed solution for estimating support costs, highlighting the information model for persisting information from the various phases of IT projects, the processing of historical data from past projects, and the instantiation of the Bayesian model that supports our work. Section IV presents the evaluation carried out to analyze the proposed solution, and discusses obtained results. Finally, Section V closes the paper with final remarks and prospective directions for future research.

II. Related Work

The field of costs estimation, in the context of IT Project Management, has received recently a great deal of attention from the scientific community. One may note, however, that research efforts have been mainly focused on methods to predict software project costs (human and material resources, and time). Moreover, as far as we are aware of, not only the scope of investigations lies on projects having a very specific nature; the relationship between the various phases that compose the project lifecycle is barely addressed as well. Given this overview, next we discuss those publications that have been found as being more correlated to our research, despite having distinct goals from those of this paper.

Constructive Cost Model (COCOMO) [3] is one of the most popular models for cost estimation present in the literature. Its goal is estimating time and efforts that a software project will demand to be carried out. COCOMO uses a simple regression model, relying on attributes such as *function points* and *lines of code*. Since its formulation was based on the study of 63 software projects carried out in 1981, with the establishment of static relationships (according to the proposed regression model), its efficacy is questionable in current projects. Furthermore, its estimates focus on project development complexity only, without establishing a relationship among other phases of the project lifecycle, such as test and support. More recently, some derivatives, like COCOMO II and COINCOMO [4], have been proposed with the purpose of updating the original model, but without violating their native characteristics.

Mendes and Mosley [5] carried out a study using eight Bayesian models, four created dynamically using the *Hugine PowerSoft* tool, and four developed by specialists. The goal of the study was comparing the effectiveness of those models to estimate development efforts of Web-related projects. Even though the authors explore a variety of models, the lack of variables from other phases of the project lifecycle (for example, variables from test and support phases) prevents estimating projects' support costs.

The ED^3M model, proposed by Haider et al. [6], enables estimating – during the test phase – how many "bugs" a software project will present once concluded. This model is based on Probability Theory (Estimation Theory) and does not require already established knowledge as input. In other words, it does not rely on historical data from past projects in order to compute the estimates. If, on one hand, not relying on the analysis of historical traces may facilitate the process of obtaining estimates, on the other hand, the obtained results may not accurately reflect real life situations, given that the current performance of organizations may be seen as a projection of their past projects.

Other investigations merit attention, either because they explore alternative techniques to predict IT project-related costs, or because they propose different, but still related-enough uses of Bayesian networks. Verhoef [7] presents simple ways to estimate both the execution time of a software project and the size of the development staff. In regard to Bayesian networks, Fung et al. [8] employed this technique to model the sharing of collective experiences on alerts raised by various Intrusion Detection Systems. In other research, Adolfson et al. [9] use Bayesian networks to dynamically estimate various economic indexes, such as unemployment rate, inflation, and net exports of a country.

In summary, in spite of the recent research observed on the topic of costs estimates, none of the aforementioned works enable the prediction of support costs from information gathered from development and test phases. Furthermore, these investigations use limited approaches as they are either simplistic (i.e., based on a limited number of variables) or static (i.e., based on outdated analytic models), and do not take into account learning from past experiences. In order to deal with these limitations, in the following section we introduce a conceptual solution for prediction of support costs, by means of a Bayesian network fed with data gathered from projects executed in the past.

III. Proposed Solution

Our solution focuses on providing a systematic approach for estimating support (or, conversely, development and/or test) costs from historical data of IT projects. Figure 1 depicts the basis of our solution, highlighting its main conceptual components, personnel involved, and their interactions.

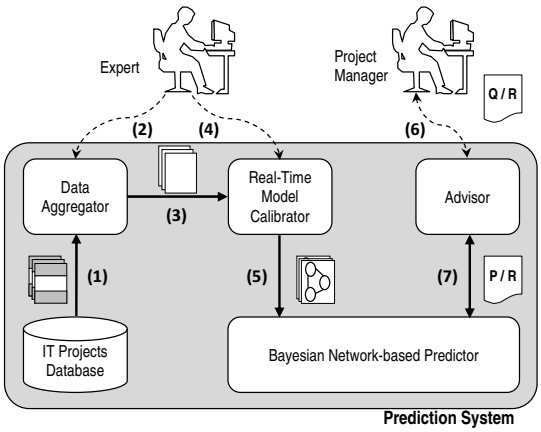


Fig. 1: Elements of the proposed solution and interactions.

Raw data about IT projects carried out in the organization are captured and persisted in the *IT Projects Database* repository, following the information model further discussed

in subsection III-A. This data is subsequently processed by the *Data Aggregator* component in order to extract, from each project, relevant information (e.g., development, test, and support time) for computing the estimates. After that, this information is stored in variables (flow 1 in Figure 1). An *Expert* may interact with the *Data Aggregator* component (flow 2) to influence the aggregation process. He/she may determine, for example, how to populate the variables with information gathered from the raw data, or even indicate which aspects should be privileged during the aggregation (e.g., accuracy, efficiency, or granularity).

Once the aggregation process is complete, the values (of variables) associated with the analyzed projects are passed to the *Real-Time Model Calibrator* component (flow 3). This component is responsible for normalizing the observed values (for each variable), and clustering them into mutually exclusive states. The clustering may be performed either fully- or semi-automatedly, the latter being subject to intervention by an expert to aid the process (4). Both the aggregation and calibration processes are further discussed in Subsection III-B.

The results of the previously mentioned clustering are used as input parameters by the *Bayesian Network-based Predictor* component (flow 5). This component models the variables (nodes) along with their causality relationships (edges) by means of an acyclic graph. Moreover, for each variable Y that has as parents X_1, ..., X_n, there is a probability table P($Y \mid X_1$, ..., X_n). In case Y does not have a parent node, the probability table is reduced to an unconditional probability P(Y). This Bayesian model (further described in Subsection III-C) enables computing the probabilities for some nodes (i.e., estimates) one may desire, given a number of pre-defined, fixed values for the other nodes (hypotheses). In order to compute project estimates, the project manager interacts with the *Advisor* component (6), which is a graphical front-end. This component, in turn, interacts with the inference engine of the *Bayesian Network-based Predictor* (7), by submitting to it the project manager's queries (i.e., set of hypotheses), and returning back the computed estimates.

Having presented a general view of our solution, in the following subsections we describe in more detail (*i*) the model for management and persistence of IT projects, (*ii*) the process for aggregation and normalization of project data; and (*iii*) the Bayesian model used to generate estimates of support (or development/test) costs.

A. IT Project Information Model

As previously mentioned, the prediction of support costs requires access to detailed information about the lifecycle of a set of IT projects (e.g., composing activities, resources involved, time consumed) previously deployed by/within the organization. In order to formalize this information, we propose an information model that aggregates classes from the *Common Information Model* (CIM) [10] and the *Workflow Management Coalition Specification* [11]. The model also incorporates classes that materialize information maintained by IT project management systems, such as *HP Quality Center* [12]. Figure 2 presents a partial view of the model.

An instance of class *IT Project*, the starting point of the model, defines an IT project, i.e., a temporary effort employed to create and/or deploy a new product, service, or software ordered based on the needs of the stakeholder (i.e., the client or end-user). Every IT project may be delivered to the stakeholder by means of one or more releases (instances of class *Release*). Each release corresponds to a partial version of the IT asset being delivered and, in turn, may be accomplished by means of one or more iterations (*Iteration*). Finally, an iteration is composed of one or more cycles (*Cycle*), such as analysis, planning, development, and testing. The aforementioned classes belong to the *Development* package shown in Figure 2.

In order to organize the activities carried out within each cycle, one or more work plans (instances of class *WorkPlan*) are defined. A work plan is a workflow of activities that follows the WfMC standard [11]. Taking as an example the test cycle associated with a given iteration/release of an IT project for deploying a wireless network infrastructure, a set of activities (*Activity*) – along with transitions (*Transition*) between them – would model an ordered test work plan. Instances of class *Activity* always have associated participants. These are represented by instances of class *ParticipantSpecification*, and may be either human or material resources assigned to the activities. Such resources are mapped by means of instances of classes from CIM. At the same time, activities may produce and/or consume artifacts (*Artifact*). Examples of artifacts are requirement (*RequirementDocument*) and test (*TestDocument*) documents. In addition to enabling a systematic documentation of the IT project, these artifacts also facilitate information sharing between the various phases of the project lifecycle. For example, test items (documented by means of instances of class *TestItem*) may be associated with the validation of one or more requirements (*RequirementItem*).

The model illustrated in Figure 2 also enables the persistence of information about support actions, which may be of two distinct types. The former consists of the identification of an incident (instance of class *ProblemReport*), whose solution comprises both the creation of a trouble ticket (*TroubleTicket*) and the execution of a simple workaround plan (*WorkPlan*). This plan incorporates the guidelines, provided by the support staff (*Person*), for the execution of a set of corrective activities (*Activity*). The second type of support action is also initiated upon identification of an incident and creation of a trouble ticket. However, due to its nature/severity, it is associated with a problem. Its resolution demands either the development of fixing *patches*, or the maintenance of elements of the infrastructure, which may occasionally characterize a new project to be developed (*ITProject*).

B. Aggregation and Clustering of Project Data

As mentioned earlier in this section, the summarization of project data into variables is performed by the *Data Aggregator* component. Even though the conceptual solution does not aim at establishing a unique set of variables, some of them are intuitively important for the estimates proposed hereafter. Among the most prominent examples, we cite *DevProjectSize* and *Dev/Test/SupStaffSize*. The aggregation process consists of navigating through each project available in the *IT Projects Database* repository and assigning, from the observation of several objects from the model, values to these variables. In the case of variable *SupStaffSize*, for example, it is necessary (*i*) to navigate through all activities (instances of *Activity*) of the work plans (*WorkPlan*) associated with trouble tickets (*Trou-*

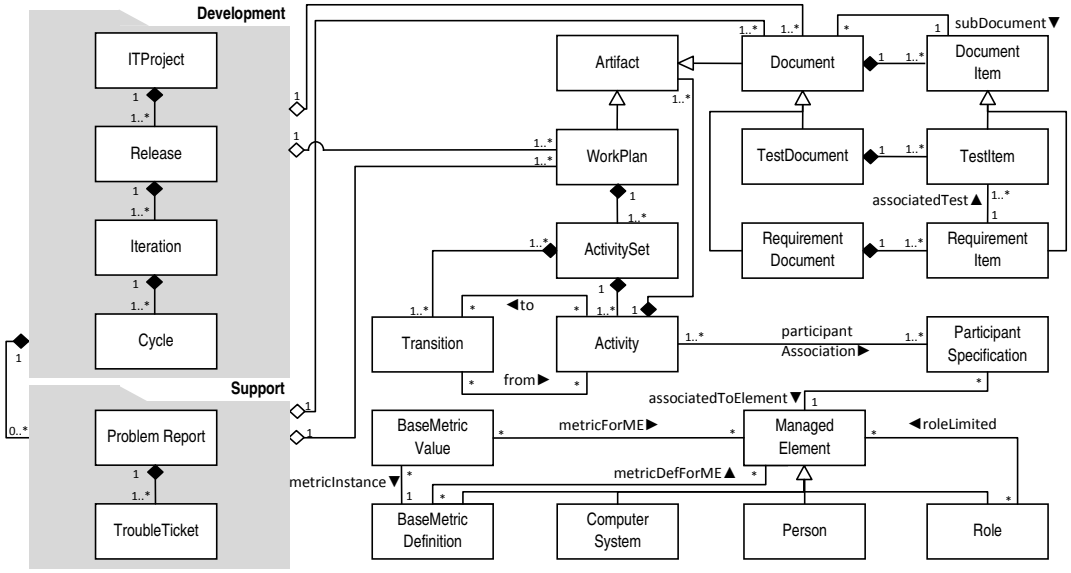


Fig. 2: Partial view of the model for persistence of IT project lifecycle related information

bleTicket), and (*ii*) to count the number of human operators involved (*Person*). The other variables are valued analogously.

In complement to the aggregation process, the *Real Time Model Calibrator* component runs (for each variable found) a clustering process of these values. As a result, each variable is now represented by a set of states (which model distinct natures or magnitudes for that variable). Considering again the example of the variable *SupStaffSize*, this procedure could generate as result the following states: *small*, *medium*, and *large*. Note that the clustering may be carried out using an algorithm to this end, such as *k-means*, *fuzzy c-means*, and *naive Bayes* classification [13], or even taking advantage of the knowledge owned/acquired by an expert.

As a result of the clustering, we have the unconditional variables of the estimate model properly fed (which represent the knowledge available *a priori*). Still as part of the calibration process, the relationships existing between these variables are analyzed, and an influence graph is established. Such a graph may be seen as the skeleton of the Bayesian estimate model. This model is further described in the following subsection.

C. Bayesian Model

The Bayesian model for estimating support (or, conversely, development and/or tests) costs is composed of sixteen (16) variables, which are organized in three groups: *Development/Deployment*, *Test*, and *Support*. Figure 3(a) illustrates the model, highlighting these groups in light gray, dark gray, and white, respectively. Variables such as *StaffProductivity*, *StaffSize*, *ProjectSize*, and *Time* appear in every group, whereas the others appear in one group or another, given their specificity (for example, *TestCoverage*). The linkage between these variables, represented in the figure by means of directed edges, expresses causality relationships between them. Please observe that such linkage is established during the calibration process.

The configuration of the aforementioned model was outlined based on (*i*) the opinion of experts, (*ii*) an analysis of the dataset published by the *International Software Benchmarking Standards Group* (ISBSG) [14], which contains information from over than 5,000 IT projects, and (*iii*) the observation of variables employed in similar investigations [5]. Note that the proposed model may be either *simplified* – in order to prioritize processing time – or *extended* – in order to capture other aspects (variables and relationships) not initially envisaged.

The unconditional variables of the proposed model – which were populated in the calibration process – influence either direct or indirectly the computation of probabilities for the states of conditional ones. This is the case of variable *SupProjectSize*, for example, in regard to *SupStaffSize*. As one may note in Figure 3(b), *SupStaffSize* models a probability table that associates each of its states to the ones of *SupProjectSize* (parent variable). A cell from this table represents the probability that *SupStaffSize* (X=*small*, *medium*, *large*) occurs in *SupProjectSize* (Y=*null*, *small*, *medium*, *large*). Analogously, the other conditional variables also have their own tables.

Once the model is fully populated, it is possible to run *what if* queries (by establishing fixed values for one or more variables) and obtain the estimates for them (by observing the effect of the fixed values on the other variables). In the context of this work, it is now possible to answer questions such as *what are the expected support costs, given the time spent during the development/deployment of a project?* Or *how much one should invest in the development/test cycles of a project, given a limited, fixed yearly budget for support?* The estimates produced by the model are probabilities associated with the states of variables. In order to have a more intuitive summarization of such estimates, one may alternatively look at, for example, the value ranges associated with the resulting predominant state (the one with higher probability).

IV. Evaluation

To prove the concept and technical feasibility of our solution, we have developed $UPPORT, a development/test/support cost prediction system, and used it for qualitatively and quantitatively evaluating our solution. In this section we describe the $UPPORT system, the methodology employed, and the obtained results.

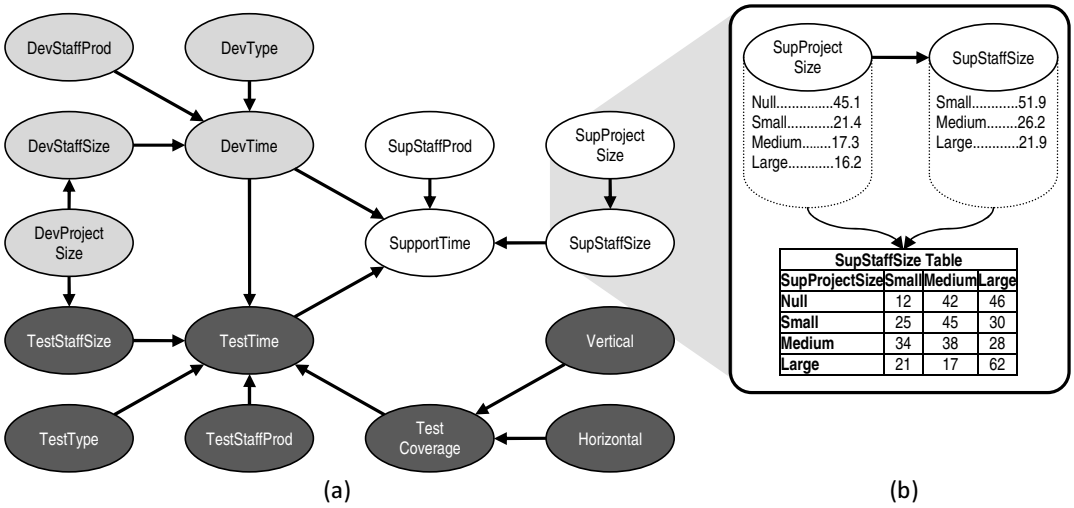


Fig. 3: Bayesian model for estimating development/test/support costs

For the purpose of the evaluation of our solution, we consider as a case study the prediction of costs (either development/test/support) associated with projects for the deployment of wireless network infrastructures (also referred to as WNet throughout this section). This case study was chosen since this kind of project is a representative instance of IT projects closely related to the topic of network and service operations & management. Other projects, omitted in this paper for the sake of space constraints, were also analyzed and will be discussed in future submissions.

A. The $UPPORT System

The $UPPORT system consists of a prototypical implementation of our solution, presented in the previous section. Developed using the Netica toolkit [15] (for the creation of the Bayesian model) and other off-the-shelf components, it enables computing costs estimates for a given project. Figure 4(a) illustrates the *Project Lifecycle* tab, in which one may observe the structure of a specific WNet project. Note that the project is described according to the information model described in Section III-A, being organized in releases, iterations, and cycles. Known data regarding the development, test, and support phases are reported within *Development*, *Test*, and *Support*, respectively. The idea is that all these data may be imported from other software such as HP Quality Center in an automated fashion, requiring little intervention from the project manager. By having these data, $UPPORT is able to extract, from the project being analyzed, values (in this case, already confirmed information) for states of a number of variables from the Bayesian model.

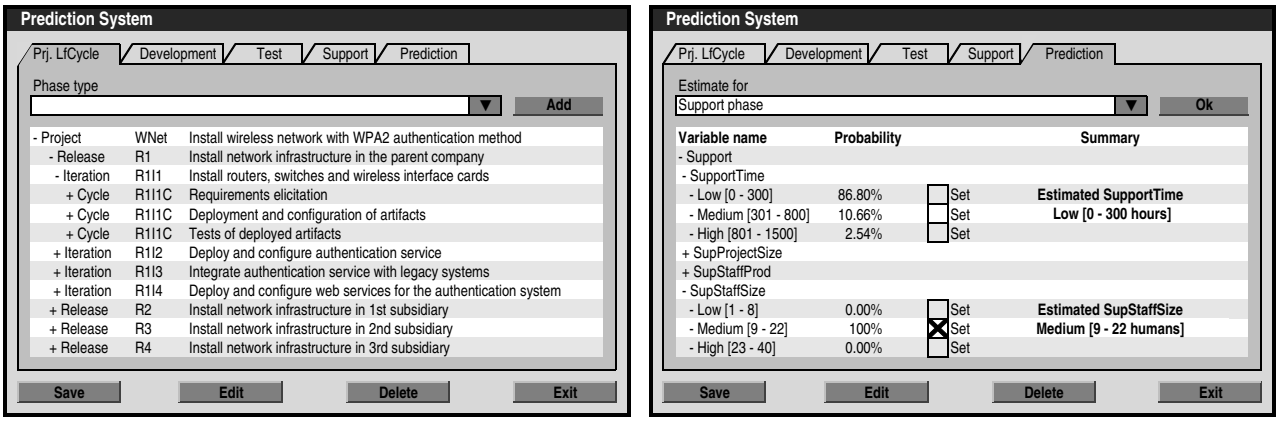
Alternatively (or complementarily), a project manager may inform, in the *Prediction* tab, arbitrary values (in this case, hypotheses) to states of the model variables. An example of this situation is illustrated in Figure 4(b) where, for the project under analysis, the manager establishes the hypothesis that the support staff size is *medium*. The $UPPORT system then recalculates the probabilities of the states of variables (from the Bayesian model) that are open and reports that, with 86.80% of chance, the time demanded for support shall be *low*. From this estimate, the system also presents the value ranges associated with the state exhibithing higher probability. In the case of this example, the summarization reports, as result, the interval 0 - 300 support hours.

B. Methodology

The evaluation was performed from three different perspectives. In the first perspective, a set of *what-if* questions regarding the instantiation of a WNet project was elaborated and submitted to the $UPPORT system, in order to obtain as result the estimate for a variable left undefined. In the second perspective, various hypothetical scenarios of instantiation of a WNet project were assessed. The goal in this case was investigating whether these scenarios would be feasible in actual instantiations. In both the first and second perspectives, the results obtained were compared with the opinion of specialists, in order to assess their accuracy. Finally, the third perspective considered the submission of a set of *what-if* questions in order to evaluate the sensibility of the proposed solution considering varying database profiles. Next we describe the main steps of our evaluation effort, whereas in the following subsection we present and discuss the results achieved.

The project database setup was an important task of the evaluation. We have resorted to the ISBSG dataset, from which we have selected 450 projects of our interest, to extract key parameters such as real value ranges for most of the model variables (e.g., *DevTime*, *TestTime* and *SupportTime*). In addition, looking at these values (for each project) we were able to clusterize them and consolidate variable states. Table I summarizes the outcome of this step. Take *DevTime* (development time) as an example. This variable has been organized in three states: *low* [0 - 1,056 hours], *medium* [1,057 - 3,168 h], and *high* [3,169 - 5,000 h].

Ideally, one would use the projects available in the ISBSG dataset as input to the Bayesian model. However, this was not possible because each project had an incomplete set of data available (variable values) and because IT projects were not linked to their corresponding support counterparts. To overcome this limitation, we have generated a synthetical project database merging known real data with information provided by a group of experienced IT project managers, as explained next. First, we have selected five variables (*DevTime*, *TestTime*, *SupStaffSize*, *SupStaffProd*, and *SupportTime*) and produced a



(a) Phases of the lifecycle of an instance of a WNet project

(b) *What if* questions and estimates obtained

Fig. 4: Graphical user interface of the $UPPORT system

TABLE I: Bayesian model variables and corresponding states

Variable	States*
DevTime	low [0-1,056], medium [1,057-3,168], high [3,169-5,000]
TestTime	low [0-300], medium [301-500], high [501-1800]
SupStaffSize	small [1-8], medium [9-22], large [23-40]
SupStaffProd	low [0-0.25], medium [0.26-0.75], high [0.76-4.00]
SupportTime	low [0-300], medium [301-800], high [801-1500]
DevStaffProd	low, medium, high
DevType	innovation, improvement, workaday
DevStaffSize	small, medium, large
DevProjectSize	small, medium, large
TestStaffProd	low, medium, high
TestType	manual, automatic
TestStaffSize	small, medium, large
TestCoverage	small, medium, large
Vertical	low, medium, high
Horizontal	low, medium, high
SupProjectSize	null(0), small, medium, large

**DevTime*, *TestTime*, and *SupportTime* are measured in hours. *SupStaffSize* is measured in number of humans, and *SupStaffProd* in ManPower/hour.

list of their possible state combinations ranging from {*low*, *low*, *small*, *low*, *low*} to {*high*, *high*, *large*, *high*, *high*}. Each combination represents an IT project scenario. Second, we have asked the aforementioned IT project managers to, based on their experience, classify each project scenario as *probable*, *possible*, or *improbable*. Third, we have created three project scenario "bags", containing, respectively, the *probable*, the *possible*, and the *improbable* scenarios. Finally, we have picked a varied proportion of scenarios from these "bags", depending on the desired project database profile, and populated both the unconditional and the conditional variables of the Bayesian model.

In order to analyze the accuracy of the predictions (our first evaluation objective), we have populated the Bayesian model with information gathered from 710 project scenarios (*control* project database) in the proportion of 70% (probable), 20% (possible), and 10% (improbable). This profile represents the case in which 70% of the projects conducted by an organization exhibits an "expected" behavior, while 30% performs badly, sometimes in an abnormal way. These proportions have been proposed by the consulted IT project managers and confirmed by the analysis of the ISBSG dataset. As for the model sensibility evaluation (our second evaluation objective), we have employed several proportions but, for the sake of space constraints, focus on the 33%-33%-33%, 50%-30%-20%, and 70%-20%-10% ones. In regard to the number of *test* project scenarios, we have employed a same set of thirty (randomly-chosen) in both evaluations.

C. Results and Discussion

The presentation and the discussion of the results obtained are organized as follows. First we report on the *accuracy* of both the cost predictions and the assessment of hypothetical project scenarios. Subsequently, we analyze the *sensibility* of the model to different project database profiles.

1) What-if prediction questions: Table II shows a set of 10 (out of 30) *what-if* questions that have been submitted to the $UPPORT system. In each of these questions, four variables have been assigned a state, and one of them (whose state should be predicted by the system) has been left undefined.

To illustrate, consider the case of Scenario 1. In this scenario, the submitted question was: *what is the expected support time necessary for maintenance of the assets delivered by a specific WNet project, if executed given a high amount of time available for carrying out the development and test phases, and a large number of support staff members whose average productivity is considered to be medium?* In other words, given *DevTime = high*, *TestTime = high*, *SupStaffSize = large*, and *SupStaffProd = medium* for the deployment of a particular WNet project, what is the expected state for *SupportTime*?

For the previously mentioned *what-if* question, the $UPPORT system returned, as result, the state *low*. It means that the WNet project in hand is expected to demand low support time after the final delivery of its assets. In order to validate the question, we have asked a different group of experienced IT managers (than the one that helped in the creation of the IT project database), called *specialists* henceforth, to evaluate whether they agreed or disagreed with the system prediction.

TABLE II: *What-if* questions for predicting costs of WNet projects.

Scenario	Variables					Expected	Specialists'
	DevTime	TestTime	SupStaffSize	SupStaffProd	SupportTime	Result	Opinion
1	high	high	large	medium	?	low	Agree
2	?	high	medium	high	low	high	Agree
3	medium	?	small	medium	low	medium	Agree
4	low	medium	?	medium	medium	medium	Agree
5	low	medium	large	?	low	medium	Disagree
6	high	?	medium	high	low	high	Agree
7	high	medium	medium	low	?	*any*	No Formed Opinion
8	low	high	large	?	medium	low	Partially Agree
9	low	?	large	low	medium	high	Disagree
10	high	high	large	medium	?	high	Partially Agree

For the case of Scenario 1, the specialists found the assignment of *low* to *SupportTime* a good estimate.

Consider also the case of Scenario 8, whose *what-if* question was: *what should be the average productivity of the support staff members, if there is a low amount of time available for carrying out the development phase, high amount of time for the support phase, a large number of support staff members, and one also wishes to deliver assets that require medium support time?* Put in another way, given *DevTime = low*, *TestTime = high*, *SupStaffSize = large*, and *SupportTime = medium* for the execution of a given WNet project, what should be the state of *SupStaffProd*?

For this question, the $UPPORT system returned as result the state *low*. This result indicates that, in order to carry out the WNet project considering the circunstances stated in the submitted question, the average productivity of the support staff members should be *low*. The specialists, however, indicated that the estimate given by the $UPPORT system for the *SupStaffProd* variable, although reasonable, might not hold for every execution of similar WNet projects. It means that *SupStaffProd = low* may be sufficient for running some WNet projects, given the circumstances stated in the submitted question. In general, however, *SupStaffProd* should be equal to either *medium* or *high* for successfully executing them.

From the other scenarios depicted in Table II, one may note that our solution was able to accurately estimate the results for six of the submitted *what-if* queries (1, 2, 3, 4, 6, and 7); in two cases (8 and 10), the estimates obtained were valid, although they do not conform to what would be generally expected (according to the specialists); and the system prediction was found to be incorrect for two cases only (5 and 9).

In general, from the 30 *what-if* queries submitted, the $UPPORT system was able to accurately estimate the expected costs and required conditions for the execution of WNet projects in 80% of the specified scenarios; only 3% of the estimates obtained were found to be incorrect, and 17% were inconclusive. These results evidence the effectiveness and eficacy of our solution in accurately estimating expected costs.

2) Assessment of hypothetical project scenarios: Table III presents the hypothetical scenarios – regarding the execution of WNet projects – submitted to the $UPPORT system in order to assess their feasibility in actual instantiations.

To illustrate, consider the hypothesis of Scenario 6. One wants to assess whether it is reasonable that an instance of a WNet project – executed considering a medium amount of time for the development phase, a high amount for the test phase, and a medium size support staff – requires a high amount of support hours, given that the productivity of each member of the support staff is *high*. According to the $UPPORT system, the probability of such a scenario actually take place would be 55.56% – which was deemed correct according to the opinion of the specialists we have consulted.

Table III also depicts probabilies for variations of the hypothesis of Scenario 6. Consider, for example, that the average productivity of each member of the support staff is *low*, instead of *high*; under this new setting, this hypothesis has 22.22% chance of taking place.

Now consider the case of Scenario 10, whose hypothesis is: *a WNet project – with a high amount of time available for both development and testing, and a large staff size with medium productivity allocated for support – should require a high amount of support hours for maintenance of the delivered assets (during their lifetime)*. The $UPPORT system reported that such a hypothetical scenario has 33.33% chance of actually taking place, whereas the specialists found it *improbable*; consequently, this prediction is regarded was inconclusive (since this scenario is neither probable nor improbable).

In summary, the $UPPORT system was able to provide accurate assessments for 8 out of the 10 hypotheses depicted in Table III. In one of the cases (Scenario 3), the system has returned an incorrect assessment (when compared to the specialists' opinion), whereas in other case (Scenario 10) the assessment was inconclusive.

3) Sensibility analysis of the proposed solution: Table IV shows the success rate that $UPPORT obtained when fed with varying database profiles. As one may note, evaluations of WNet projects considering a database profile where the majority of projects exhibited similar development/deployment patterns resulted in more accurate estimates. This is the case of the 70%-20%-10% profile, which provided the highest success rate (80% out of the 30 scenarios evaluated).

TABLE IV: Accuracy obtained with various database profiles

Project Profile	Correct (%)	Incorrect (%)	Inconclusive (%)
33-33-33	40	23	37
50-30-20	60	23	17
70-20-10	80	03	17

TABLE III: Snapshot of the predictions for the 70%-20%-10% project database profile

Scenario	Variables					Prediction (%)*			Specialists'	Analysis
	DevTime	TestTime	SupStaffSize	SupStaffProd	SupportTime	low	medium	high	Opinion	
1	high	high	large	medium	low	83.33	04.76	11.90	Probable	Correct
2	high	high	medium	high	low	05.13	05.13	89.74	Probable	Correct
3	medium	medium	small	medium	low	33.33	33.33	33.33	Probable	Incorrect
4	low	medium	medium	medium	medium	48.61	48.61	02.78	Probable	Correct
5	low	medium	large	high	high	22.22	22.22	55.56	Possible	Correct
6	medium	high	medium	high	high	22.22	22.22	55.56	Possible	Correct
7	high	medium	medium	low	high	16.67	41.67	41.67	Possible	Correct
8	low	high	large	medium	medium	16.67	41.67	41.67	Possible	Correct
9	low	high	large	low	medium	41.67	16.67	41.67	Improbable	Correct
10	high	high	large	medium	high	33.33	33.33	33.33	Improbable	Inconclusive

*Small/Medium/Large, if *SupStaffSize* is being considered for prediction, and Low/Medium/High otherwise.

In contrast, evaluations based on more amorphous profiles (i.e., profiles that do not present a prevailing project pattern) led to an increase in the number of incorrect estimates. This is the case of profiles 50%-30%-20% (with 60% of success rate) and 33%-33%-33% (with 40% of success rate only). These observations evidence that the number of unconclusive estimates (i.e., those that do not offer any meaningful information to the operator using the prediction system) grows proportionally to the number of project patterns found in the profile. Conversely, more accurate results may be achieved by profiles exhibiting most of the projects aggregated into fewer execution patterns.

V. Final Considerations and Future Work

In this paper we have proposed a solution that, taking advantage of existing IT project lifecycle data, is able to predict support costs. Actually, as shown along the paper, our Bayesian network-based solution goes further, also enabling predictions associated with development and test costs. By experimenting different values for coverage tests, team sizes, and other key variables, a project manager is able to test hypotheses regarding a project execution and thus optimize costs (without degrading quality of delivered assets). A comparative analysis of the obtained estimates with the opinion of specialists evidenced that our solution is able to predict expected costs with a high degree of confidence.

It is important to mention that our solution for predicting project costs requires a consistent (in terms of profiles) and rich (in terms of instances) base of past IT projects. This aspect should not limit, however, the applicability of our solution, for two reasons. First, there is a growing tendency for organizations (especially medium and large-size ones) to employ best practices and processes proposed by widely accepted and adopted project lifecycle management frameworks, such as PMBOK [1]. And second, organizations with a certain maturity level already use (either off-the-shelf or customized) IT project lifecycle management tools – which may facilitate the applicability of our solution in their contexts.

Observe also that, regarding the granularity of the generated predictions, finer-grained estimates may be obtained by clustering the observed values (for each variable of the Bayesian model) into a higher number of mutually exclusive states. In a future investigation we intend to focus on a automated, finer-grained summarization of the estimates generated, in order to offer more precise and intuitive information regarding project costs. Prospective directions for future research also include: (*i*) an evaluation the proposed solution considering traces generated by off-the-shelf IT project management tools; and (*ii*) an investigation of other IT project-related variables, in order to analyze and predict other aspects of project execution not initially envisaged in this work.

References

[1] Project Management Institute, Inc., *A guide to the project management body of knowledge : (PMBOK guide)*, 4th ed. Newtown Square, PA: Project Management Institute, Inc., 2008.

[2] Office of Government Commerce (OGC), "Information Technology Infrastructure Library (ITIL)," Office of Government Commerce (OGC), 2010, [Online]. Avaiable: http://www.itil-officialsite.com/.

[3] B. W. Boehm, *Software Engineering Economics*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1981.

[4] B. W. Boehm and R. Valerdi, "Achievements and challenges in cocomo-based software resource estimation," *IEEE Software*, vol. 25, pp. 74–83, 2008.

[5] E. Mendes and N. Mosley, "Bayesian network models for web effort prediction: A comparative study," *IEEE Transactions on Software Engineering*, vol. 34, pp. 723–737, 2008.

[6] S. W. Haider, J. W. Cangussu, K. M. Cooper, R. Dantu, and S. Haider, "Estimation of defects based on defect decay model: Ed3m," *IEEE Transactions on Software Engineering*, vol. 34, pp. 336–356, 2008.

[7] C. Verhoef, "Quantitative IT Portfolio Management," *Science of Computer Programming*, vol. 45, pp. 1–96, 2002.

[8] C. J. Fung, Q. Zhu, R. Boutaba, and T. Basar, "Bayesian Decision Aggregation in Collaborative Intrusion Detection Networks," *Network Operations and Management Symposium (NOMS 2010)*, pp. 349 –356.

[9] M. Adolfson, S. Laséen, J. Lindé, and M. Villani, "Bayesian estimation of an open economy dsge model with incomplete pass-through," *Journal of International Economics*, vol. 72, no. 2, pp. 481 – 511, 2007.

[10] Distributed Management Task Force, "Common Information Model," [Online]. Avaiable: http://www.dmtf.org/standards/cim.

[11] The Workflow Management Coalition Specification, "Workflow Process Definition Interface - XML Process Definition Language," [Online]. Avaiable: http://docs.oasis-open.org/wsbpel/2.0/.

[12] Hewlett-Packard, "Quality center," February 2009. [Online]. Available: http://h50281.www5.hp.com/software/index.html

[13] R. Duda, P. Hart, and D. Stork, *Pattern classification*, 2nd ed. Wiley-Interscience, 2000.

[14] International Software Benchmarking Standards Group Dataset, "Release 10," 2007. [Online]. Available: http://www.isbsg.org/

[15] Norsys Software Corp, "Netica APIs," February 2010. [Online]. Available: http://www.norsys.com/netica_api.html

ANEXO B ARTIGO ACEITO - SBES 2011

Neste anexo, o artigo intitulado "Variáveis de Projetos de TI "na Balança": Uma Abordagem Bayesiana para Previsão de Custos de Suporte" é apresentado. Essa foi a segunda submissão no tema desta dissertação em eventos científicos. Além do melhor detalhamento da solução para previsão de custos de suporte, esse artigo também trata de um estudo comparativo entre os resultados obtidos com um modelo Bayesiano produzido automaticamente pelo *software* Genie 2.0, opiniões de gerentes de projetos de TI e com o sistema $UPPORT.

- **Título:**
  Variáveis de Projetos de TI "na Balança": Uma Abordagem Bayesiana para Previsão de Custos de Suporte
- **Conferência:**
  XXV Simpósio Brasileiro de Engenharia de Software (SBES 2011)
- **URL:**
  `http://www.each.usp.br/cbsoft2011/portugues/sbes/sbes_pt.html`
- **Data:**
  26-30 Setembro de 2011
- **Local:**
  Universidade Presbiteriana Mackenzie, São Paulo, Brasil

Variáveis de Projetos de TI "na Balança": Uma Abordagem Bayesiana para Previsão de Custos de Suporte

Bruno L. Dalmazo*, Lincoln Rabelo*,
Weverton L. Cordeiro*, Juliano A. Wickboldt*,
Roben C. Lunardi*, Ricardo L. dos Santos*,
Luciano P. Gaspary* e Lisandro Z. Granville*
*Universidade Federal do Rio Grande do Sul, Brasil
{bldalmazo, wlccordeiro, rabelo, jwickboldt,
rclunardi, rlsantos, paschoal, granville}@inf.ufrgs.br

Claudio Bartolini‡ e Marianne Hickey†
‡Hewlett Packard Laboratories, Palo Alto, USA
†Hewlett Packard Laboratories, Bristol, UK
{claudio.bartolini, marianne.hickey}@hp.com

***Resumo*—Existe uma noção intuitiva de que os custos associados a ações de suporte de projetos de Tecnologia da Informação (TI), muitas vezes considerados já muito elevados e em crescimento, possuem forte vinculação com esforços empreendidos nas fases de desenvolvimento e teste. Apesar da importância de caracterizar e compreender a sistemática dessa relação, pouco tem sido feito neste domínio, principalmente devido à falta de mecanismos adequados tanto para o compartilhamento de informações entre as fases de um projeto de TI, quanto para aprender com experiências passadas. Para lidar com essa problemática, neste trabalho apresentamos um modelo Bayesiano para realizar previsões de custo de suporte baseado em dados da fase de desenvolvimento e teste em projetos de software. Além disso, apresentamos uma análise qualitativa e quantitativa do modelo, procurando demonstrar sua eficácia e eficiência, bem como discutimos suas potencialidades e limitações.**

***Abstract*—There is an intuitive notion that the costs associated with IT project support actions, currently deemed too high and increasing, are directly related to the effort spent during their development and test phases. Despite the importance of systematically characterizing and understanding this relationship, little has been done in this realm mainly due to the lack of proper mechanisms for both sharing information between IT project phases and learning from past experiences. To tackle this issue, in this paper we present a Bayesian model to perform support cost predictions based on data from software development and test phases. In addition, we present a qualitative and quantitative analysis of the model, in order to demonstrate its effectiveness and efficiency, and also discuss its potentialities and limitations.**

I. Introdução

O gerenciamento de projetos de Tecnologia da Informação (TI) consiste em uma abordagem sistemática para organizar e controlar a execução de projetos e tem por finalidade contribuir para aumentar produtividade da equipe, aprimorar qualidade dos produtos e reduzir custos dos projetos [1]. Projetos de TI geralmente consistem no desenvolvimento, implantação ou manutenção de uma infraestrutura de *software* e *hardware*. Ao estabelecer o uso de boas práticas e preconizar a monitoração e controle de cada fase (que, em conjunto, configuram um processo) deseja-se facilitar e padronizar a execução de projetos dessa natureza.

Três fases importantes do gerenciamento do ciclo de vida de projetos recebem atenção especial neste trabalho: desenvolvimento, teste e manutenção/suporte. O relacionamento entre essas três fases ocorre, tipicamente, da seguinte forma. Uma vez que um projeto é aprovado e os seus requisitos de negócio são capturados e entendidos, o mesmo pode, então, ser executado. Paralelamente à execução, ou em um próximo momento, o produto ou serviço em desenvolvimento pode ser testado, com o propósito de assegurar que satisfaz os requisitos de qualidade funcionais e não funcionais previamente elicitados.

Durante a fase de teste, erros podem ser encontrados, levando assim à criação de relatórios. Enquanto alguns desses erros são corrigidos, outros, devido a restrições de tempo e custo, não são detectados. Quando tais erros se manifestam após a implantação e entrega do projeto, eles são tratados como incidentes e/ou alçados para problemas [2]. A partir desse momento, passam a demandar, do setor da organização responsável pelo suporte, o consumo de recursos (materiais e humanos) para atenuar os efeitos negativos produzidos. Desta forma, há duas situações possíveis: a equipe de suporte pode dar origem a um novo projeto para lidar com o problema relatado, ou pode apenas indicar ao usuário qual é o procedimento de solução a ser adotado.

O esforço despendido para executar e apoiar as ações de suporte, tem, naturalmente, um custo associado. Acredita-se que exista uma forte relação entre tal esforço e o realizado durante as fases de desenvolvimento e teste de um projeto de TI. Caracterizar e compreender de maneira sistemática esta relação está longe de representar uma tarefa trivial e, consequentemente, é pouco estudada por duas razões principais. Em primeiro lugar, o compartilhamento de informação entre as várias fases que compõem o ciclo de vida de projetos de TI é dificultado pela falta de apoio apropriado por parte das ferramentas existentes, com poucos (ou nenhum) relaciona-

mentos estabelecidos entre essas fases. O segundo problema, por sua vez, é que pouco conhecimento é extraído a partir dessas ferramentas de modo a propiciar o aprendizado com experiências passadas.

A motivação para abordar os problemas recém mencionados e, mais especificamente, determinar a relação entre as fases de desenvolvimento, teste e suporte reside na possibilidade de apoiar gerentes de projetos a responderem perguntas como: *quanto tempo e esforço um projeto exigirá da equipe de suporte após sua implantação?* Ou: *como planejar o desenvolvimento e o teste dado um limite máximo de custo de suporte?* Respostas para essas perguntas podem oferecer às organizações a oportunidade de aumentar a produtividade da equipe e a qualidade dos produtos/serviços, bem como melhorar o planejamento e a implantação de projetos futuros.

Para lidar com o problema recém mencionado, este artigo apresenta um modelo para apoiar o relacionamento de informações produzidas ao longo do ciclo de vida de projetos e esboça um modelo Bayesiano para realizar previsão de custos de suporte. Além disso, descreve uma avaliação detalhada de sua eficácia para auxiliar no entendimento do compromisso entre custos de desenvolvimento, de teste e de suporte.

O restante do artigo está organizado como segue. Na Seção II aborda-se os trabalhos relacionados. Na Seção III apresenta-se a solução proposta para estimar custos de suporte, com destaque para o modelo de informação que permite persistir, de forma integrada, dados produzidos em diferentes fases do ciclo de vida de projetos e para o modelo Bayesiano que embasa o processo de predição. Na Seção IV descreve-se o estudo de caso conduzido e discute-se os resultados obtidos. Por fim, a Seção V encerra o artigo com as considerações finais e perspectivas de trabalhos futuros.

II. Trabalhos Relacionados

A área de estimativas de custos, no contexto da gerência de projetos, tem recebido grande atenção da comunidade científica nos últimos anos. Observa-se, contudo, que os esforços de pesquisa concentram-se em propor métodos para prever custos (recursos humanos/materiais e tempo) enfatizando a fase de desenvolvimento do *software*. Até onde sabemos, não apenas o escopo das investigações reside em projetos de natureza específica, como também relações entre fases do ciclo de vida de diferentes projetos são pouco exploradas. Diante deste panorama, discute-se a seguir alguns trabalhos correlatos.

O *Constructive Cost Model* (COCOMO) [3] é um dos modelos de estimativas de custo mais citados em trabalhos científicos da área. Seu objetivo é estimar o tempo e o esforço que um projeto de *software* despenderá ao ser implementado. O COCOMO funciona a partir de um modelo de regressão simples, sendo baseado em atributos como, por exemplo, Pontos de Função (PF) e linhas de código. Sua formulação foi fundamentada no estudo de 63 projetos de *software* produzidos em torno do ano de 1981, estabelecendo estimativas estáticas (conforme o modelo de regressão proposto) e colocando sob suspeita a eficácia delas em projetos atuais. Outro ponto fraco desse modelo é que suas estimativas focam em complexidade de desenvolvimento, não realizando relação com outras fases de projetos como, por exemplo, testes e suporte. Mais recentemente, alguns modelos derivados, a exemplo de COCOMO II e COINCOMO [4], foram propostos com propósito de atualizar o modelo original, sem, contudo, violar suas características nativas.

Mendes e Mosley [5] introduzem oito modelos Bayesianos, quatro criados dinamicamente (através das ferramentas *Hugin* e *PowerSoft*) e quatro criados por especialistas. A finalidade do trabalho foi realizar um estudo comparativo entre esses modelos para estimar o esforço de desenvolvimento em projetos *Web*. Embora os autores explorem vários modelos diferentes, a inexistência de variáveis de outras fases do ciclo de vida dos projetos como, por exemplo, variáveis das fases de teste e suporte impossibilitam a estimativa de esforço de manutenção dos sistemas.

O modelo ED^3M proposto por Haider *et al.* [6] permite, durante a fase de testes, estimar quantos defeitos um projeto de *software* apresentará depois de concluído. Esse modelo é baseado na Teoria das Probabilidades (*Estimation Theory*) e não necessita de conhecimento prévio como entrada, ou seja, não depende de dados históricos de projetos passados para realizar suas estimativas. Se por um lado não analisar históricos pode facilitar o processo de estimativas, por outro, os resultados obtidos podem não refletir precisamente situações reais, visto que o desempenho atual da organização pode ser visto como uma projeção de seus projetos passados. Além disso, as estimativas são calculadas somente com base em dados de testes, desconsiderando informações fundamentais provenientes de fases como, por exemplo, desenvolvimento e manutenção.

Em resumo, mesmo que o tópico a respeito de estimativas de custo tenha sido explorado em algumas investigações recentes, nenhum dos trabalhos citados permite prever custos de suporte em projetos de gerenciamento de TI a partir de informações produzidas nas fases de desenvolvimento e testes. Além disso, os trabalhos apresentados nessa seção usam abordagens simplistas (baseadas em um número limitado de variáveis) ou estáticas, abrindo mão da aprendizagem a partir de experiências passadas. Para tratar essas deficiências, na próxima seção é apresentada uma abordagem Bayesiana para gerar previsões de custos de suporte.

III. Solução Conceitual

Nossa solução concentra-se em fornecer uma maneira sistemática para realizar estimativas de custo de manuteção/suporte (ou, no sentido inverso, de desenvolvimento e/ou teste) a partir de dados históricos de projetos de gerenciamento de TI. A Figura 1 ilustra a base da nossa

solução, destacando seus principais componentes conceituais, pessoal envolvido e interações.

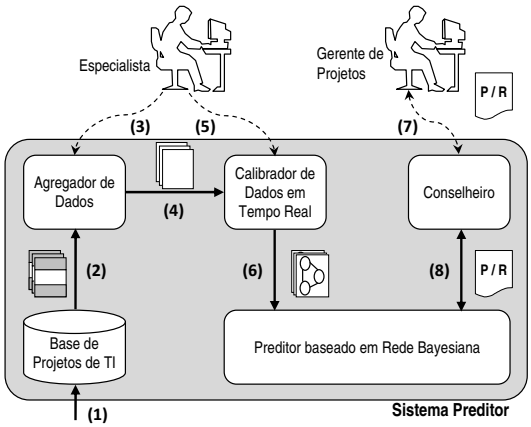


Figura 1. Elementos da solução proposta e suas interações

Dados brutos acerca de projetos de TI conduzidos na organização são continuamente coletados (fluxo 1 na Figura 1) e persistidos em uma *Base de Projetos*, segundo modelo de informação a ser apresentado na Subseção III-A. Esses dados brutos de cada projeto são, então, processados pelo componente *Agregador de Dados* visando a extração de informações-macro por projeto (ex: *Development/Test/Support*), que são armazenadas em variáveis (fluxo 2). Esse processo de agregação pode ser apoiado pela figura de um *Especialista* (fluxo 3), que pode determinar como popular as variáveis (privilegiando acurácia, eficiência, etc.) a partir dos dados brutos.

Como próximo passo da solução, os valores (das variáveis) associados aos projetos são repassados ao componente *Calibrador de Dados em Tempo Real* (fluxo 4 na Figura 1), este responsável por normalizar valores observados para cada variável e clusterizá-los em estados mutuamente exclusivos. A clusterização pode ser realizada de maneira automática ou semi-automática, nesse caso com intervenção do *Especialista* para apoiar o processo (fluxo 5). Tanto o processo de agregação quanto o de calibragem são explicados com mais detalhes na Subseção III-B.

O resultado da referida clusterização é empregado como parâmetro de entrada do componente *Preditor baseado em Redes Bayesianas* (fluxo 6 na Figura 1). Esse componente modela as variáveis (nós) e suas relações causais (arcos) por meio de um grafo acíclico. Ademais, para cada variável Y que possui como antecessores X_1, ..., X_n, existe uma tabela P(Y – X_1, ..., X_n). Caso Y não possua um nó antecessor, a tabela de probabilidades é reduzida para uma probabilidade incondicional P(Y). Esse modelo Bayesiano, descrito com detalhe na Subseção III-C, permite que, fixadas as probabilidades de alguns nós (hipóteses), sejam computadas as demais probabilidades que se deseja (estimativas). Um gerente de projeto interessado em utilizar a solução para realizar estimativas interagirá com o componente *Conselheiro* (fluxo 7), *front-end* gráfico que contatará o motor de inferência (fluxo 8) repassando-lhe consultas do *Gerente de Projetos* (*i.e.*, conjunto de hipóteses) e retornando ao gerente estimativas computadas.

Tendo apresentado uma visão geral da solução proposta, as próximas subseções têm por objetivo detalhar (*i*) o modelo de informação de projetos; (*ii*) o processo de agregação e normalização de dados de projetos; e (*iii*) o modelo Bayesiano para gerar estimativas de custos de suporte.

A. IT Project Information Model

Conforme apresentado na solução conceitual, a previsão de custos de suporte requer acesso a informações detalhadas sobre o ciclo de vida (ex: composição das atividades, recursos envolvidos, tempo consumido) de um conjunto de projetos realizados dentro da organização. Com o objetivo de representar essas informações, propõe-se um modelo que agrega classes oriundas do *Common Information Model* (CIM) [7] e da *Workflow Management Coalition Specification* [8]. O modelo também incorpora classes que materializam e mantém informações oriundas da observação de sistemas de acompanhamento de projetos, como o *HP Quality Center* [9]. A Figura 2 ilustra uma visão parcial do modelo proposto.

Uma instância da classe *ITProject*, ponto de partida do modelo, representa um projeto, *i.e.*, um esforço temporário empregado para criar um produto encomendado a partir das necessidades do usuário (*i.e.*, cliente ou usuário final). Todo projeto de TI pode ser entregue ao usuário por meio de uma ou mais *releases* (instâncias da classe *Release*). Cada *release* corresponde a uma versão parcial do produto ou serviço que está sendo desenvolvido e, por sua vez, pode ser concluída por meio de uma ou mais iterações (*Iteration*). Por fim, uma iteração é composta por um ou mais ciclos (*Cycle*), tais como análise, planejamento, desenvolvimento e teste. As classes recém descritas fazem parte do pacote *Development* ilustrado na Figura 2.

Com o objetivo de organizar as atividades realizadas em cada ciclo, um ou mais planos (instâncias da classe *WorkPlan*) são definidos. Um plano é um *workflow* de atividades seguindo a definição proposta pela WfMC [8]. Tomando como exemplo o ciclo de testes associado a uma dada iteração/*release* de um projeto de desenvolvimento de uma aplicação Web, atividades (*Activity*) e transições (*Transition*) entre elas modelariam um plano (*WorkPlan*) ordenado de testes. Instâncias da classe *Activity* sempre possuem participantes associados. Esses são representados pela classe *ParticipantSpecification* e podem ser recursos humanos e/ou materiais empregados nas atividades. Tais recursos são mapeados a partir de classes do CIM [7]. Ao mesmo tempo, atividades podem tanto produzir quanto consumir artefatos (*Artifact*). Exemplos de artefatos são documentos de requisitos (*RequirementDocument*) e de teste (*TestDocument*). Além de permitirem a documentação sis-

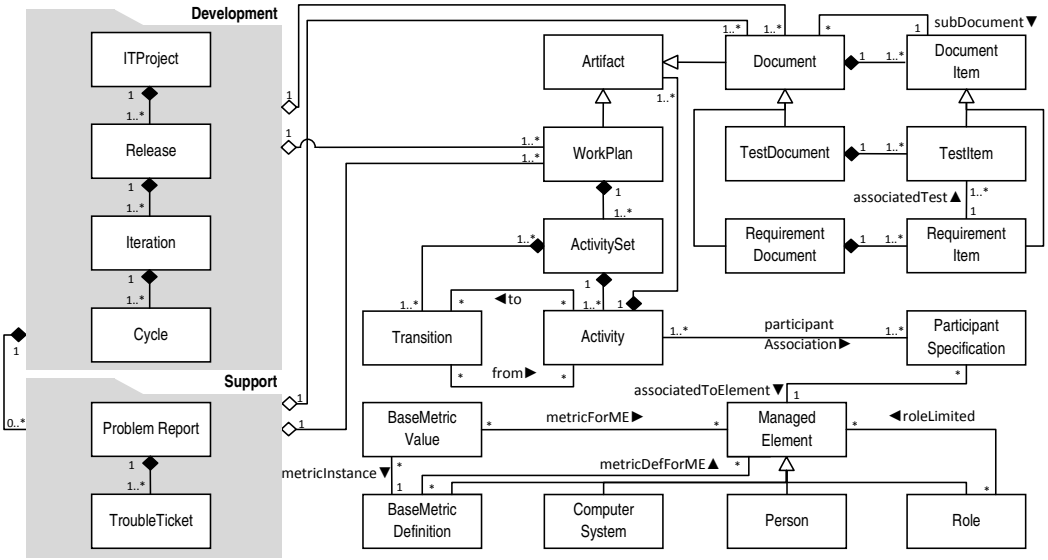


Figura 2. Visão parcial do modelo de informação para persistência de dados do ciclo de vida de projetos de gerenciamento de TI

temática do projeto, esses artefatos também permitem o compartilhamento de informações entre fases do ciclo de vida. No exemplo específico, itens de teste (documentados por meio de instâncias da classe *TestItem*) estão associados à validação de um ou mais requisitos (*RequirementItem*).

O modelo representado na Figura 2 permite, ainda, a persistência de informações a respeito de ações de suporte, que podem ser de dois tipos distintos. O primeiro consiste na identificação de um incidente (instância da classe *Problem-Report*), cuja resolução compreende a abertura de um bilhete (*TroubleTicket*) e a execução de um plano de contorno simples (*WorkPlan*). Esse plano envolve a orientação, por parte da equipe de suporte (*Person*), para a execução de um conjunto de atividades corretivas (*Activity*). O segundo tipo de ação de suporte também é iniciado pelo registro de um incidente e pela abertura de um bilhete. Contudo, devido a sua natureza/gravidade, é alçada a problema. Sua resolução demanda o desenvolvimento de *patches* de correção, podendo, ocasionalmente, caracterizar um novo projeto associado (*ITProject*).

B. Agregação e Clusterização de Dados de Projetos

Como já mencionado, a sumarização de dados de projetos em variáveis é realizada pelo componente *Data Aggregator*. Ainda que a solução conceitual não tenha por objetivo fixar um conjunto único de variáveis, algumas são intuitivamente importantes para as estimativas propostas neste trabalho. Cita-se, como exemplo, tamanho do projeto e tamanho das equipes. O processo de agregação consiste em percorrer cada instância de projeto e calcular, observando diversos objetos do modelo, valores a essas variáveis. No caso específico da variável tamanho da equipe de suporte, por exemplo, é necessário percorrer todas as atividades (instâncias da classe *Activity*) dos planos (*WorkPlan*) vinculados a bilhetes de cada projeto (*TroubleTicket*) e contabilizar todos os humanos envolvidos (*Person*). Demais variáveis são valoradas de forma análoga.

Em complementação ao processo de agregação, o componente *Real Time Model Calibrator*, após os valores já estarem associados às variáveis de cada projeto, executa (para cada variável encontrada) um procedimento de clusterização desses valores. Como resultado, cada variável passa a ser representada por um conjunto de estados (modelando diferentes naturezas ou grandezas). Voltando ao exemplo da variável tamanho da equipe de suporte, o procedimento poderia gerar, como resultado, os estados *pequeno*, *médio* e *grande*. Vale lembrar que o procedimento de clusterização pode ser conduzido usando algoritmo para esse fim, como *k-means*, *fuzzy c-means* e classificação *naive Bayes* [10] ou valendo-se da experiência de um especialista.

Como resultado do procedimento de clusterização, tem-se as variáveis incondicionais do modelo de estimativa devidamente alimentadas (representando o conhecimento *a priori*). Ainda como parte do processo de calibragem, analisa-se relações existentes entre as variáveis e estabelece-se um grafo de influências, podendo este ser entendido como o esqueleto do modelo Bayesiano de estimativa. Esse modelo é explicado na próxima subseção.

C. Modelo Bayesiano

O modelo Bayesiano para estimativas de custos de suporte proposto neste trabalho é composto por dezesseis variáveis, organizadas em três grupos: *desenvolvimento*, *teste* e *suporte*. A Figura 3(a) ilustra o modelo, destacando esses grupos em cinza claro, cinza escuro e branco, respectivamente. Variáveis como *StaffProd*, *StaffSize*, *ProjectSize* e *Time* são representadas nos três grupos, enquanto outras, dada sua especificidade, aparecem apenas em um ou outro grupo (ex:

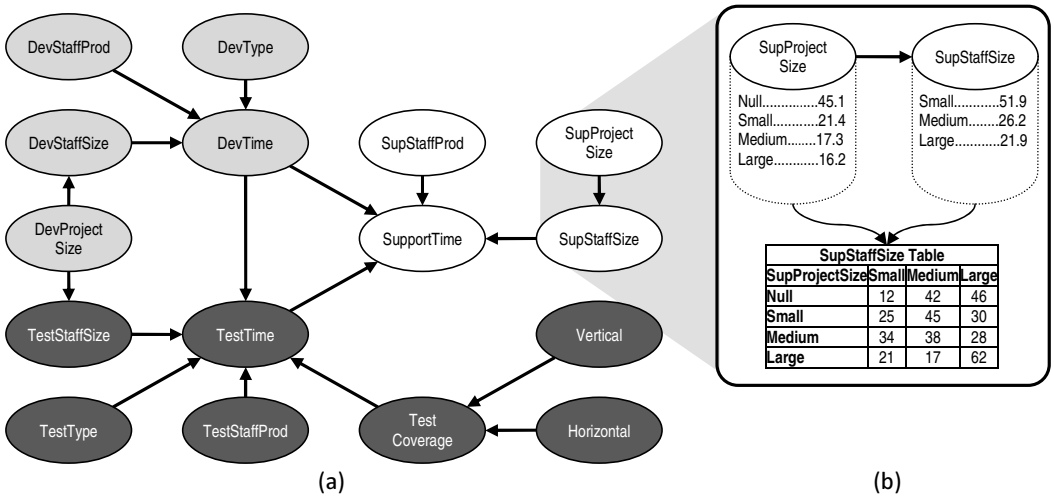


Figura 3. Modelo Bayesiano para estimativa de custos de desenvolvimento, teste e suporte

TestCoverage). As ligações entre as variáveis, representadas na figura por meio de arcos orientados, expressam relações causais entre elas. Lembrando, tais ligações são estabelecidas no processo de calibragem.

A configuração do modelo recém mencionado foi delineada tendo como base (*i*) opiniões de especialistas, (*ii*) análise de uma base com mais de 5.000 projetos disponibilizados pelo *International Software Benchmarking Standards Group* (ISBSG) [11] e (*iii*) observação de variáveis empregadas em investigações correlatas [5]. Ressalta-se, ainda, que o modelo é aberto, podendo ser *simplificado*, priorizando reduzir processamento, ou *estendido*, visando à captura de aspectos (variáveis e relações) não contemplados.

Voltando ao modelo, as variáveis incondicionais, populadas no processo de calibragem, influenciam direta ou indiretamente o cálculo de probabilidades dos estados das variáveis condicionais. Este é o caso, por exemplo, da variável *SupProjectSize* em relação à variável *SupStaffSize*. Como pode ser observado na Figura 3(b), *SupStaffSize* modela uma tabela de probabilidades que associa os seus estados com os de *SupProjectSize* (variável antecessora). Uma célula da tabela representa a probabilidade de *SupStaffSize* (X=pequeno, médio, grande) ocorrer em *SupProjectSize* (Y=*null*, pequeno, médio, grande). De forma análoga, as demais variáveis condicionais possuem suas próprias tabelas.

Com o modelo devidamente populado, é possível realizar consultas *what-if* (fixando estados de uma ou mais variáveis) e obter estimativas (observando o efeito provocado nas demais). No contexto do trabalho, passa a ser possível responder perguntas como: *conhecido tempo/esforço despendidos no desenvolvimento e teste de um projeto, qual a expectativa de custos de suporte?* Ou, ainda, *quanto investir em desenvolvimento e teste dado um limite máximo tolerável de custos de suporte?* As estimativas produzidas pelo modelo serão probabilidades associadas aos estados das variáveis.

IV. Avaliação

Para avaliar a viabilidade técnica da solução proposta, foi desenvolvido o $UPPORT, um sistema de predição de custos de desenvolvimento, de testes e de suporte. Esse sistema foi utilizado tanto para gerar uma avaliação qualitativa quanto quantitativa da nossa solução. Nesta seção será descrito o sistema $UPPORT, a metodologia de avaliação empregada e os resultados obtidos.

Na avaliação considerou-se, como estudo de caso, a predição de custos associados com projetos de desenvolvimento Web (também referenciados como PWeb ao longo desta seção). Outros projetos, omitidos neste trabalho por limitação de espaço, também foram analisados e serão discutidos em trabalhos futuros.

A. O Sistema $UPPORT

O sistema $UPPORT consiste em um protótipo que implementa o modelo apresentado na seção anterior. Desenvolvido em *Java* com auxílio de uma API chamada *JavaBayes* (conjunto de ferramentas para programação de redes Bayesianas) [12], o sistema permite realizar estimativas acerca de custos, esforço e/ou tempo de um dado projeto. A Figura 4(a) ilustra a aba *Profiles*, em que se observa as variáveis e estados de um projeto PWeb. Lembre-se que o projeto é organizado de acordo com o modelo de informação apresentado na Seção III-A, sendo organizado em *releases*, *iterações* e *ciclos*. A ideia é que dados históricos de projetos passados possam ser importados de sistemas como *HP Quality Center*, sem (ou com pouca) intervenção do usuário. De posse dessas informações, $UPPORT tem condições de extrair, do projeto em análise, valores (no caso, informações já computadas) para estados de algumas das variáveis do modelo.

Além disso, o usuário pode informar, diretamente em *Prediction*, valores arbitrários (nesse caso, hipóteses) a estados das variáveis do modelo. Tal é ilustrado na Figura 4(b), onde, para o projeto em análise, cria-se a hipótese de que o tempo

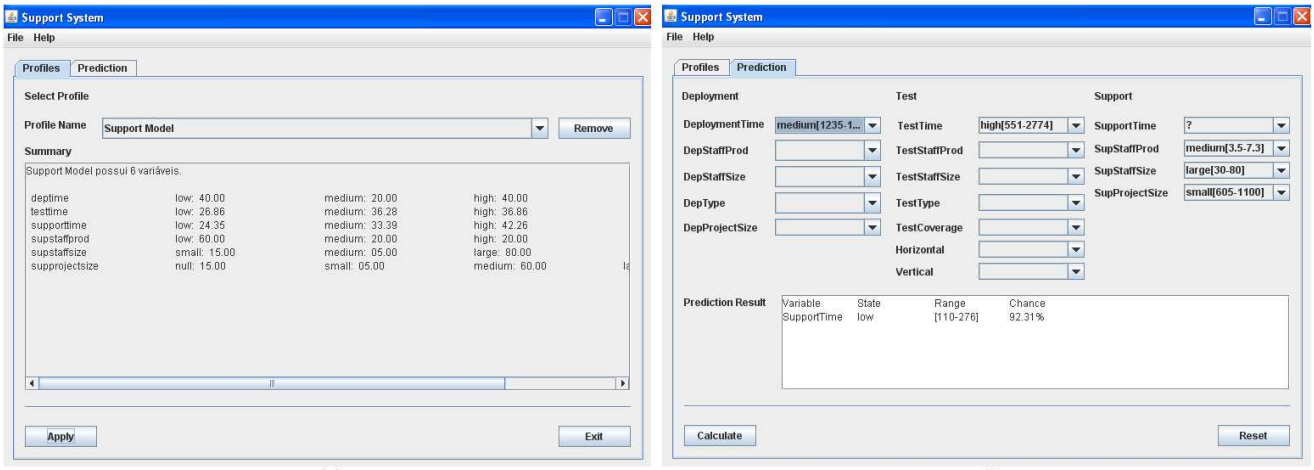


Figura 4. Interface gráfica do sistema $UPPORT

de desenvolvimento é *médio*. Como resultado, $UPPORT recalcula as probabilidades dos estados das variáveis (do modelo) que estão em aberto. No exemplo apresentado na Figura 4, o sistema informa que, com 92,31% de chance, o tempo a ser despendido em suporte no projeto (PWeb) será *pequeno*. A partir das estimativas, o sistema também apresenta uma faixa de valores associada ao estado de maior probabilidade. Voltando ao exemplo, a caixa de texto *Prediction Result* exibe como resultado o intervalo de 110 - 276 horas gastas com suporte.

B. Metodologia

A avaliação foi realizada por meio de duas abordagens distintas, mas complementares. Em ambas as abordagens, um conjunto de perguntas *what-if* sobre a instanciação de projetos PWeb foi elaborado e submetido ao sistema $UPPORT, a fim de obter como resultado a estimativa de uma variável deixada como indefinida. Na primeira abordagem, os resultados obtidos foram comparados com a opinião de um segundo grupo de especialistas em projetos de TI (diferente do grupo que auxiliou na classificação dos projetos, mencionado na Subseção III-C). Já na segunda abordagem, os resultados obtidos pela nossa solução foram comparados aos produzidos por um modelo Bayesiano criado automaticamente por meio do *software* Genie 2.0 [13].

A obtenção e a preparação da base de dados de projetos foi uma tarefa importante da avaliação. A partir dos dados do *dataset* do ISBSG, foram selecionados mais de 200 projetos de nosso interesse, para extrair parâmetros-chave, tais como faixas de valores reais para as variáveis do modelo (por exemplo, *DevTime*, *TestTime* e *SupportTime*). Além disso, olhando para esses valores (de cada projeto), foi possível clusterizá-los e consolidar os estados das variáveis. A Tabela I resume o resultado desta etapa. Pegando *DevTime* (tempo de desenvolvimento/implementação) como exemplo, essa variável foi organizada em três estados: *baixo* [0 - 1.234 horas], *médio* [1.235 - 1.968 horas] e *alto* [1.969 - 5.000 horas].

Tabela I
VARIÁVEIS DO MODELO BAYESIANO E SEUS ESTADOS CORRESPONDENTES

Variável	**Estados***
DevTime	low [0-1.234], medium [1.235-1.968], high [1.969-5.000]
TestTime	low [0-300], medium [301-550], high [551-2.774]
SupStaffSize	small [1-8], medium [9-29], large [30-80]
SupStaffProd	low [0-3,4], medium [3,5-7,3], high [7,4-12,0]
SupportTime	low [110-276], medium [277-800], high [801-1.500]
DevStaffProd	low, medium, high
DevType	innovation, improvement, workaday
DevStaffSize	small, medium, large
DevProjectSize	small, medium, large
TestStaffProd	low, medium, high
TestType	manual, automatic
TestStaffSize	small, medium, large
TestCoverage	small, medium, large
Vertical	low, medium, high
Horizontal	low, medium, high
SupProjectSize	null(0), small, medium, large

**DevTime*, *TestTime*, e *SupportTime* são medidos em horas. *SupStaffSize* é medido em número de humanos, e *SupStaffProd* em ManPower/hora.

Idealmente, o objetivo era usar os projetos disponíveis no *dataset* do ISBSG como entrada para o modelo Bayesiano. No entanto, tal não foi possível porque a maioria dos projetos tinha um conjunto incompleto de dados disponíveis (valores das variáveis) e também porque os projetos de TI não estavam relacionados com suas respectivas fases de suporte. Para superar essa limitação, foi gerada uma base de dados sintética de projetos agrupando os dados reais

conhecidos com informações fornecidas por um grupo de especialistas na gerência de projetos de TI, como explicado a seguir. Primeiro, foram selecionadas cinco variáveis (*DevTime*, *TestTime*, *SupStaffSize*, *SupStaffProd* e *SupportTime*) e gerada uma lista com todas as combinações possíveis entre seus estados, desde *baixo, baixo, pequena, baixo, baixo* até *alta, alta, grande, alta, alta*. Cada combinação representa uma instância de projeto de TI, também chamada de *cenário* ao longo deste trabalho. Na sequência, foi solicitado aos especialistas que, com base em sua experiência, classificassem cada um dos cenários como *provável*, *possível* ou *improvável* e, em seguida, foram criados três conjuntos de projetos distintos conforme essa classificação. Finalmente, foi escolhida uma proporção variada de cenários a partir desses conjuntos, dependendo do perfil desejado da base de dados e, então, povoadas tanto as variáveis incondicionais, quanto as variáveis condicionais do modelo Bayesiano.

A fim de analisar a precisão das predições, o modelo Bayesiano foi populado com informações coletadas a partir de 710 cenários de projetos, na proporção de 70% (provável), 20% (possível) e 10% (improvável). Esse perfil representa o caso em que 70% dos projetos conduzidos por uma organização apresenta um comportamento esperado, enquanto os outros 30% se comportam de maneira irregular e, por vezes, de forma anormal. Essas proporções foram propostas através do estudo analítico do *dataset* do ISBSG e confirmadas pelos especialistas.

A avaliação da solução proposta foi conduzida por meio de 50 perguntas submetidas ao sistema $UPPORT. Na primeira abordagem da avaliação, os resultados obtidos foram confrontados com a opinião de um grupo de especialistas. Na segunda abordagem, por outro lado, os resultados foram comparados com os produzidos por um novo modelo Bayesiano gerado automaticamente, de modo *bottom-up*, a partir da base de dados (caso ótimo). O novo modelo, ilustrado na Figura 5, foi gerado com o *software* Genie 2.0 [13], selecionando-se o algoritmo PC [14]. Em comparação com o modelo que propomos, esse introduz uma relação causal entre as variáveis *DevTime* e *SupStaffProd*. Além disso, algumas relações, destacadas em cinza na figura, são redefinidas.

C. Resultados e Discussão

A Tabela II ilustra um conjunto de 10 (do total de 50) questões submetidas aos sistemas $UPPORT e Genie, bem como a um grupo de especialistas. Em cada uma dessas questões foi marcado um estado para cada uma das quatro variáveis, e uma variável, que deveria ser prevista, foi deixada em aberto. Para ilustrar, considere o Cenário 1. Neste cenário, a questão colocada foi: *qual é o tempo necessário de suporte para a manutenção dos ativos entregues por um projeto PWeb específico, se para isso for despendido grande tempo para desenvolvimento e testes, e a organização possuir um grande número de humanos na equipe de suporte, cuja produtividade seja considerada média?* Em outras palavras: dado *DevTime = alto*, *TestTime = alto*, *SupStaffSize = grande* e *SupStaffProd = médio* para a implantação de um projeto PWeb específico, qual é o estado esperado para *SupportTime*?

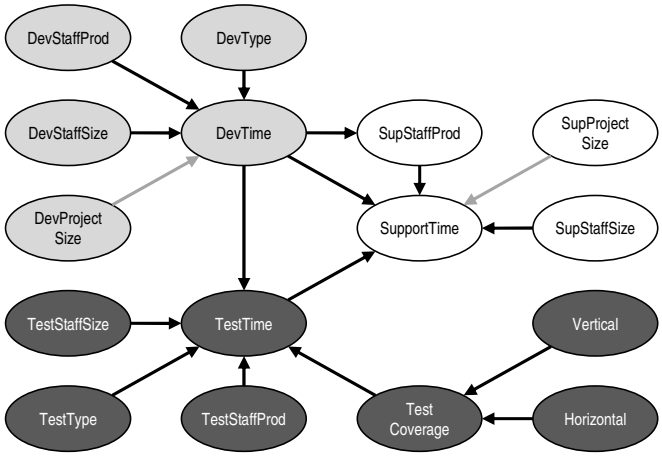


Figura 5. Modelo Bayesiano criado automaticamente através do Genie 2.0

Para a pergunta apresentada o sistema $UPPORT retornou como resultado o estado *baixo*. Isso significa que, para projeto PWeb em questão, é esperado um baixo tempo de suporte após a entrega final dos seus ativos. Os especialistas encontraram na atribuição de *baixo* para *SupportTime* uma boa estimativa. De forma análoga, o mesmo cenário foi experimentado no modelo Bayesiano criado automaticamente pelo *software* Genie. O resultado obtido confirmou as respostas do sistema $UPPORT e dos especialistas.

Agora considerando o Cenário 3, cuja pergunta foi: *qual deve ser o tamanho da equipe de suporte, se houver uma quantidade média de tempo disponível para o desenvolvimento do projeto, quantidade média de tempo destinado para a fase de testes, uma equipe de suporte com produtividade considerada média, e deseja-se gastar tempo baixo com a fase de suporte?* Dito de outro modo: dado *DevTime = médio*, *TestTime = médio*, *SupStaffProd = médio* e *SupportTime = baixo* para a execução de um projeto PWeb, qual o estado esperado para *SupStaffSize*?

O sistema $UPPORT retornou como resultado a mesma probabilidade para todos os estados. Esse resultado indica que, considerando as circunstâncias declaradas na pergunta apresentada e o histórico de projetos, cada estado possui 33,33% de chance de ocorrer. No entanto, os especialistas discordaram da estimativa fornecida pelo sistema, que, embora factível com o histórico de projetos, não representa uma boa resposta para a questão. Quando avaliado pelo modelo Bayesiano produzido pelo Genie, o cenário apontou *SupStaffSize = alto*, estimativa também discordante daquela produzida por $UPPORT.

Entre os cenários expostos na Tabela II, pode-se notar

Tabela II
QUESTÕES *what-if* PARA PREDIÇÃO DE CUSTOS DE PROJETOS PWEB.

#	Variáveis					Resultado do $UPPORT	Opinião dos Especialistas	Resultado do Genie
	DevTime	TestTime	SupStaffSize	SupStaffProd	SupportTime			
1	high	high	large	medium	?	low	Concordo	low
2	high	?	medium	high	low	high	Concordo	high
3	medium	medium	?	medium	low	*Any*	Discordo	high
4	?	medium	medium	medium	medium	low	Concordo	medium
5	low	medium	large	?	high	medium	Discordo	low
6	medium	high	small	?	high	low	Concordo	low
7	high	medium	?	low	high	medium	Concordo	medium
8	low	?	large	medium	medium	medium	Concordo	medium
9	low	high	large	low	?	high	Concordo	high
10	?	high	large	medium	high	low	Concordo	low

que a nossa solução foi capaz de estimar precisamente os resultados de oito entre dez perguntas (1, 2, 4, 6, 7, 8, 9 e 10) pelo ponto de vista dos especialistas em projetos de TI. Já se comparada com os resultados obtidos com o Genie, nossa solução foi capaz de estimar corretamente os resultados de sete perguntas (1, 2, 6, 7, 8, 9 e 10). Observando os resultados por uma perspectiva mais ampla, para as 50 consultas *what-if* analisadas, o sistema $UPPORT foi capaz de estimar com precisão os custos previstos e as condições necessárias para a execução de projetos PWeb em 82% dos cenários especificados, segundo a opinião dos especialistas. Pela visão do modelo Bayesiano gerado, $UPPORT apresentou um índice de acerto de 74%.

Como constatação geral, os resultados produzidos pela nossa solução são bastante satisfatórios. Mesmo com uma base contendo 30% de instâncias "ruidosas" de projetos, foi possível realizar predições com taxa de acerto entre 74 e 82%. Fosse a base de projetos mais uniforme, certamente os resultados seriam ainda mais favoráveis. Analisando nosso modelo em comparação com o produzido automaticamente (dito "ótimo"), vale ressaltar que o nosso (construído de forma *top-down* refletindo visão de especialistas) tem potencial para ser empregado em projetos de natureza distinta daquele explorado no artigo. Essa maior generalidade explica os resultados um pouco piores (26%) em relação ao ótimo, obtido de modo *bottom-up* a partir, exclusivamente, das instâncias de projeto PWeb.

D. Análise de Sensibilidade

Os experimentos discutidos anteriormente nesta seção foram realizadas considerando o perfil do conjunto de dados 70% (provável), 20% (possível) e 10% (improvável). Essas porcentagens refletem o comportamento de um conjunto de dados reais (de acordo com a opinião de especialistas e confirmado por meio de um estudo aprofundado do conjunto de dados ISBSG). Uma vez efetuada a avaliação experimental considerando tal conjunto de dados, repetimos os mesmos experimentos com outros conjuntos, que eram essencialmente caracterizados por proporções variáveis de projetos prováveis, possíveis e improváveis. Nosso objetivo foi observar o quão sensível é a precisão das estimativas obtidas com o sistema $UPPORT frente a diferentes perfis de conjuntos de dados.

A Tabela III ilustra a taxa de sucesso que $UPPORT obteve quando alimentado com diferentes perfis de bases de dados. Como se pode notar, as avaliações dos cenários considerando um perfil de base de dados onde a maioria dos projetos apresentou padrão semelhante de desenvolvimento/implantação resultou em estimativas mais precisas. Este é o caso do perfil 70%-20%-10%, o que proporcionou a maior taxa de sucesso (de 80% dos 50 cenários avaliados).

Tabela III
PRECISÃO OBTIDA COM OUTROS PERFIS DE BASES DE DADOS

Perfil de dados	Correto (%)	Incorreto (%)	Inconclusivo (%)
33-33-33	40	23	37
50-30-20	60	23	17
70-20-10	80	03	17

Em contraste, as avaliações com base em perfis mais amorfos (ou seja, perfis que possuem projetos com alto nível de disparidade no valor de suas variáveis) levaram ao aumento do número de estimativas incorretas. Este é o caso dos perfis de 50%-30%-20% (60% de taxa de sucesso) e 33%-33%-33% (com somente 40% de taxa de sucesso). Essas observações provam que o número de estimativas inconclusivas (aquelas que não oferecem qualquer informação relevante para o gerente que utiliza o sistema de previsão) crescem proporcionalmente ao aumento de número de projetos fora de padrão. Por outro lado, resultados mais precisos podem ser alcançados por perfis que exibem uma maior taxa de projetos semelhantes.

V. Considerações Finais

Neste artigo foi proposta uma solução que, aproveitando-se de dados do ciclo de vida de projetos de TI, é capaz de prever os custos atrelados à fase de suporte. Na verdade, como mostrado ao longo do artigo, a nossa solução baseada em redes Bayesianas vai além, permitindo também previsões, no sentido inverso, associadas com as fases de desenvolvimento e testes. Uma análise comparativa entre as estimativas obtidas no artigo com a opinião de especialistas em projetos de gerenciamento de TI constatou que a nossa solução é capaz de prever custos com elevado grau de confiança. Tal foi confirmado, também, pelo modelo Bayesiano criado automaticamente.

É importante mencionar que os bons resultados da solução consideraram uma base consistente (em termos de qualidade de perfis) e rica (em termos de instâncias) de projetos anteriores de TI. No entanto, acreditamos que esses requisitos não limitem a aplicabilidade de nossa solução por três razões. Primeiro, há uma tendência crescente de que organizações (especialmente as de médio e grande porte) empreguem boas práticas e processos propostos por *frameworks* de gerência de projetos. Segundo, organizações com um determinado nível de maturidade já adotam ferramentas computacionais de gerenciamento (de prateleira ou personalizadas) de projetos de TI. E terceiro, a abordagem Bayesiana viabiliza o uso da experiência de especialistas (caráter qualitativo), o que pode ser muito útil em ambientes com ausência de dados.

Uma discussão que merece atenção diz respeito aos resultados obtidos com os dois modelos Bayesianos apresentados no trabalho. Na comparação com a opinião de especialistas em projetos de TI, o modelo do sistema $UPPORT apresentou ligeira vantagem sobre o modelo Bayesiano do *software* Genie. Essa vantagem pode ser explicada pela sensível diferença na topologia das duas redes. No modelo $UPPORT o tamanho do projeto (*DevProjectSize* e *SupProjectSiza*) influencia no tempo, mas a propagação de sua influência é moderada pelo tamanho da equipe (*DevStaffSize* e *SupStaffSize*). Isso significa que os resultados julgados pelos especialistas corroboram com nossa visão de que a relação de causa-efeito entre tamanho e tempo é subordinada a uma variável intermediária (no nosso caso *DevStaffSize* e *SupStaffSize*). Por outro lado, no modelo proposto pela ferramenta Genie o tempo do projeto (*DevTime* e *SupportTime*) possui relação de dependência direta com tamanho do projeto e tamanho da equipe. A partir disso, é possível chegar a uma constatação de relevância prática sobre os esforços em estimativas de projeto, na qual o tamanho do projeto é um fator importante e que deve ser considerado. Porém, a relação de causa-efeito entre tamanho e tempo deve ser mediada por outro fator, que no contexto deste estudo foi o tamanho da equipe.

Observe também que, em relação à granularidade das previsões geradas, estimativas mais detalhadas podem ser obtidas pela agregação dos valores observados (para cada variável do modelo Bayesiano) em um número maior de estados mutuamente exclusivos. Em investigações futuras pretendemos explorar formas de apresentar resultados mais detalhados das estimativas geradas. Outras direções de pesquisa incluem: (*i*) avaliação da solução proposta considerando traços gerados por ferramentas de gerenciamento de projetos de TI e (*ii*) investigação de outras variáveis relacionadas ao ciclo de vida de projetos de TI, a fim de analisar e prever outros aspectos.

Referências

[1] PMBOK, *A guide to the project management body of knowledge : (PMBOK guide)*, 4th ed. Newtown Square, PA: Project Management Institute, Inc., 2008.

[2] ITIL, "Office of Government Commerce (OGC). Information Technology Infrastructure Library." Office of Government Commerce (OGC), 2010, [Online]. Disponível em: http://www.itil-officialsite.com/.

[3] B. W. Boehm, *Software Engineering Economics*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1981.

[4] B. W. Boehm and R. Valerdi, "Achievements and Challenges in Cocomo-Based Software Resource Estimation," *IEEE Software*, vol. 25, pp. 74–83, 2008.

[5] E. Mendes and N. Mosley, "Bayesian Network Models for Web Effort Prediction: A Comparative Study," *IEEE Transactions on Software Engineering*, vol. 34, pp. 723–737, 2008.

[6] S. W. Haider, J. W. Cangussu, K. M. Cooper, R. Dantu, and S. Haider, "Estimation of Defects Based on Defect Decay Model: ED^3M," *IEEE Transactions on Software Engineering*, vol. 34, pp. 336–356, 2008.

[7] CIM, "Distributed Management Task Force. Common Information Model," 2007, [Online]. Disponível em: http://www.dmtf.org/standards/cim.

[8] WfMC, "Workflow Process Definition Interface - XML Process Definition Language," 2007, the Workflow Management Coalition Specification. [Online]. Disponível em: http://docs.oasis-open.org/wsbpel/2.0/.

[9] Hewlett-Packard, "Quality center," February 2009, [Online]. Disponível em: http://h50281.www5.hp.com/software/index.html. [Online]. Available: http://h50281.www5.hp.com/software/index.html

[10] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, 2nd ed. Wiley-Interscience, 2000.

[11] ISBSG, "International Software Benchmarking Standards Group Dataset. Release 10," 2007, [Online]. Disponível em: http://www.isbsg.org/.

[12] F. Cozman, "Javabayes: Bayesian Networks in java," *http://www.cs.cmu.edu/javabayes*, 2001.

[13] Genie and Smile Systems, "The Decision Systems Laboratory of the University of Pittsburgh," Agosto 2010, [Online]. Disponível em: http://genie.sis.pitt.edu/.

[14] P. Spirtes, C. Glymour, and R. Scheines, *Causation, Prediction, and Search*. The MIT Press, 2000.