

Pontifícia Universidade Católica do Rio Grande do Sul
Faculdade de Informática
Programa de Pós-Graduação em Ciência da Computação

**Avaliação de Qualidade de Dados
de Métricas de Esforço Baseado em
Data Provenance e *Fuzzy Logic***

Rita Cristina Galarraga Berardi

**Dissertação apresentada como
requisito parcial à obtenção do grau
de mestre em Ciência da Computação**

Orientador: Prof. Dr. Duncan Dubugras Alcoba Ruiz

Porto Alegre
2009

Dados Internacionais de Catalogação na Publicação (CIP)

B483a Berardi, Rita Cristina Galarraga
Avaliação de Qualidade de Dados de Métricas de
Esforço Baseado em Data Provenance e Fuzzy Logic /
Rita Cristina Galarraga Berardi. Porto Alegre, 2009.
134 f.

Diss. (Mestrado) – Fac. de Informática, PUCRS
Orientador: Prof. Dr. Ducan Dubugras Ruiz

1. Qualidade de Software. 2. Qualidade de dados
(Informática). 3. Lógica Difusa. 4. Informática. I. Ruiz,
Duncan Dubugras. II. Título.

CDD 006.32

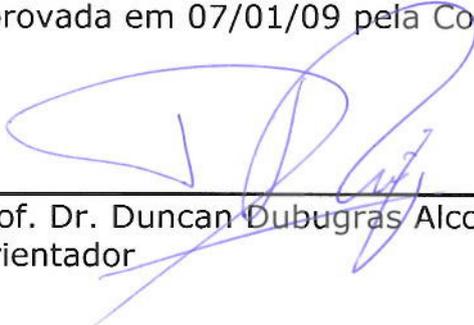
**Ficha Catalográfica elaborada pelo
Setor de Tratamento da Informação da BC-PUCRS**



Pontifícia Universidade Católica do Rio Grande do Sul
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

TERMO DE APRESENTAÇÃO DE DISSERTAÇÃO DE MESTRADO

Dissertação intitulada "**Avaliação de Qualidade de Dados de Métricas de Esforço Baseado em *Data Provenance e Fuzzy Logic***", apresentada por Rita Cristina Galarraga Berardi, como parte dos requisitos para obtenção do grau de Mestre em Ciência da Computação, Sistemas de Informação, aprovada em 07/01/09 pela Comissão Examinadora:


Prof. Dr. Duncan Dubugras Alcoba Ruiz -
Orientador

PPGCC/PUCRS

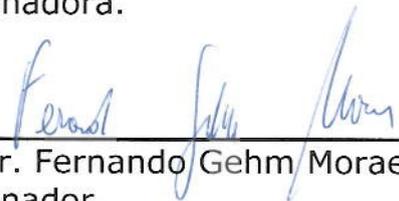

Prof. Dr. Jorge Luis Nicolas Audy -

PPGCC/PUCRS


Profa. Dra. Renata de Matos Galante -

UFRGS

Homologada em...02/06/09..., conforme Ata No. 609... pela Comissão Coordenadora.


Prof. Dr. Fernando Gehm Moraes
Coordenador.

PUCRS

Campus Central

Av. Ipiranga, 6681 - P32 - sala 507 - CEP: 90619-900

Fone: (51) 3320-3611 - Fax (51) 3320-3621

E-mail: ppgcc@pucrs.br

www.pucrs.br/facin/pos

Dedico mais esta vitória à melhor mãe do mundo e que por circunstâncias da vida também se transformou no melhor pai do mundo!

Agradecimentos

Agradeço primeiramente a Deus que me deu coragem e força para vir lutar pelos meus objetivos, longe de casa e do colo da mamãe. Obrigada Pai pelas tantas bênçãos que me proporcionaste nesses últimos dois anos.

Agradeço à minha mãe pelo constante apoio e incentivo para seguir meu caminho, mesmo que a saudade entalasse na garganta cada vez que eu pegava o ônibus de volta para Poa. Obrigada Mãe por ser um exemplo que eu posso seguir!

Ao Rodrigo, pela paixão, pelo amor e companherismo que foram peças fundamentais nessa caminhada. Teu carinho e dedicação foram de extrema importância nos momentos bons e mais ainda naqueles difíceis, em que parece que nada vai dar certo. Obrigada pelo ombro, pela inspiração, pelas idéias. Obrigada pela tua constante presença mesmo quando não estavas fisicamente por perto!!

Ao meu mais novo amigo Nelson, que me acolheu quando eu cheguei em Poa e que pouco a pouco foi se transformando em um amigo muito especial. Obrigada pela força, pela companhia e pelos momentos de muita risada que ajudaram a amenizar os problemas de uma etapa como essa. Espero que esta amizade perdure e que supere as distâncias físicas.

Ao meu orientador, professor Duncan, por ter acreditado em mim desde a seleção até o último momento antes da defesa. Obrigada pelas lições tanto acadêmicas, quanto de vida. Obrigada por toda preocupação e compreensão.

Ao professor Jorge Audy, que apesar de possuir uma agenda lotada, sempre esteve disponível para contribuir com o trabalho tanto no sentido de desenvolvimento quanto de avaliação. Obrigada por todo aprendizado proporcionado neste período de convivência.

Ao Convênio HP EAS Brasil/PUCRS por viabilizarem a bolsa de estudos bem como todo o aprendizado fornecido durante esse tempo.

Ao grupo GPIN (Grupo de Pesquisa em Inteligência de Negócio) pelo conhecimento trocado.

Ao Programa de Pós-Graduação em Ciência da Computação e a todos os professores que contribuíram com a minha formação tanto em sala de aula quanto em convivência extra-classe.

E agradeço a todas as pessoas que me acompanharam e contribuíram para que este período se tornasse em mais uma vitória na minha vida.

Resumo

Cada vez mais as organizações de software estão preocupadas com melhoria do seu processo e conseqüentemente do seu produto. Para isso, as organizações utilizam modelos de maturidade, os quais indicam a coleta de métricas para o controle de seus processos. No entanto, o esforço com relação a essas métricas está relacionado à sua intensa coleta e utilização e não é dada a devida atenção à qualidade dos dados das mesmas. O impacto da falta de qualidade dos dados dessas métricas é refletido diretamente nos custos da organização visto que as métricas embasam o processo de tomada de decisão o qual pode ser de baixa confiabilidade devido os seus dados de base também o serem. Uma avaliação adequada da qualidade desses dados é o primeiro passo para garantir que as métricas possam ser usadas com a devida confiabilidade. Uma abordagem que pode auxiliar essa avaliação está relacionada ao uso de *data provenance* (proveniência de dados) associado a um mecanismo de inferência lógica.

Este trabalho propõe uma arquitetura para avaliação da qualidade de dados de esforço composta por quatro principais componentes: 1-uma base de data provenance de métricas, 2-um modelo de inferência baseado em fuzzy logic, 3-uma base de dados para armazenamento de avaliações e 4- um modelo analítico para análise de histórico de qualidade de dados de esforço. A contribuição deste trabalho é prover uma avaliação da qualidade dos dados de métricas de esforço em PDS, buscando evidenciar as razões da eventual baixa qualidade. Através do modelo de inferência, é possível atribuir níveis de qualidade aos dados possibilitando assim a identificação daqueles que são efetivamente úteis para um processo de tomada de decisão confiável. Além disso, de acordo com seus níveis de qualidade, os dados podem ser direcionados para diferentes tipos de acompanhamento do projeto, cujos níveis de exigência de qualidade podem ser distintos.

Abstract

Increasingly, software organizations are concerned about improving the product and the process by which the product was made. For that, organizations use models of maturity, which indicates the collection of metrics to control their processes. However, in most of the time they make effort in collecting and storing these data but do not pay attention on their quality. Since making decision process is based on metric, if this metric is not reliable consequently the making decision process will not be. In this sense, a proper assessment of the quality of such data is the first step to ensure that the metrics can be used in sufficient reliability. An approach that can assist this assessment is related to the use of data provenance associated with a mechanism of logical inference. This research proposes an architecture for assessing the quality of data effort composed of four main components: 1-a metric provenance database, 2- a model of inference based on fuzzy logic, 3-a database for storing ratings of quality and 4 - an analytical model for analysis of historical data of quality of effort. The contribution of this work is to provide an assessment of data quality metrics of effort in Software Development Process, searching evident the reasons for a low quality. Having the model of inference it is possible assigning levels of quality to the data, and thus enabling the identification of those who are actually useful to a decision making trust.

Lista de Tabelas

Tabela 2.1	Níveis de maturidade CMMI- representação em estágios. Fonte:[Institute, 2002]	21
Tabela 2.2	Níveis de competência CMMI – representação contínua. Fonte: [Institute, 2002]	23
Tabela 2.3	Categorias e dimensões de DQ. Fonte:[Strong et al., 1997]	31
Tabela 2.4	Conjunto de operações fuzzy	39
Tabela 2.5	Exemplo de processamento de regras	39
Tabela 2.6	Principais diferenças entre técnicas de DP.Fonte: [Berardi, 2007]	45
Tabela 3.1	Tabela de avaliações	52
Tabela 3.2	Tabela de resultados avaliação 1	53
Tabela 3.3	Tabela de resultados avaliação 2	53
Tabela 4.1	Trabalhos Relacionados	61
Tabela 4.2	Trabalhos relacionados - segundo grupo	63
Tabela 5.1	Dimensões de qualidade	69
Tabela 5.2	Variáveis de DP para cada dimensão	71
Tabela 5.3	Componente de Qualidade	89
Tabela 6.1	Exemplo de #DR-DL	96
Tabela 6.2	Exemplo de #DFF-DR	97
Tabela 6.3	Exemplo de #VO-VM	98
Tabela 6.4	Exemplo de #DC-DA	98
Tabela 6.5	Preparação de dados com qualidade inferior a Boa (≤ 7)	100
Tabela 6.6	Comparação dos índices de erro do modelo preditivo 1 e do modelo preditivo 2	101
Tabela 6.7	Comparação dos índices de erro do modelo preditivo 1 e do modelo preditivo 3	102
Tabela 6.8	Teste estatístico MMRE e PRED(25)	103
Tabela 6.9	Tabela mês Março, equipe Server	104
Tabela 6.10	Tabela de qualidade do mês Fevereiro, por projeto e dimensão	105
Tabela 6.11	Tabela de média de qualidade por equipe no mês de Fevereiro	105
Tabela 6.12	Tabela de qualidade total do ano de 2007	106
Tabela 6.13	Tabela de qualidade total do ano de 2007 por dimensões	107
Tabela 7.1	Comparação final entre os trabalhos	110

LISTA DE TABELAS

Tabela 7.2 Análise da arquitetura com relação aos requisitos da solução . . . 110

Lista de Figuras

Figura 2.1	CMMI na representação em estágios. Fonte: [Institute, 2002]	21
Figura 2.2	CMMI na representação contínua. Fonte: [Institute, 2002]	22
Figura 2.3	Problemas de "Intrinsic DQ" Fonte:[Strong et al., 1997]	32
Figura 2.4	Diferença entre validade e confiabilidade. Fonte:[Kan, 2003]	34
Figura 2.5	Exemplo de cálculo da variável Distância. Fonte: [Altrock, 1995]	39
Figura 2.6	Diagrama de um modelo de inferência Mamdani. Fonte: [Rezende, 2003]	40
Figura 2.7	Método de defuzzyficação Centro de Gravidade para a variável Força. Fonte: [Altrock, 1995]	41
Figura 2.8	Taxonomia de Data Provenance. Fonte: [Yogesh L. Simmhan, 2005]	42
Figura 3.1	Fluxo das métricas	48
Figura 5.1	Solução proposta	65
Figura 5.2	Diagrama de classes UML do Componente de Proveniência	67
Figura 5.3	Lista de variáveis de DP na entrevista	70
Figura 5.4	Exemplo de entrada do Componente de Inferência	72
Figura 5.5	Componente de Inferência Fuzzy	73
Figura 5.6	Exemplo de resultado de avaliação do Componente de Inferência	74
Figura 5.7	Parte da entrevista	74
Figura 5.8	Exemplo de marcação pelo entrevistado	75
Figura 5.9	Exemplo de função de pertinência para acurácia	76
Figura 5.10	Modelo Mamdani da dimensão Acurácia	77
Figura 5.11	Função de Pertinência da Variável #DR-DL	78
Figura 5.12	Função de Pertinência da Variável #DFF-DR	78
Figura 5.13	Função de Pertinência da Variável #VO-VM	79
Figura 5.14	Função de Pertinência da Variável de saída Acurácia	79
Figura 5.15	Regras de produção da dimensão Acurácia	79
Figura 5.16	Modelo Mamdani da dimensão Atualidade	80
Figura 5.17	Função de Pertinência da Variável #DR-DL	80
Figura 5.18	Função de Pertinência da Variável #DC/DA	81
Figura 5.19	Função de Pertinência da Variável de saída Atualidade	81
Figura 5.20	Regras de produção da dimensão Atualidade	81
Figura 5.21	Modelo Mamdani da dimensão Completude	82
Figura 5.22	Função de Pertinência da Variável #DR-DL	82
Figura 5.23	Função de Pertinência da Variável #DFF-DR	83

LISTA DE FIGURAS

Figura 5.24	Função de Pertinência da Variável de saída Completude	83
Figura 5.25	Regras de produção da dimensão Completude	83
Figura 5.26	Modelo Mamdani da dimensão Consistência	84
Figura 5.27	Regra de Produção da Variável #VO-VM	84
Figura 5.28	Função de Pertinência da Variável de saída Consistência	85
Figura 5.29	Regras de produção da dimensão Consistência	85
Figura 5.30	Modelo Mamdani da Qualidade	85
Figura 5.31	Função de Pertinência da Variável Acurácia	86
Figura 5.32	Função de Pertinência da Variável Atualidade	86
Figura 5.33	Função de Pertinência da Variável Completude	87
Figura 5.34	Função de Pertinência da Variável Consistência	87
Figura 5.35	Função de Pertinência da Variável de saída Qualidade	87
Figura 5.36	Regras de produção para cálculo da Qualidade	88
Figura 5.37	Classificação linguística	88
Figura 5.38	DW-Q: Modelo do DW para a qualidade dos dados	91
Figura 6.1	Gráfico de qualidade de membro de equipe no mês Março na equipe Server	104
Figura 6.2	Gráfico de qualidade de membro de equipe no mês Março na equipe Server por dimensões	104
Figura 6.3	Gráfico de qualidade do mês de Fevereiro por equipe e dimensão	105
Figura 6.4	Gráfico de média de qualidade por equipe no mês de Fevereiro	106
Figura 6.5	Gráfico de qualidade geral no ano de 2007	106
Figura 6.6	Gráfico de qualidade de equipe no ano de 2007 por dimensões	107

Sumário

1: Introdução	14
2: Fundamentação teórica	17
2.1 Processo de Desenvolvimento de Software	17
2.1.1 Modelos de processos	18
2.2 Mensuração de PDS	20
2.2.1 CMMI	20
2.3 Mensuração e Análise	23
2.3.1 Métricas	24
2.3.2 Validação de métricas	27
2.3.3 Metodologias de validação de métricas de software	28
2.3.3.1 Padrão IEEE 1061	28
2.3.3.2 Outras propostas	28
2.3.3.3 Observações sobre a seção	30
2.4 Qualidade de dados	30
2.4.1 Qualidade de dados em métricas	32
2.4.2 Confiabilidade e validade de dados	33
2.5 Avaliação de qualidade de dados em métricas de esforço	35
2.5.1 Lógica difusa (fuzzy)	36
2.5.1.1 Regras de produção fuzzy	37
2.5.1.2 Variáveis lingüísticas e termos	38
2.5.1.3 Funções de pertinência	38
2.5.1.4 Operações fuzzy	39
2.5.1.5 Métodos de <i>Defuzzification</i> (defuzzificação)	40
2.5.2 Data Provenance	41
2.6 Observações sobre o capítulo	46
3: Cenário	47
3.1 Descrição	47
3.2 O Problema	49
3.2.1 Métrica para avaliação estatística	49
3.2.2 Método de análise	49

SUMÁRIO

3.2.3	Forma de comparação	51
3.2.4	Resultados	52
3.3	Considerações sobre o estudo realizado	55
3.4	Requisitos da solução	55
4:	Trabalhos relacionados	58
4.1	Análise	58
4.1.1	Primeiro grupo	59
4.1.1.1	Considerações sobre trabalhos do primeiro grupo	60
4.1.2	Segundo grupo	62
4.1.2.1	Considerações sobre trabalhos do segundo grupo	63
4.2	Considerações sobre o Capítulo	63
5:	Solução	64
5.1	Arquitetura proposta	64
5.2	Processo de avaliação	65
5.3	Componentes da arquitetura	66
5.3.1	Componente Proveniência	66
5.3.1.1	Construção do componente através de dimensões de qualidade	67
5.3.1.2	Aquisição de conhecimento	69
5.3.1.3	Sobre o processo empregado nas entrevistas	71
5.3.2	Componente de Inferência	72
5.3.2.1	Construção do componente através de entrevista	74
5.3.2.2	Implementação da máquina de inferência	77
5.3.3	Componente de Qualidade	88
5.3.4	Componente de Análise	90
5.4	Considerações finais sobre o Capítulo	91
6:	Testes da solução proposta	93
6.1	Objetivos e procedimento	93
6.2	Execução	95
6.3	Análise dos resultados	102
6.4	Outras análises	103
6.4.1	Estado da qualidade dos dados por mês no ano de 2007	103
6.4.1.1	Estado da qualidade dos dados por equipe ao longo do ano de 2007	106
6.5	Considerações sobre os testes	107

SUMÁRIO

7: Conclusão	109
7.1 Considerações finais	109
7.2 Trabalhos futuros	111
REFERÊNCIAS	112
Apêndice A: Resumo da Norma 1061 de 1998	116
Apêndice B: Entrevista semi-estruturada	122
Apêndice C: Modelos preditivos	133

1

Introdução

Segundo [Kan, 2003] a medição é crucial para o progresso da ciência e da engenharia, pois sem a verificação empírica de dados de medições, teorias e proposições continuariam abstratas. A medição de PDS (Processo de Desenvolvimento de Software) é prática comum nas organizações para, através de comparações entre os valores das métricas e as metas da organização, tirar conclusões sobre a qualidade do processo ou produto de software [Sommerville, 2004]. No entanto, muitas vezes não é feito um esforço para garantir a qualidade desses valores e conseqüentemente estes deixam de ser confiáveis.

Problemas com qualidade de dados é uma realidade nas grandes organizações, pois a preocupação é geralmente com a quantidade de dados armazenados e não com a qualidade dos mesmos. As conseqüências da baixa qualidade são refletidas nos custos da organização, visto que as decisões tomadas são baseadas nesses dados.

No contexto de operações de software, pode ser armazenada uma grande quantidade de dados referentes às medições no processo de desenvolvimento. Modelos de maturidade, como CMMI [Institute, 2006], indicam a coleta de métricas como meio para controlar e analisar processos, para sua contínua melhoria. Dentre as métricas geralmente coletadas em PDS (por exemplo: esforço, defeitos, tamanho, variação de cronograma), a métrica de esforço exerce um dos maiores impactos na gestão dos projetos, pois mede o tempo que a equipe precisou para desenvolver, revisar e/ou consertar um produto [Society, 1998]. Por isso, a avaliação de dados referentes a essa métrica merece especial tratamento.

A qualidade dos dados de métricas de esforço é especialmente complexa de ser avaliada porque os dados não são obtidos por instrumentos automatizados que possibilitem o apontamento direto de falhas na medição. Os dados de esforço são dados lançados manualmente pelos membros da equipe, normalmente em horas, e não obrigatoriamente logo ao término de cada atividade. Os aspectos subjetivos e manuais de coleta tornam a avaliação

da qualidade dos dados coletados uma tarefa complicada. Tal tarefa exige a consideração de diversos fatores inerentes ao PDS para um julgamento menos propenso a falhas. Outro aspecto que torna a avaliação de qualidade de dados de esforço uma tarefa não tão simples é a existência de uma variável que influencia diretamente na avaliação: o real conceito de “bom” ou “ruim”. Os dados estão inseridos em diferentes contextos, formatos, e sob diferentes propósitos de coleta, cujos aspectos adicionam relevante complexidade na definição do que pode ser considerado um dado bom ou um dado ruim [Redman, 2001].

São encontrados na literatura alguns trabalhos que visam a avaliação de qualidade de dados em diferentes contextos. Dentre eles se destacam: AIQM (*A methodology for information quality assessment*) [Lee et al., 2002], ASM-IQS (*Assessment of software measurement an information quality study*) [Berry et al., 2004], DQA (*Data quality assessment*) [Francalanci and Pernici, 2004], DQMIM (*A data quality measurement information model*) [Caballero et al., 2007], ORME-DQ (*A framework and a methodology for data quality assessment and monitoring*) [Batini et al., 2007], MDB:PA (*Measuring data believability: a provenance approach*) [Nicolas Prat, 2008] e PEDI (*Process-embedded data integrity*) [Lee et al., 2004]. No entanto, nenhum deles contempla de maneira adequada a avaliação da qualidade desses dados, obtidos especificamente de métricas de esforço de software, considerando as características inerentes a um PDS e atendendo às necessidades de gestores desse meio como, por exemplo, rapidez na avaliação e apontamento das potenciais causas dos problemas eventualmente encontrados.

Esta pesquisa visa oferecer suporte à essa avaliação de dados referentes a métricas de esforço de software atendendo às reais necessidades do processo de tomada de decisão desse contexto. Além disso, focaliza em oferecer funcionalidade adequada e objetividade, e apontando causas de problemas através do uso de dimensões de qualidade e através da apresentação de níveis de qualidade (ótima, boa, razoável, muito ruim).

Este trabalho está organizado da seguinte forma. O Capítulo 2 apresenta o referencial teórico necessário para o entendimento dos conceitos e técnicas adotados por esta pesquisa. O Capítulo 3 descreve o cenário onde a pesquisa está inserida. O Capítulo 4 relata um estudo sobre os trabalhos relacionados à presente pesquisa. O Capítulo 5 descreve a solução proposta, através da descrição de uma arquitetura de avaliação e a implementação da mesma. O Capítulo 6 apresenta os testes realizados para verificar se a avaliação proposta consegue tratar adequadamente os aspectos que envolvem a avaliação de dados referentes a medições de esforço de software juntamente com análises adicionais que mostram como a avaliação pode ser utilizada por gestores neste contexto. O Capítulo 7 apresenta uma as

considerações finais e os trabalhos futuros.

2

Fundamentação teórica

Este capítulo tem o objetivo de oferecer um melhor entendimento sobre alguns conceitos envolvidos na avaliação de qualidade de dados no contexto de métricas coletadas durante o desenvolvimento de software. Tendo em vista que o presente trabalho está focalizado em dados oriundos de medições de PDS, é apresentada uma breve descrição sobre:

- (i) PDS juntamente com modelos de processos;
- (ii) mensuração de PDS envolvendo a justificativa de utilização de métricas em modelos de maturidade como o CMMI e aprofundando conceitos de medições e metodologias utilizadas na validação dessas medições,
- (iii) conceitos de qualidade de dados abordando questões de confiabilidade e validade e
- (iv) conceitos que podem estar envolvidos em avaliação de qualidade de dados como lógica fuzzy e *data provenance*.

2.1 Processo de Desenvolvimento de Software

De maneira genérica, Pfleeger em [Pfleeger, 2004] entende um processo como um conjunto de tarefas ordenadas, ou seja, uma série de etapas que envolvem atividades, restrições e recursos para alcançar a saída desejada. Mais especificamente, quando um processo envolve a elaboração de um produto de software, este processo (PDS) pode ser entendido como ciclo de vida do mesmo. Isso porque este tipo de processo descreve a vida do produto começando da concepção, passando pela implementação, entrega, utilização e manutenção.

Sommerville em [Sommerville, 2004] define que um PDS é um conjunto de atividades que resultam em um produto de software. Essas atividades podem envolver o desenvolvimento de software desde o início ou partir da extensão e modificação de sistemas existentes,

configurando e integrando componentes específicos ao produto desejado. O autor afirma que não há um processo ideal e muitas organizações desenvolvem seus próprios processos, buscando assim atender as suas necessidades específicas. Existem diversos PDSs. Entretanto algumas atividades são fundamentais e comuns a todos os processos, tais como: (a) Especificação, onde devem ser definidas as funcionalidades do software; (b) Projeto e Implementação, onde o software é produzido de acordo com as suas especificações; (c) Validação, o software é validado para assegurar que este faz o que o cliente deseja e (d) Evolução, onde o software deve evoluir conforme as mudanças que o cliente necessita.

Rocha em [Ana R. Rocha, 2001] diz que um software consiste em conjuntos de informações em diferentes níveis de abstração, transformações e decisões relacionadas a essas transformações. Também enfatiza que vários aspectos são fundamentais no desenvolvimento de software, mas para que um processo seja de qualidade, um requisito básico é que a atividade seja sistemática e passível de repetição, independente de quem a execute, bem como há a necessidade da qualidade do produto de software ser independente de quem o produziu. O desenvolvimento deste trabalho é baseado nos conceitos de PDS de Sommerville [Sommerville, 2004] e também na visão de qualidade de PDS de Rocha [Ana R. Rocha, 2001].

2.1.1 Modelos de processos

Sommerville em [Sommerville, 2004] define modelo de processo como uma representação abstrata do processo de software. Cada modelo representa um processo sob uma visão particular e assim fornece informações parciais sobre este.

Pfleeger em [Pfleeger, 2004] classifica modelos em dois tipos específicos e diferentes entre si: (1) modelos que indicam um caminho que o desenvolvimento de um software deveria seguir e (2) modelos que são descrições de como realmente é feito um software. O autor entende que esses dois tipos de modelos deveriam ser semelhantes ou iguais; porém não é o que acontece na prática. Modelar um processo resulta em importantes benefícios, pois o registro da descrição do desenvolvimento de software possibilita o entendimento sobre os recursos, atividades e restrições envolvidas no projeto. A presença de um modelo auxilia na detecção de inconsistências, redundâncias e futuros problemas na execução das atividades do processo pois, conforme o modelo vai sendo desenvolvido, é possível analisar as atividades com mais detalhes e com enfoque maior nos seus objetivos. Dentre os modelos existentes, alguns são conhecidos como genéricos. Esses são abordados a seguir de acordo

com [Sommerville, 2004].

- **Modelo Cascata:** O modelo Cascata foi o primeiro modelo proposto derivado de processos de engenharia de sistemas. As atividades comuns a todos os processos, citadas anteriormente, são organizadas em fases: (1) Definição e Análise de Requisitos, (2) Projeto, (3) Implementação e Teste Unitário, (4) Teste de integração e sistema e (5) Manutenção. O resultado de cada fase é um ou mais documentos que são aprovados. Uma fase não deve começar antes que a anterior tenha terminado.
- **Modelo Evolutivo Iterativo:** O modelo evolutivo é baseado na idéia de desenvolver uma implementação inicial, expô-la para o usuário comentar, para que assim o sistema seja refinado em várias versões até que o sistema mais apropriado seja desenvolvido. O ciclo é baseado na execução de atividades de Especificação, Desenvolvimento e Validação, as quais são intercaladas e não ocorrem separadamente.
- **Modelo Espiral:** O modelo espiral ao invés de representar o processo de desenvolvimento de software como uma seqüência de atividades com iterações entre atividades, o processo é representado como um espiral. Cada volta no espiral representa uma fase do processo. Cada volta desse modelo é composta por quatro etapas: (1) Determinar objetivos, (2) Avaliar alternativas, identificar e resolver riscos, (3) Desenvolver, verificar produto do próximo nível e (4) Planejar a próxima fase. Cada etapa é representada por um quadrante de um diagrama cartesiano. O raio do espiral representa o custo acumulado do projeto.
- **Processo Unificado:** O Processo Unificado (UP) é um modelo de processo de desenvolvimento de software que define quem, o que, quando e como é desenvolvido o software. O RUP (*Rational Unified Process*) é baseado na linguagem de modelagem UML (*Unified Modeling Language*), com algumas diferenças. O RUP combina a forma disciplinada do modelo Cascata e a responsabilidade dinâmica do modelo iterativo [Jim Arlow, 2005]. Por esse motivo, Sommerville [Sommerville, 2004] afirma que RUP é um bom exemplo de modelo de processo híbrido, já que este une elementos de todos os modelos genéricos.

2.2 Mensuração de PDS

Em tempos de alta competitividade, a qualidade é fator essencial como vantagem competitiva e por isso as organizações buscam aprimorar seus processos para que estes resultem em produtos de qualidade, já que a qualidade do produto está diretamente relacionada à qualidade do processo pelo qual este é desenvolvido [Khoshgoftaar et al., 2001]. Este comportamento depende da evolução e melhoria contínua do PDS, alcançadas através do uso de modelos de maturidade que guiam as organizações na obtenção do controle de seus processos e na evolução para uma cultura de excelência em gestão [Institute, 2006]. Os autores tratam esta iniciativa de melhoria contínua como modelos de SPI (*Software Process Improvement*) que objetiva a evolução do PDS.

As organizações buscavam a certificação em algum padrão logo no início da abordagem de qualidade de software. Inicialmente o padrão mais utilizado foi ISO9000. Porém este não tinha objetivos específicos para a área de software, surgindo então o modelo CMM que era voltado às organizações de software. O CMM foi desenvolvido pelo SEI (*Software Engineering Institute*) ligado à Universidade *Carnegie Mellon*. O objetivo foi estabelecer um padrão de qualidade para o software desenvolvido para as forças armadas americanas, pois estes eram em sua maioria ineficientes e entregues em prazos totalmente inesperados.

Esses modelos de maturidade indicam a coleta de métricas como meio para controlar e analisar os processos e obter essa melhoria contínua acima mencionada. Atualmente um dos modelos de maturidade mais difundidos é o CMMI, estudado com maior detalhe a seguir.

2.2.1 CMMI

O CMM foi substituído pelo modelo CMMI onde a aplicabilidade não é restrita a processos de desenvolvimento de software. O CMMI possui dois modos de representação: (1) em estágios (*staged*) e (2) contínuo (*continuous*) e para ambos, o elemento-chave é a PA (Área de Processo). A representação em estágios é a mesma utilizada no SW-CMM que é o modelo do CMM voltado para software. Esta representação considera os processos da organização como um todo e provê um caminho de melhorias com grupos de PAs, estabelecendo fácil migração do SW-CMM para o CMMI. A Figura 2.1 ilustra os componentes do CMMI na representação em estágios, onde as PAs são organizadas em níveis de maturidade. Dentro das PAs estão definidos os objetivos genéricos e específicos, assim como as atividades ou

práticas genéricas e específicas. As características em comum (*Common Features*) definem as práticas genéricas [Institute, 2002].

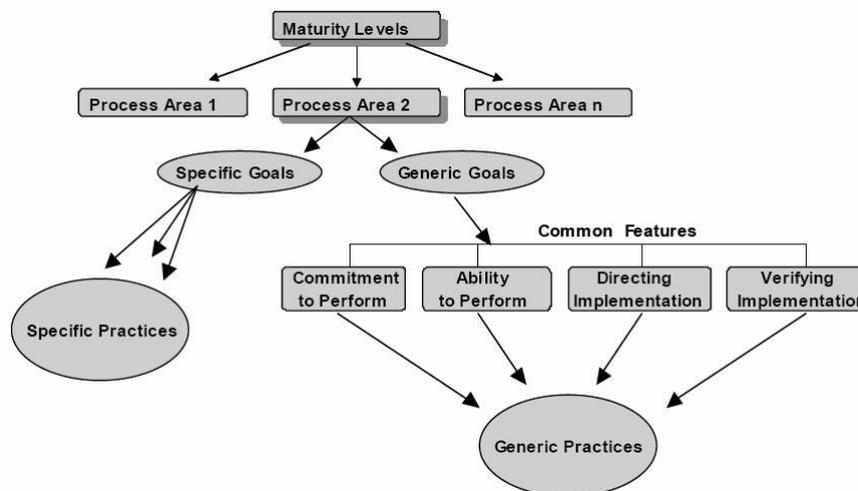


Figura 2.1 – CMMI na representação em estágios. Fonte: [Institute, 2002]

De acordo com [Institute, 2002] na representação em estágios, o CMMI considera 5 níveis de maturidade. Estes níveis indicam a maturidade da organização como um todo, sendo que cada nível compreende um conjunto de PAs a serem satisfeitas. Nesta representação, para a organização atingir determinado nível é preciso que atinja os níveis em seqüência. Por exemplo, não é possível alcançar o nível 4 sem antes alcançar os níveis 2 e 3. A Tabela 2.1 mostra os níveis de maturidade da representação em estágios do CMMI.

Tabela 2.1 – Níveis de maturidade CMMI- representação em estágios. Fonte:[Institute, 2002]

Nível	Maturidade
1	Inicial
2	Gerenciado
3	Definido
4	Gerenciado Quantitativamente
5	Otimizado

O nível Inicial 1 indica um estado caótico, onde a organização tem um controle informal dos processos podendo estes resultarem ou não, produtos de qualidade. O CMMI considera que toda a organização está, por definição, no nível 1. O nível 2 é o gerenciado, focado nos

projetos, sugerindo o gerenciamento dos processos de cada projeto, podendo estes serem diferentes entre si. O nível 3, Definido, é focado na organização, ou seja, na padronização dos processos da organização através da análise dos processos dos projetos onde os mais bem sucedidos são institucionalizados. O nível 4, Gerenciado Quantitativamente, sugere o gerenciamento quantitativo através da coleta e análise de métricas dos processos, dentre outras práticas, sendo esta a mais importante. E por fim, o nível 5 é o Otimizado, onde prevê a melhoria contínua dos processos da organização [Kanungo, 2004].

A representação contínua utiliza níveis de competência para indicar a maturidade do processo, visa a melhoria individual de áreas de processos, escolhidas de acordo com as necessidades específicas da organização [Kanungo, 2004]. A Figura 2.2 ilustra os componentes do CMMI na representação contínua, onde para cada PA são definidos objetivos específicos, os quais possuem práticas específicas. Os níveis de competência são caracterizados por PA.

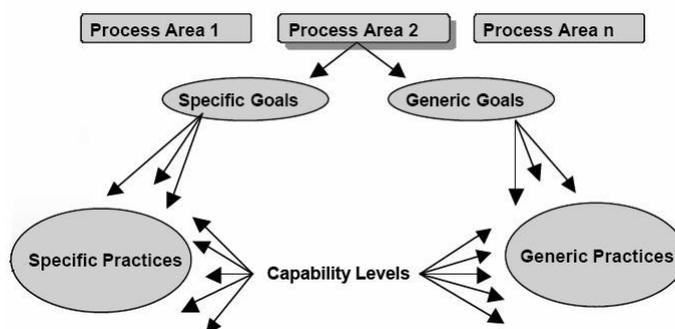


Figura 2.2 – CMMI na representação contínua. Fonte: [Institute, 2002]

Na representação contínua existem 6 níveis de competência numerados de 0 à 5, como pode ser visualizado na Tabela 2.2. Cada nível de competência corresponde a um objetivo genérico e a um conjunto de práticas genéricas e específicas.

Tabela 2.2 – Níveis de competência CMMI – representação contínua. Fonte: [Institute, 2002]

Nível	Capacidade
0	Incompleto
1	Executado
2	Gerenciado
3	Definido
4	Gerenciado Quantitativamente
5	Otimizado

Cada PA é um grupo de atividades relacionadas que buscam atender um grupo de objetivos. Neste contexto, processos dizem “o que fazer” ao invés de “como fazer”. A PA especifica objetivos e atividades que, ao serem executadas, espera-se que atinjam estes objetivos. Alguns objetivos e práticas são específicos à PA, outros são genéricos e aplicados em todas as PAs. Estes objetivos genéricos descrevem maneiras essenciais para que os processos possam ter um bom grau de repetição, padronização e controle adequado [Kanungo, 2004]. Independentemente do modo de representação, o modelo CMMI indica a coleta de métricas na PA de MA (Mensuração e Análise) .

2.3 Mensuração e Análise

O papel desta PA é definir a coleta e análise de dados e relatar a informação aos respectivos projetos, proporcionando um retorno aos interessados. A MA abrange 3 objetivos, sendo que destes, 2 são específicos e 1 é genérico.

Alinhar atividades de MA é o primeiro objetivo específico, com as seguintes práticas específicas: (a) estabelecer objetivos de mensuração, (b) especificar as métricas, (c) especificar procedimento de coleta e armazenamento de dados e (d) especificar procedimentos de análise. Em outras palavras, preocupar-se em por quê/o quê/como medir e o que fazer com os resultados obtidos.

O segundo objetivo específico é prover resultados de mensuração. As práticas específicas são: (a) coletar dados da mensuração, (b) analisar dados da mensuração, (c) armazenar dados e resultados e (d) comunicar os resultados. A preocupação está em seguir o planejamento e enfatizar a necessidade de comunicar os resultados aos interessados.

O terceiro objetivo, que é o genérico, é institucionalizar processo gerenciado com as seguintes práticas genéricas: (a) estabelecer uma política organizacional, (b) planejar o

processo, (c) prover recursos e (d) atribuir responsabilidades. Este objetivo busca, portanto, a institucionalização da MA, diferentemente do que acontecia no CMM, onde MA era um componente para institucionalizar os processos: no CMMI, MA é um processo a ser institucionalizado. Esta institucionalização envolve treinamento de pessoas, gerenciamento de configurações, identificação e envolvimento de *stakeholders* e monitoração e controle de processos [Zubrow, 2001].

Esta PA recomenda que os dados específicos de cada projeto sejam armazenados em um repositório. O CMMI define que quando esses dados são largamente compartilhados entre os projetos, então esses dados devem ficar em um repositório geral da organização. Esse repositório é o resultante da integração de um plano de métricas organizacional e do OSSP (*Organizational Standard Set of Processes*) que é o conjunto de processos que toda a organização adota e considera como padrão para suas atividades. O repositório contém medidas de processo e produto relatadas no OSSP, onde estão as informações dos processos que guiam as atividades da organização [Palza et al., 2003].

2.3.1 Métricas

A melhoria do processo e do produto de software visa reduzir a probabilidade de erro, aumentar a produtividade e facilitar manutenção, entre outros objetivos [Ana R. Rocha, 2001]. São vários os fatores utilizados para verificar a qualidade de um software, como por exemplo a manutenibilidade, confiabilidade, flexibilidade, entre outros. No entanto estes fatores são de caráter qualitativo e para se melhorar um processo é preciso medi-lo quantitativamente para eliminar a subjetividade desta atividade. O uso de métricas de software reduz esta subjetividade para o alcance à qualidade do software, provendo uma base quantitativa para a tomada de decisão sobre o desenvolvimento do software. Não obstante, o uso de métricas de software não elimina a necessidade de julgamento humano, ou seja, não elimina a necessidade da interpretação e criatividade das pessoas na avaliação do produto e processo [Society, 1998].

Segundo Kan em [Kan, 2003], as métricas de software podem ser classificadas em três categorias: produto, processo e projeto. As de produto descrevem as características do produto (e.g. tamanho, complexidade, peculiaridades do projeto, desempenho e nível de qualidade). As de processo podem ser utilizadas para melhorar o desenvolvimento e a manutenção do PDS (e.g. defeitos encontrados e o esforço despendido para a sua correção, eficiência de remoção de defeitos). As de projeto representam as características do projeto

e da sua execução (e.g. número de recursos, custo, cronograma e produtividade).

O padrão da IEEE 1061 [Society, 1998] classifica as métricas de software em medidas diretas e medidas indiretas, e define que uma métrica direta é uma métrica que não depende da medida de nenhum outro atributo, nem necessita de validação. Por sua vez, uma métrica indireta depende de outros atributos e necessita ser validada. O padrão IEEE 1061 definiu os termos aplicados em medição de software a fim de padronizá-los, e alguns destes. Os mais utilizados neste trabalho, estão listados a seguir [Society, 1998]:

- Atributo: uma propriedade física ou abstrata mensurável de uma entidade;
- Fator de qualidade: um tipo de atributo, ou seja, um atributo orientado a gestão de software que contribui para sua qualidade;
- Métrica: é uma função mensurável;
- Métrica direta: uma métrica que não depende da medida de outro atributo;
- Métrica de qualidade: uma função cujas entradas sejam dados do software e cuja saída seja um único valor numérico que possa ser interpretado como o grau que o software possui;
- Métrica validada: uma métrica cujos valores estão sendo estatisticamente associados com o fator de qualidade correspondente;
- Subfator de qualidade: uma decomposição do fator de qualidade ou um subfator para seus componentes técnicos;
- Validação de métrica: o ato ou processo de assegurar que uma métrica é confiável e realmente prediz ou avalia o fator de qualidade.

O padrão IEEE 1061 diz que uma métrica direta é pressupostamente válida. Diferentemente, [Kan, 2003] aponta que todas as métricas devem ser validadas. Exemplos de métricas diretas são: (i) número de defeitos, (ii) tamanho e (iii) número de linhas de código. Exemplos de métricas derivadas são:

- Produtividade = número de linhas de código/tempo

- Densidade de defeitos = números de defeitos/tamanho
- Estabilidade dos requisitos = número de requisitos inicial/número total de requisitos

Neste trabalho, são adotados os termos de métrica e medida como sinônimos, como descritos acima. Conforme KAN, alguns tipos de medidas são usadas com maior frequência em desenvolvimento de software, são elas:

Relação

O resultado de uma relação é a divisão de uma quantidade por outra. O numerador e o denominador são de populações distintas e são mutuamente exclusivos - se um membro pertence à população referente ao numerador, então esse membro não pertencerá à população referente ao denominador.

Exemplo de métrica:

$$DensidadeDeFeitosEntregues = \frac{NumeroDeFeitosEncontradosNoCliente}{TamanhoAtualProduto}$$

O numerador “NumeroDeFeitosEncontradosNoCliente” pertence à população “Defeitos” e o denominador “TamanhoAtualProduto” pertence à população “Tamanho”.

Proporção

A proporção é diferente da relação porque o numerador é parte do denominador. Exemplo de uma métrica baseada nesse tipo de medida é métrica derivada Eficiência de Remoção de Defeitos:

$$EficienciaDeRemocao = \frac{NumeroDeFeitosInternos}{NumeroDeFeitosInternos + NumeroDeFeitosExternos}$$

Note que o numerador “NumeroDeFeitosInternos” faz parte do denominador que é a soma da quantidade de defeitos internos com a quantidade de defeitos externos.

Porcentagem

Uma proporção ou fração se transforma em porcentagem quando é expressa em termos de “por 100 unidades” (o denominador é normalizado para 100).

Alguns autores recomendam que sejam medidas no mínimo 50 casos para se obter resultados sem distorções usando porcentagem. Dependendo do número de categorias, Kan [Kan, 2003] recomenda no mínimo 30 casos, que, segundo este mesmo autor, é o menor tamanho de população para análises estatísticas.

2.3.2 Validação de métricas

Um dos fatores mais importantes e críticos na utilização de métricas é a maneira pela qual os resultados são avaliados, o que depende substancialmente da validação das métricas. O padrão IEEE (1998) define que, para uma métrica ser válida, esta deve ter um alto grau de associação com os fatores de qualidade que ela representa. Na documentação do padrão IEEE (1998) são descritos, detalhadamente, critérios para verificar quando uma métrica é válida ou não.

Em dezembro de 1992 foi aprovado o padrão IEEE 1061, *IEEE Standard for a Software Quality Metrics Methodology*, que é uma metodologia para identificar, implementar, analisar e validar métricas de qualidade de software, processos e produtos. É importante salientar que o padrão é um processo e não especifica quais métricas utilizar. Dentro de 5 anos um padrão deve ser revisado ou reafirmado, este trabalho é baseado na versão do padrão aceito em 1998 e reafirmado em 2004 [Society, 1998].

O padrão define que qualidade de software é um degrau através do qual o software possui uma combinação de atributos. Definir a qualidade de um sistema é equivalente a definir uma lista de atributos de qualidade “exigidos” pelo sistema. Para medir estes atributos um conjunto apropriado de métricas de software deve ser estabelecido. Entre outros aspectos, o uso do padrão IEEE 1061 auxilia uma organização a:

- (i) estabelecer requisitos de qualidade para um sistema e suas saídas,
- (ii) alcançar os objetivos de qualidade,
- (iii) avaliar o nível de qualidade alcançada de acordo com os requisitos estabelecidos,
- (iv) detectar anomalias ou potenciais problemas no sistema e
- (v) validar um conjunto de métricas. Para isso, um conjunto de métricas deve estar representado em um plano de métricas.

2.3.3 Metodologias de validação de métricas de software

2.3.3.1 Padrão IEEE 1061

A norma define uma metodologia de validação de métricas através da definição de limites, critérios e um procedimento para adotá-los. O apêndice A contém uma sumarização do padrão IEEE 1061 que pode ser acessado para maiores informações e detalhamento sobre suas definições para limites, critérios e aplicação do seu próprio procedimento de validação.

De acordo com o padrão [Society, 1998], se uma métrica não passa em todos os critérios de validação, ela deve ser usada somente de acordo com o critério que ela passou. Por exemplo, se uma métrica é considerada válida quanto o critério Rastreabilidade, ela deve ser usada somente para acompanhar a evolução (ou involução) do processo ou produto em questão.

O padrão também diz que é importante ressaltar que a validação das métricas deve ser aplicado em ambientes estáveis de desenvolvimento, ou seja, onde não muda com frequência a linguagem de implementação, as ferramentas de desenvolvimento e outras especificidades do projeto. Além disso, a aplicação e validação de métricas deve ser implementada durante as mesmas fases do ciclo de vida em diferentes projetos. Por exemplo, se uma métrica A é coletada durante a fase de design do projeto P e tem relação com o fator de qualidade F, então uma outra métrica B seria coletada durante a fase de design do projeto Q e teria relação com o fator de qualidade F.

2.3.3.2 Outras propostas

Alguns pesquisadores também estudam questões relativas à validação de métricas. Kaner em [Kaner and Bond, 2004] sugere um *framework* para avaliar métricas. Este framework é composto por 10 perguntas. O autor entende que respondendo essas perguntas, vários aspectos quanto à definição das métricas serão esclarecidos, garantindo assim sua validade. As perguntas a serem respondidas para cada métrica do plano de métricas de uma organização são as seguintes:

1. Qual a proposta da métrica?
2. Qual o escopo da métrica?
3. Que atributo estamos tentando medir?

4. Qual a escala natural do atributo que estamos tentando medir?
5. Qual a variabilidade natural do atributo?
6. O que é a métrica? (a função atribuída ao valor do atributo) Que instrumento de medição usamos para fazer a medição? (Exemplos de instrumentos: contagem, comparação, etc)
7. Qual a escala natural da métrica?
8. Qual a variabilidade natural de leitura do instrumento? (Erro de medição)
9. Qual a relação do atributo com o valor da métrica? (Questão da validade)
10. Quais os efeitos naturais de usar esse instrumento?

No estudo de [Kaner and Bond, 2004] esse *framework* foi aplicado com exemplos de métricas de defeitos.

Florac em [Florac et al., 1997] também sugere uma avaliação de dados coletados de métricas. Para essa avaliação o autor indica que os critérios estabelecidos por ele devem ser satisfeitos:

- Critério 1: Dados devem ser verificados
- Critério 2: Dados devem estar sincronizados
- Critério 3: Dados devem ser consistentes
- Critério 4: Dados devem ser válidos

O critério 1 indica que os dados devem ser examinados para se assegurar que foram coletados de acordo com as especificações e não contêm erros. Essa verificação avalia se os dados: (a) são do tipo correto, (b) estão no formato correto, (c) estão dentro das escalas especificadas - valores discrepantes devem ser investigados e corrigidos antes de serem carregados, (d) são aritmeticamente corretos e (e) são internamente e externamente consistentes.

O critério 2 diz respeito à medições de atributos de um processo, produto e recursos para descrever o desempenho do processo. Um exemplo de situação frequentemente encontrada é quando relatos de esforço são preparados mensalmente enquanto que medidas de tamanho são relatadas em intervalos de semanas, sendo que as duas medições são utilizadas em uma métrica.

O critério 3, de consistência, tem relação com o tempo, cuidados com definições que podem mudar de projeto para projeto, reclassificações de categorias e atividades durante o processo de medição.

O critério 4, de validade, verifica se os valores usados para descrever um atributo verdadeiramente descrevem o atributo de interesse. Para isso, a definição das medições devem ser muito bem feitas. Sem definições adequadas e consistentes para todos os momentos do processo de medição, nenhum dado pode ser considerado para representar um assunto de interesse.

2.3.3.3 Observações sobre a seção

Todas essas propostas descritas anteriormente estão preocupadas em garantir que as métricas possuam qualidade quanto ao seu propósito de uso, ou seja, se a métrica de defeito realmente contabiliza os problemas encontrados no produto e não apenas melhorias solicitadas. Dentre as propostas apresentadas nesta seção a Norma 1061 se mostra como a mais completa e detalhada através da definição de limites e critérios. Enquanto que as outras não apresentam todo esse detalhamento e até mesmo maturidade em sua utilização. Na próxima seção são tratadas questões referentes à qualidade de dados que representam as métricas. Diferentemente das propostas anteriores, a preocupação está em afirmar o quão verdadeiro é o valor associado à métrica, mesmo que já validada.

2.4 Qualidade de dados

A qualidade de dados é uma questão crítica nas organizações devido à grande quantidade de dados que elas armazenam. Cada vez mais, a qualidade dos dados tem recebido atenção especial dos grupos de garantia de qualidade devido aos benefícios de alta confiabilidade que se pode alcançar. Dados de alta qualidade podem melhorar a satisfação dos clientes se tratados estrategicamente como vantagens competitivas no mercado[Lee et al., 2006].

De forma generalizada, Strong et. al.[Strong et al., 1997] entendem dados de alta qualidade como dados acessíveis e “amigáveis” para os seus usuários, sendo este um conceito amplamente adotado, inclusive por este trabalho. Neste sentido, o autor classificou algumas características de dados de alta qualidade em quatro categorias baseadas em características: (1) *Intrinsic* DQ - Categoria de aspectos intrínsecos, (2) *Accessibility* DQ - Categoria de aspectos de acessibilidade, (3) *Contextual* DQ - Categoria de aspectos de contexto e

(4) *Representational DQ* - Categoria de aspectos de representação. Cada categoria possui dimensões relacionadas ao tipo de aspecto tratado, como mostra a Tabela 2.3.

Tabela 2.3 – *Categorias e dimensões de DQ. Fonte:[Strong et al., 1997]*

DQ Category	DQ Dimensions
Intrinsic DQ	Accuracy, Objectivity, Believability, Reputation
Accessibility DQ	Accessibility, Access security
Contextual DQ	Relevancy, Value-Added, Timeliness, Completeness, Amount of data
Representational DQ	Interpretability, Ease of understanding, Concise representation, Consistent representation

Strong em [Strong et al., 1997] sugere que um problema de DQ é qualquer dificuldade encontrada em uma ou mais dimensões que endereçam a completude ou a falta de usabilidade do dado.

Em [Hulse et al., 2007] pode ser encontrada uma melhor definição do que é tratado em cada uma das categorias da Tabela 2.3, tais como:

- Categoria “*Intrinsic DQ*”: onde são discutidas os tipos mais comuns de problemas, tratando-se de acurácia e objetividade do dado fonte;
- Categoria “*Accessibility DQ*”: onde são discutidas as questões referentes à habilidade de se obter dados necessários em um tempo adequado de maneira eficiente;
- Categoria “*Contextual DQ*”: onde são discutidas questões de dados faltantes, incompletos ou definições inadequada dos dados;
- Categoria “*Representational DQ*”: onde são discutidas questões de interpretabilidade, facilidade de entendimento e representação concisa e consistente de dados.

O uso de dimensões para a observação da qualidade dos dados é um consenso entre a comunidade científica. O que não é objetivo e rígido é a definição de quais dimensões devem ser consideradas, ficando esta decisão por conta dos analistas dos dados ligados ao seu contexto de uso.

Inicialmente os usuários dos dados, geralmente gestores, não sabiam a qual fonte atribuir os problemas de qualidade de dados, visto que agrupamentos inadequados de dados de fontes distintas eram, e ainda são, causas comuns de problemas tipo intrínsecos. Os usuários apenas sabiam que o dado era conflitante e que a produção dele podia ter um

caráter subjetivo, evidenciando assim os problemas de *confiabilidade*. Com o tempo, a questão evoluiu para a avaliação da *acurácia* das diferentes fontes, identificando assim uma baixa reputação das mesmas. Essas fontes com baixa reputação começaram a ser vistas como fontes que agregavam pouco valor à organização, resultando então em uma redução significativa do uso dos dados. A Figura 2.3 ilustra esse raciocínio.

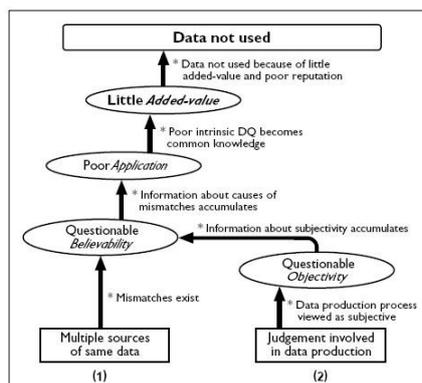


Figura 2.3 – Problemas de "Intrinsic DQ" Fonte:[Strong et al., 1997]

Como mostra a Figura 2.3, julgamento no processo de produção dos dados envolvendo a subjetividade é outra causa comum de problemas desta categoria. Por exemplo, um dado interpretado é considerado de menor qualidade que um dado bruto, ou seja, um dado que não passou por interpretação e interferência humana. Outro problema em potencial é a *objetividade* do dado, sobre os quais não se questionavam apenas os usuários que tinham conhecimento sobre o processo de produção dos dados. O resultado geral de todos esses problemas é a redução de uso desses dados, tidos como suspeitos [Strong et al., 1997].

2.4.1 Qualidade de dados em métricas

Gopal em [Gopal et al., 2002] realizou um estudo sobre fatores determinantes para o sucesso em medição de PDS. Neste estudo o autor discute sobre a importância de uma implementação cuidadosa de um programa de métricas e a necessidade de se construir credibilidade sobre as métricas para o sucesso na melhoria de processo de software. Em seu estudo, o autor propõe um modelo de fatores determinantes, com basicamente dois tipos: (1) o contexto organizacional - o nível de comprometimento da organização com as métricas e (2) ambiente técnico - quão "amigável" é o programa de métricas. O ambiente em que as

métricas estão inseridas deve ser não intrusivo, ou seja, o esforço gasto para coleta das informações das métricas não pode adicionar carga de trabalho significativa à organização.

Dentre os fatores que compõem o ambiente técnico estão: (a) Análise das métricas, (b) Treinamento, (c) Uso de Ferramentas automatizadas, (d) Coleta de métricas, (e) Procedimento de coleta de métricas e (f) Qualidade das métricas coletadas. Esses estão diretamente envolvidos em fazer com que as métricas tenham maior acurácia, estejam no tempo certo e sejam fáceis de usar. Destacamos os fatores referentes a: (d) Coleta de métricas, que trata dos tipos de métricas que são coletadas e com qual frequência; (e) Procedimento de coleta de métricas, que quanto mais otimizado, mais positivo é o impacto na qualidade das métricas, ou seja, melhora a acurácia e confiabilidade do dado e, conseqüentemente, aumenta a confiança por parte dos usuários que tomarão decisões com base nesses dados; (f) Qualidade das métricas coletadas, onde a grande maioria de usuários de métricas concordam que estas devem ter precisão, relevância e deve prover informação útil.

Através do seu estudo, Gopal [Gopal et al., 2002] identificou que os gerentes de software precisam primeiro focar nos fatores do ambiente técnico e prover incentivos para os desenvolvedores de software usarem as informações de métricas nas suas ações diárias. Uma vez usadas as métricas, então os gerentes podem focar no contexto organizacional buscando a melhoria de desempenho da organização.

Dadas as colocações acima, a qualidade dos dados é a questão que merece destaque por sua relevância identificada por vários pesquisadores [Buneman et al., 2000], [Buneman et al., 2001], [Cui and Widom, 2000], [Gopal et al., 2002], [Chengalur-Smith et al., 1999] e também por ser objeto de estudo neste trabalho.

2.4.2 Confiabilidade e validade de dados

Na visão de [Kan, 2003], confiabilidade e validade são os dois critérios mais importantes de qualidade de métrica. Confiabilidade se refere à consistência de um número de medições usando o mesmo método de medição sobre o mesmo assunto. Se medições repetidas são altamente consistentes ou até semelhantes, então o método de medição ou as definições operacionais de medição têm um alto grau de confiabilidade. Por outro lado, se as variações entre medições repetidas são grandes, então a confiabilidade é baixa. A confiabilidade pode ser expressa em termos do desvio-padrão das medições repetidas. Quando duas variáveis são comparadas, geralmente o índice de variação (IV) é usado. Quanto menor o IV, mais confiável a medição. IV é dado pela Equação 2.1:

$$IV = \frac{\text{DesvioPadrao}}{\text{Media}} \quad (2.1)$$

Ainda na visão de [Kan, 2003], validade existe quando uma medição ou métrica realmente mede o que se espera que ela esteja medindo. Em alguns casos onde a medição não envolve um alto grau de abstração, por exemplo, a medição da altura do corpo, a validade é simplesmente a acurácia. O autor [Kan, 2003] ainda afirma que é importante esclarecer a diferença entre confiabilidade e validade. Medições confiáveis não são necessariamente válidos e vice-versa. A Figura 2.4 mostra graficamente a diferença entre os 2 critérios. Se a proposta da medição é estar no centro do alvo, vemos que a confiabilidade parece um padrão não muito rígido de qual círculo estar, isso porque a confiabilidade é uma função de consistência. Validade, por outro lado, é uma função de busca da “mosca”. Em termos estatísticos, se o valor esperado (ou a média) está no centro do alvo, então é válido, se a variação é pequena em relação ao círculo todo, então é confiável. Confiabilidade e validade são considerados termos de acurácia e exatidão respectivamente, quando não envolvem conceitos abstratos.

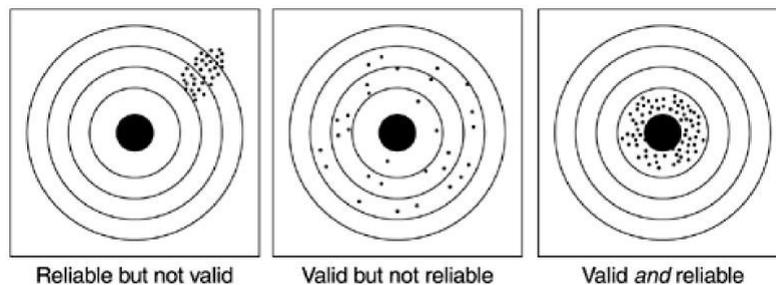


Figura 2.4 – Diferença entre validade e confiabilidade. Fonte:[Kan, 2003]

De acordo com [Kan, 2003] existem diversas maneiras para avaliar a confiabilidade de medições empíricas, entre elas: (a) método teste/reteste, (b) método alternativa/forma, (c) método divisão/metades e (d) método da consistência interna. Dentre esses, o método adotado pelo autor [Kan, 2003] para realizar seus experimentos foi o teste/reteste, por ser este o mais simples e que auxilia na interpretação das métricas de software, segundo o autor.

Apesar da existência desses métodos, suas características puramente estatísticas não atendem algumas especificidades da avaliação de qualidade de dados em um contexto específico como na coleta de métricas de esforço. Essas especificidades que tornam essa tarefa

não trivial são discutidas com detalhe no Capítulo 3. Por hora, estudamos outras alternativas para uma avaliação de qualidade dos dados de esforço adequada ao seu contexto, as quais são descritas na próxima seção.

2.5 Avaliação de qualidade de dados em métricas de esforço

De posse do conhecimento do impacto que uma organização pode ter como consequência do uso de dados de baixa qualidade, é possível inferir que uma avaliação dessa qualidade se faz necessária para apontar os pontos falhos dos dados para que a tomada de decisão não seja prejudicada pela falta de confiabilidade nos mesmos. No entanto, essa avaliação é um processo que está diretamente relacionado ao ambiente de inserção do dado e aos seus usuários [Redman, 2001]. Ainda de acordo com [Redman, 2001], um dado pode ser usado para diferentes usos e, conseqüentemente, pode estar bem adequado para algum uso, nem tanto para um segundo e definitivamente inadequado para um terceiro uso. Por isso, rotular a qualidade de um dado não é uma tarefa trivial, estando intimamente relacionada ao encontro de necessidades específicas de usuários também específicos.

Alguns usuários consideram que um dado é problemático se não possui acurácia, se não é atual, se o dado é de difícil interpretação e assim por diante. Muitos usuários vêem dados com algum problema como de baixa qualidade; por outro lado é difícil encontrar um dado que não tenha pelo menos algum desses problemas [Redman, 2001]. Por isso, as estratégias de avaliação que permitam essa flexibilidade no apontamento de problemas são mais adequadas no contexto de DQ. Essa flexibilidade diz respeito ao julgamento definitivo e rotulador, afirmando que o dado é bom ou ruim independente dos problemas que tem, ou dos que não tem, e nem na proporção e no impacto real do problema para o uso do dado. Em se tratando de métricas de software, esforço mais especificamente, a necessidade dessa flexibilidade se mantém.

Uma estratégia que pode ser considerada, neste caso, é o uso de lógica já que ela lida com aproximações e níveis de pertinência, isenta de caráter rotulador. A próxima seção introduz o assunto através de conceitos básicos, teoria de conjuntos e operações realizadas com essa lógica, para o processamento do conhecimento através da mesma.

2.5.1 Lógica difusa (fuzzy)

O termo *fuzzy* em inglês pode ter diferentes significados dependendo do contexto abordado, mas o conceito básico está geralmente ligado a algo **vago, indistinto, incerto**. Na língua portuguesa ainda não existe um consenso sobre a melhor tradução, mas os termos difuso e nebuloso são os mais utilizados e portanto os mais populares na comunidade. Por este motivo, neste trabalho é considerado o termo em inglês (*fuzzy*).

Na lógica clássica bivalente, um elemento x do universo de discurso U é pertencente a um dado conjunto A se esse satisfaz determinada propriedade e não pertence se não a satisfaz, conforme pode ser visualizado na Função 2.2:

$$X_A(u) = \begin{cases} 0, & \text{se } x \notin A \\ 1, & \text{se } x \in A \end{cases} \quad (2.2)$$

De acordo com [Rezende, 2003], em situações reais de avaliação, existem propriedades que são vagas, incertas ou imprecisas e impossíveis de serem caracterizadas por predicados da lógica clássica bivalente. A teoria dos conjuntos fuzzy pode ser vista como uma extensão da teoria clássica e foi criada para tratar graus de pertinência intermediários entre a total (1) e a não-pertinência (0). Assim, na lógica fuzzy, o grau de pertinência de um elemento em relação a um conjunto é definido por uma função que assume como valor qualquer número real pertencente ao intervalo fechado $[0, 1]$, algebricamente mapeada por 2.3:

$$\mu_P(x) : U \rightarrow [0, 1] \quad (2.3)$$

No processo de tomada de decisão (ou avaliação) os humanos relacionam seu conhecimento a fatos com características incertas sem definir um resultado exato, por exemplo o uso de “*mais quente*” ou “*menos quente*” no lugar da bivalente classificação *quente/frio*. Para um melhor entendimento de como funciona a representação fuzzy de conhecimento é interessante abordar três diferentes tipos de incertezas que podem estar caracterizadas no processo de decisão humano: (a) estocástica; (b) léxica e (c) lingüística [Altrock, 1995].

A incerteza estocástica é um princípio de incerteza matemática que lida com a incerteza quanto à ocorrência de um determinado evento, por exemplo: “*A probabilidade de acertar o alvo é de 0,8*”. O evento “*acertar o alvo*” é bem definido e a incerteza está em quando isto ocorrerá ou não. Essa incerteza é quantificada por um grau de probabilidade, podendo ser assim processada usando métodos estocásticos como por exemplo redes *Bayesianas*.

Outro tipo de incerteza é a léxica que trata com a imprecisão presente na maioria dos conceitos humanos que derivam suas conclusões, como exemplo “*homem alto*”, “*dia quente*”,

onde não existe uma definição fixa do que seja um homem alto ou um dia quente. Além disso, a realidade do humano que está concluindo tais situações tem um papel importante já que para uma criança o conceito de homem alto é diferente de uma pessoa adulta. Até mesmo para pessoas adultas o conceito de “*homem alto*” pode variar. Um exemplo de afirmação baseada nesse tipo de incerteza é “*Provavelmente nós teremos um ano financeiro de sucesso*”. Em um primeiro momento esta afirmação pode parecer muito similar à primeira (estocástica), no entanto, existem diferenças significativas, tais como: (1) o evento propriamente dito não está definido (para algumas companhias um ano financeiro de sucesso pode ter um sentido e para outras, outro); (2) nessa afirmação não existe uma quantificação probabilística, logo essa probabilidade é percebida mas não matematicamente explícita como na estocástica.

Afirmações do tipo léxica têm um importante papel no processo humano de tomada de decisão, apesar de não apresentarem conteúdos quantitativos. Em muitos casos, a incerteza adicionada através do uso de palavras adicionam um certo grau de flexibilidade. Essa flexibilidade e a sua definição são características da incerteza linguística, que na verdade, é uma caracterização da incerteza léxica com uma certa flexibilidade de conceitos. Essa flexibilidade pode ser notada no seguinte exemplo: “*Sindicatos e indústrias querem alcançar o mesmo objetivo: um adequado aumento de salários. O problema está quando cada um deve se manifestar sobre o que é um salário adequado*”.

A lógica fuzzy permite a representação do conhecimento humano considerando a incerteza léxica fazendo uso de artifícios linguísticos e de marcações matemáticas através de cálculo de probabilidades. Para essa representação fuzzy de conhecimento são utilizadas regras de produção fuzzy, variáveis linguísticas, funções de pertinência, operações fuzzy (aqui são exploradas apenas as básicas) e modelos de inferência fuzzy. Cada um desses conceitos é apresentado a seguir.

2.5.1.1 Regras de produção fuzzy

Uma base de conhecimento fuzzy geralmente é construída por meio de regras de produção fuzzy. Uma regra de produção é normalmente formada de duas partes principais *se* < antecedente > *então* < conseqüente >, onde o antecedente é composto por um conjunto de condições que, quando satisfeitas (mesmo que parcialmente), determinam o processamento do conseqüente da regra por um mecanismo de inferência fuzzy. O conseqüente é composto por um conjunto de ações que são geradas com o disparo da regra

(satisfação do antecedente). Como exemplo de regras de produção considere um controlador de força de um guindaste [Rezende, 2003]:

$$SE \textit{Distancia} = \textit{baixa} \textit{ENTÃO} \textit{Força} = \textit{alta}$$

2.5.1.2 Variáveis lingüísticas e termos

Na construção das regras de produção, devem ser definidas variáveis lingüísticas que fazem parte dos antecedentes e dos conseqüentes. As variáveis lingüísticas são entidades utilizadas para representar de modo impreciso, e portanto, lingüístico, um conceito ou uma variável de um dado problema. No exemplo utilizado acima (controlador da cabeça de um guindaste) as variáveis lingüísticas utilizadas são *Distância* e *Força*. Para cada variável lingüística devem ser definidos os possíveis valores que ela pode assumir, chamados **termos**.

Considerando ainda o exemplo anteriormente citado, podem ser definidas as seguintes variáveis lingüísticas (Distância, Ângulo e Força) com os seus respectivos termos da seguinte forma de acordo com [Altrock, 1995]:

$$\textit{Força} \in \{\textit{negh} - \textit{high}, \textit{neg} - \textit{medium}, \textit{zero}, \textit{pos} - \textit{medium}, \textit{pos} - \textit{high}\}$$

$$\hat{\textit{Angulo}} \in \{\textit{pos} - \textit{big}, \textit{pos} - \textit{medium}, \textit{zero}, \textit{neg} - \textit{medium}, \textit{neg} - \textit{big}\}$$

$$\textit{Distância} \in \{\textit{far}, \textit{medium}, \textit{close}, \textit{zero}, \textit{too} - \textit{far}\}$$

2.5.1.3 Funções de pertinência

A função de pertinência é a valoração dos termos definidos para cada variável lingüística. Para um dado elemento x do universo de discurso, o valor de pertinência $\mu_A(x)$, calculado pela função de pertinência, representa o quanto este elemento satisfaz o conceito representado pelo conjunto fuzzy A . A Figura 2.5 mostra a definição gráfica de uma função de pertinência para o exemplo utilizado acima.

Para cada entrada do sistema de inferência fuzzy são calculados todos os valores de pertinências daquela entrada para cada função de pertinência. Por exemplo, se o sistema de inferência recebe como entrada 12 metros para a variável lingüística Distância a produção das regras terá o seguinte processamento através das funções de pertinência:

A Distância de 12 metros é um membro de cada termo no grau de 0.1 para MuitoLonge, no grau de 0.9 para Médio, no grau de 0 para Perto, no grau de 0 para Zero e no grau de 0 para Longe, conforme pode ser visualizado na Figura 2.5 onde está a linha pontilhada, onde a abcissa (eixo x) é a distância em metros e a ordenada (eixo y) é o valor fuzzy.

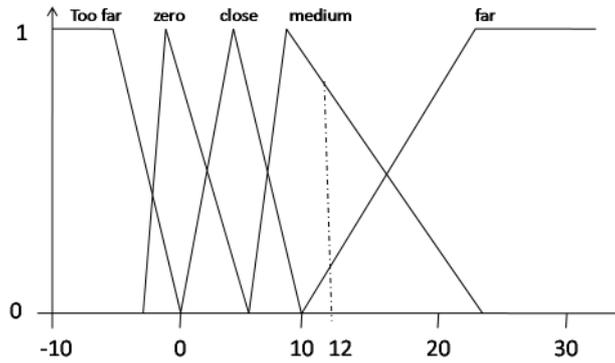


Figura 2.5 – Exemplo de cálculo da variável *Distância*. Fonte: [Altrock, 1995]

2.5.1.4 Operações fuzzy

No caso da lógica fuzzy, as operações booleanas não podem ser utilizadas porque não cobrem as incertezas de “mais verdadeiro” ou “menos verdadeiro” presentes nas sentenças fuzzy. Por isso, novas operações foram definidas para os cálculos de conectivos AND, OR e NOT. Essas operações são ilustradas na Tabela 2.4.

Tabela 2.4 – Conjunto de operações fuzzy

Operador	Operação
AND	$\mu_{A \wedge B} = \min \{ \mu_A, \mu_B \}$
OR	$\mu_{A \vee B} = \max \{ \mu_A, \mu_B \}$
NOT	$\mu_{\sim A} = 1 - \mu_A$

Através das operações a cima descritas é possível processar regras do tipo:

Regra1: *SE Distancia = média AND Ângulo = pequenonegativo ENTÃO Força = mediopositivo*

Regra2: *SE Distancia = média OR Ângulo = zero ENTÃO Força = zero*

A Tabela 2.5 ilustra como ficaria o processamento dessas regras com as seguintes entradas:

Tabela 2.5 – Exemplo de processamento de regras

Regra	Ação	Resultado
Regra 1	$\min \{ 0, 9; 0, 8 \}$	0,8
Regra 2	$\max \{ 0, 9; 0, 2 \}$	0,9

2.5.1.5 Métodos de *Defuzzification* (defuzzificação)

Um modelo de inferência fuzzy recebe como entrada um valor escalar (ou *crisp*) no mundo de observação que passa por uma máquina de inferência que se utiliza de um banco de regras para converter o valor escalar em fuzzy e por fim, este resultado passa por um processo de conversão de fuzzy para escalar, conforme mostra a Figura 2.6. Este sistema de inferência é chamado Mamdani proposto em 1970[Rezende, 2003].

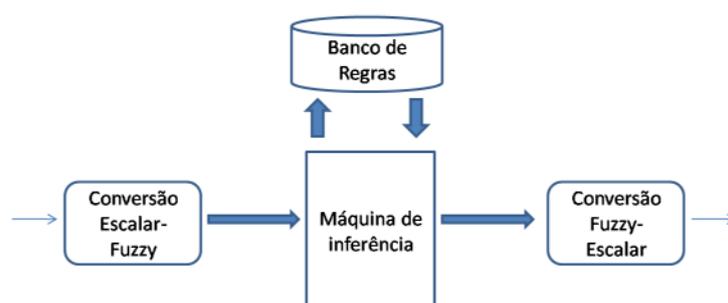


Figura 2.6 – Diagrama de um modelo de inferência Mamdani. Fonte: [Rezende, 2003]

A última etapa de um sistema de inferência é a chamada defuzzificação (apesar de não existir um consenso na comunidade sobre a melhor tradução para o português, neste trabalho usamos “defuzzificação”). Para a execução desta etapa existem métodos que se baseiam no cálculo de área para a conversão fuzzy-escalar: (1) Centro do Máximo (CoM), (2) Centro de Gravidade (CoG) e (3) Média do Máximo (MoM)[Altrock, 1995]. O método mais comum é o Centro de Gravidade onde primeiro cortam-se as funções de agregação no grau de validade do termo respectivo, e as áreas sob o resultado do corte são então sobrepostas e o cálculo do centro desta área aponta para o valor “defuzzyficado”, ou seja, o valor escalar correspondente à inferência da base de regras. A Figura 2.7 ilustra o método de defuzzyficação para o exemplo em que os valores fuzzy 0,8 e 0,2 para os termos médiopositivo e zero, respectivamente, onde a abcissa (eixo x) é a distância em metros e a ordenada (eixo y) é o valor fuzzy. Para este exemplo o resultado da defuzzyficação é a força de 8,2 Kw.

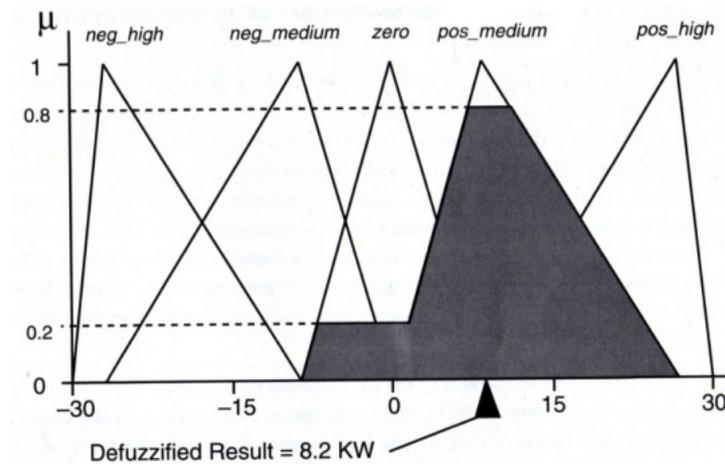


Figura 2.7 – Método de defuzzificação Centro de Gravidade para a variável Força. Fonte: [Altrock, 1995]

2.5.2 Data Provenance

Estudos referentes a *Data Provenance* (Proveniência de dados) abordam a problemática de conhecer o processo de obtenção de dados, no sentido de fonte, tempo e confiabilidade. Domínios científicos usam diferentes formas de DP (*Data Provenance*) e com diferentes propósitos. Dependendo desse domínio, DP pode ser descrita de maneira diferente. Em Ciência da Computação, *provenance* - também chamada de “*lineage*” ou “*pedigree*” - descreve a fonte e/ou a derivação do dado. No contexto de Banco de Dados, Buneman [Buneman and Tan, 2007] define DP como a descrição das origens do dado e o processo pelo qual este foi produzido para que chegasse na base de dados. O autor ainda afirma que é essencial na avaliação da qualidade de uma base de dados o conhecimento de onde vem os dados de origem. Já no contexto de SIG (Sistemas de Informações Geográficas), Lanter citado por [Yogesh L. Simmhan, 2005] caracteriza como linhagem informações que descrevem as transformações que o dado sofreu. Outros autores acrescentam que DP pode ser vista como metadados que registram o processo de experimentos e as anotações sobre a execução dos mesmos [Yogesh L. Simmhan, 2005].

Para os propósitos da presente pesquisa, DP é considerada como toda informação sobre o dado que contribui na determinação do processo de derivação e/ou transformação do mesmo, abrangendo toda influência externa sofrida desde a sua forma original até a sua forma considerada final.

A Figura 2.8 mostra uma taxonomia das características de DP, criada por Simhan em [Yogesh L. Simmhan, 2005], que facilita a compreensão do seu uso nos diferentes contextos.

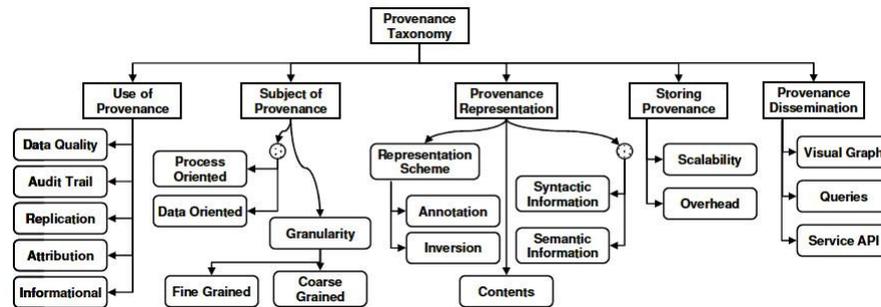


Figura 2.8 – Taxonomia de Data Provenance. Fonte: [Yogesh L. Simmhan, 2005]

A taxonomia acima ilustrada mostra os principais propósitos de uso de DP (*Use of Provenance*) podendo ser voltada à preocupação com qualidade de Dados, auditorias, como recurso de replicação de dados, ou de propósito de atribuições de dados e/ou de informações. Na presente pesquisa o uso que se aplica para é para Data Quality (qualidade de dados).

O domínio aplicado de DP (*Subject of Provenance*) pode estar orientado tanto a dados (*Data Oriented*) quanto a processo (*Process Oriented*), o que é indicado pelo círculo pontilhado na Figura acima. Independente dessa orientação, a granularidade de coleta e armazenamento dos dados é uma decisão importante no uso de DP, podendo esta ser mais detalhada (*Fine Grained*) ou menos detalhada (*Coarse Grained*). Esse nível de detalhamento pode influenciar diretamente na utilidade dos dados coletados e no custo de manter a proveniência dos dados. De acordo com Simhan em [Yogesh L. Simmhan, 2005], o custo pode ser inversamente proporcional à granularidade de coleta, ou seja, quanto menor a granularidade, mais custoso é o processo de rastrear DP.

A granularidade mais abrangente é onde se registra um histórico completo da derivação de um determinado conjunto de dados. Essa abordagem é chamada “*workflow*” ou “*coarse-grain provenance*” [Buneman and Tan, 2007]. Outra granularidade possível de ser utilizada é chamada “*fine-grain provenance*”, um cálculo da derivação de parte do dado resultante. De acordo com [Buneman et al., 2000] essa abordagem tem algumas vantagens com relação à *coarse-grained* nos seguintes sentidos:

- Na maioria dos casos, o *workflow* inteiro pode ser bastante complicado de se rastrear;
- Todo *workflow* pode não estar disponível;
- A mais simples caracterização de um *workflow* pode ser o log ou registro de ações em componentes individuais da base de dados.

Dentro da abordagem “*fine-grain provenance*”, [Buneman et al., 2001] descreve difer-

entes tipos de proveniência: *where-provenance* (onde-proveniência) e *why-provenance* (por que-proveniência). *Where-provenance* se refere à justificativa para um elemento aparecer no resultado de uma consulta, no sentido de esclarecer de **onde** o dado surgiu. Considerando o seguinte exemplo extraído de [Buneman et al., 2001]:

```
SELECT Emp.name, Dept.dname
FROM Emp, Dept
WHERE Emp.deptid = Dept.id
```

Assumindo (Kim, CS) como resultado da execução da query, o *where-provenance* de “Kim” é o atributo *name* de alguma tupla de Emp onde o valor é Kim. A identidade de qual tupla de Emp cujo valor é “Kim” é feito através do *why-provenance* de (Kim, CS). *Why-provenance* de (Kim, CS) envolve a query SQL e mais a tupla de Emp e a tupla de Dept com as seguintes propriedades: (a) satisfaz a cláusula where da query, (b) o atributo *name* da tupla de Emp é “Kim” e (c) o atributo *dname* da tupla de Dept é “CS” [Buneman and Tan, 2007].

Para a representação, armazenamento e visualização das informações referentes a proveniência dos dados existem diferentes técnicas. A escolha da melhor maneira de coletar e armazenar a proveniência está relacionada ao uso pretendido dos dados, o custo, a granularidade e até mesmo os usuários finais desses dados. Algumas dessas técnicas são apresentadas abaixo:

- Chimera [Foster et al., 2002] é um sistema que provê um catálogo de dados virtuais para representar os dados derivados e o procedimento de derivação dos mesmos. As principais motivações para desenvolver essa aplicação foi a possibilidade de encontrar erros em dados derivados e manter um registro da história de construção dos dados. Para testar a aplicação, foram usados dados científicos na área de Física e Astronomia, os quais encontravam-se armazenados em arquivos. Após os dados serem coletados desses arquivos, transformados, calculados e gerados foi criada uma base de dados réplica contendo o processo (*lineage* - linhagem) pelo qual cada dado passou, o que agregou um custo e tempo no registro da proveniência dos dados. Esta aplicação é baseada na abordagem do tipo coarse-grain provenance e utiliza Annotation, que é uma técnica geralmente adotada em abordagens desse tipo.

- POESIA [Fileto et al., 2003] é uma aplicação de DP que utiliza ontologias para correlacionar escopos dos dados e a granularidade dos dados estimando, assim, a proveniência dos dados. Essa abordagem expressa os dados e os efeitos das operações de integrações entre conjuntos de dados. A motivação para o desenvolvimento dessa aplicação foram problemáticas relacionadas a aplicações da agricultura, que se utiliza de dados de medições meteorológicas, muitos dos quais são produzidos por cálculos. O conhecimento das fontes originais do dado derivado e o processo de derivação são determinantes na confiabilidade das análises das medições, pois estes dados são oriundos de bases de dados heterogêneas e são sumarizados em um data warehouse, podendo neste processo de transação serem inseridos ruídos nos valores. A abordagem *fine-grain provenance* foi o tipo de DP considerada.
- TRIO [Widom, 2005] é uma aplicação de DP que rastreia informações de proveniência em dados armazenados em Data Warehouse, que considera acurácia e dados de proveniência como componentes inerentes. A autora deste trabalho defende que atividades de mineração e atualização em dados de DW motivam a rastreabilidade da proveniência dos dados, e que qualquer sistema científico que usa queries em bases de dados, funções de workflow e atividades de transformações de dados podem aplicar técnicas de DP. Para cada um dos itens do TRIO (dado, acurácia e proveniência) são definidos tipos de informações que são rastreadas e em qual granularidade essa rastreabilidade corre. Esta aplicação é baseada na abordagem *fine-grain provenance*. Foi testada em um exemplo específico de Biologia, que consiste na coleta e armazenamento de dados relacionados a observações da localização dos pássaros em uma época específica do ano usado para entender tendências nas populações de pássaros e correlacionar a vida dos pássaros com condições do ambiente. Na implementação, os dados coletados são modelados de acordo com uma linguagem de modelagem específica para suportar o TRIO e para serem consultados via *query-inverse* que é um tipo de consulta específica dessa área de DP, onde são utilizados recursos de bancos de dados para utilizá-la. Nesse trabalho a autora ressalta a aplicabilidade de *fine-grain provenance* em domínios como Biologia, Química e Astronomia.
- Buneman et. al. [Buneman et al., 2006] aplicam técnicas de DP para capturar linhagem dos dados quando estes são copiados manualmente de várias bases de dados fonte para uma base destino. Essa aplicação captura a linhagem do dado paralelamente à alteração do dado na base destino dentro do domínio de Bioinformática. Os

fatores motivacionais para aplicação de DP foram situações onde pesquisadores de biologia copiavam dados de diversas fontes, muitas vezes online, e colavam em uma base única para prover visão unificada dos dados, e passado determinado tempo, os pesquisadores precisavam ter conhecimento de como determinados dados foram gerados, o que não era possível. Além disso, como alternativa, os pesquisadores tentavam buscar novamente os dados nas fontes, e estes já haviam sofrido atualizações. Esta é uma aplicação baseada na abordagem fine-grain provenance e implementa provenance queries para o registro da linhagem. Nesse trabalho o autor afirma que a aplicação de DP é uma proposta que melhora a conabilidade e a transparência de registros de dados científicos.

A Tabela 2.6 ilustra as principais diferenças entre as técnicas considerando o domínio de aplicação e algumas das características presentes na taxonomia anteriormente apresentada, entre elas: o propósito de uso (*Use of Provenance* na taxonomia), a granularidade (*Subject of Provenance* na taxonomia) e a técnica adotada (*Provenance Dissemination* na taxonomia) para a representação, armazenamento e visualização de DP.

Tabela 2.6 – Principais diferenças entre técnicas de DP. Fonte: [Berardi, 2007]

	CHIMERA [2002]	POESIA [2003]	TRIO [2005]	BUNEMAN [2006]
Domínio	Física, Astronomia	Agricultura	Biologia	Bioinformática
Propósito	Auditoria de dados Histórico de derivação	Auditoria de dados Histórico de derivação Interpretação de dados	Auditoria de dados Histórico de derivação Qualidade de dados	Replicação de dados Histórico de derivação
Granularidade	Coarse-grain	Fine-grain	Fine-grain	Fine-grain
Técnica	Annotation	Ontologias	Query-Inverse	Provenance Queries

A análise da tabela possibilita observar que em diferentes domínios os propósitos podem coincidir ou não e as granularidades também podem variar. Dentre outros propósitos, na tabela é possível concluir que os mais comuns são:

- Qualidade de dados: qualidade e a confiabilidade do dado pode ser baseada nos dados fontes e nas transformações que estes sofreram para produzir um dado final;
- Auditoria de dados: para a auditoria de dados é útil o rastreamento do dado, a determinação das fontes usadas e a detecção de erros na geração dos dados;
- Replicação de dados: o conhecimento detalhado sobre a proveniência do dado pode permitir a repetição da derivação do dado, ajudando a manter a acurácia do mesmo

Tendo em vista todas as características aqui apontadas, Data Provenance se mostra uma alternativa bastante promissora na área de Qualidade de Dados e é portanto, efetivamente adotada na proposta de solução para o problema a seguir apresentado.

2.6 Observações sobre o capítulo

Neste capítulo foram expostos conceitos envolvidos na medição de PDS. Foram descritos alguns modelos de processos e ilustramos a idéia de medição de software a partir de um modelo de maturidade largamente utilizado e conceituado como o CMMI. Além disso, foram estudadas propostas de validação de métricas utilizadas na medição de PDS cuja abordagem é voltada ao funcionamento eficiente das métricas. Também foram estudados conceitos relacionados a lógica fuzzy e data provenance que servem de base para o entendimento da sua utilização como alternativa na solução do problema abordado pelo trabalho. Portanto, todos os conceitos estudados nesta Fundamentação Teórica contribuem para o entendimento do desenvolvimento seguinte do trabalho apresentado nos Capítulos seguintes.

A seguir é discutido o cenário de pesquisa onde a abordagem é voltada à eficiência e confiabilidade dos dados que representam as métricas, estando elas validadas ou não. Após a apresentação do cenário, é exposto o problema corrente e os requisitos de uma solução que elimina ou ameniza os problemas encontrados no cenário.

3

Cenário

Este capítulo descreve o cenário onde a pesquisa está inserida. Apresenta o fluxo de coleta de métrica de esforço em um projeto real e descreve uma análise estatística realizada para observar os problemas existentes nesse fluxo. Uma vez apresentados esses problemas, são apontadas algumas características essenciais para que uma solução sirva de alternativa para amenizar ou eliminar esses problemas. Essas características, que chamamos de requisitos da solução, nortearão o estudo de trabalhos relacionados (Capítulo 4) e , conseqüentemente, o desenvolvimento da solução proposta (Capítulo 5).

3.1 Descrição

A HP (*Hewlett-Packard Company Brasil Ltda*) - empresa de renome internacional, possui no TECNOPUC - Parque Tecnológico da PUC em Porto Alegre RS, uma operação de software, onde são realizadas atividades de desenvolvimento de software. Visando atrair mais clientes e adquirir maior credibilidade nesse segmento, em 2002 a organização iniciou a busca pelo CMM3. Em novembro de 2005, após um grande esforço por parte da organização e seus colaboradores, foram alcançados os requisitos para elevá-la ao nível 3 do CMM , sendo parte deste esforço em parceria com a PUCRS. Um dos aspectos que merece destaque com relação a esses requisitos, é a utilização de um repositório central de métricas coletadas de diferentes projetos da organização, chamado de BO (Base Organizacional). Este repositório central tem por objetivo prover uma visão unificada da realidade dos projetos, visando auxiliar os gestores no processo de tomada de decisão [Kulpa and Johnson, 2003].

A pesquisa do presente trabalho está focada nas circunstâncias de coleta e armazenamento das métricas. Para isso são utilizados dados referentes a um projeto de manutenção

específico da HP, que chamaremos de P1 por questão de sigilo. O PDS do projeto P1 é o modelo cascata orientado a entregáveis. O projeto P1 utiliza quatro ferramentas em seu gerenciamento, sendo que para o lançamento de horas de esforço é utilizado um software desenvolvido localmente. A Figura 3.1 ilustra o fluxo que a métrica percorre desde o seu lançamento até o seu armazenamento no repositório central nos projetos da organização.

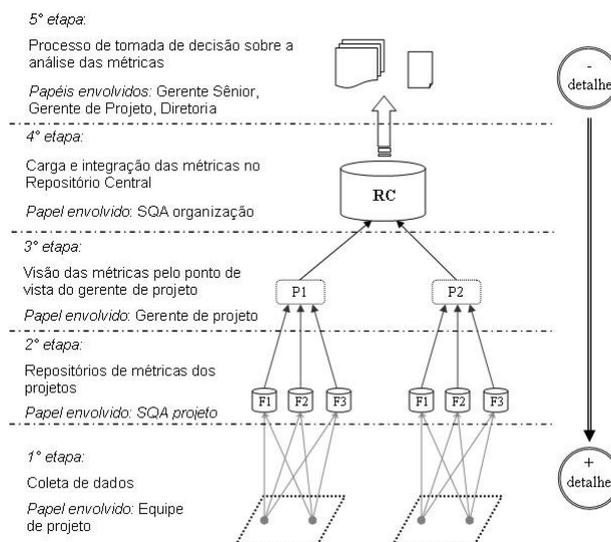


Figura 3.1 – Fluxo das métricas

Os colaboradores atuantes nos respectivos projetos (representado na Figura 3.1 pelos paralelogramas) obtêm as métricas no PDS ilustrado como a primeira etapa na Figura. Como segunda etapa ocorre o registro dessas métricas nas diferentes ferramentas que o projeto tem para apoio no gerenciamento, são elas: MS Project da Microsoft e *Project and Portfolio Management* (PPM) da Mercury (cronograma); IBM *Rational Clear Quest* da IBM (acompanhamento e controle de defeitos) e IBM *Rational RequisitePro* da IBM (requisitos). Na terceira etapa ocorre a construção de pacotes onde são agrupados os dados mensais, proporcionando uma visão do projeto naquele período. A quarta etapa é ilustrada pelo armazenamento dos pacotes no repositório central para, na quinta e última etapa, serem extraídos e apresentados aos gerentes e gestores.

3.2 O Problema

Através de um estudo realizado nessa operação de software foi possível identificar algumas dificuldades em relação à confiabilidade dos dados de esforço. Para evidenciar esses problemas de qualidade dos dados foi executada uma avaliação estatística por Berardi em [Berardi, 2007], baseada em correlação linear para observar se existia ruptura na homogeneidade dos dados ao longo das medições realizadas no projeto P1. Como resultados dessa avaliação, identificou-se uma mudança de comportamento dos dados quando estes eram coletados por processos de coletas distintos. A avaliação executada está descrita abaixo.

3.2.1 Métrica para avaliação estatística

A métrica escolhida para essa avaliação em questão foi a de esforço de retrabalho pré-release, que registra o número de horas que a equipe do projeto gastou para realizar consertos de defeitos detectados antes da entrega da release. Esta métrica apresenta informações bastante relevantes para a organização. Por exemplo, tempo gasto em retrabalho significa prejuízo para a organização no sentido de que a equipe não está produzindo, e tempo que não é gasto para produzir é sinônimo de desperdício para a organização. Além disso, o conhecimento do valor dessa métrica pode apontar falta de qualidade na equipe do projeto. Tendo em vista a importância dos dados dessa métrica, é de total interesse da organização ter confiança nos respectivos dados, independente da forma de coleta dos mesmos. A falta de qualidade desses dados pode ter impacto significativo dentro da organização, e ter como consequência o desuso do repositório central de métricas, o que geralmente acontece com os sistemas que dão esse tipo de suporte à decisão [Chengalur-Smith et al., 1999].

3.2.2 Método de análise

Conforme discutido na Fundamentação Teórica (Capítulo 2), existem vários métodos para se avaliar a confiabilidade de dados de medições, e o utilizado neste caso é o teste/reteste. O teste/reteste consiste em obter uma medição de um conjunto de indivíduos e, após todos os indivíduos serem medidos, obtém-se outra medição imediatamente após a primeira (ou ainda, algum tempo depois), sendo que esta distância entre a primeira medição e a segunda deve ser igual para todos os indivíduos. De posse dos dois conjuntos de medições

dos mesmos indivíduos, calcula-se o índice de correlação entre os valores desses conjuntos. O índice de correlação indicará o grau de confiabilidade dos dados, ou seja, quanto maior o índice, maior a confiabilidade [Kan, 2003].

Baseado nesse método, avaliamos o comportamento dos dados da métrica de retrabalho pré-release, comparando os índices de correlação obtidas com medições de cada processo de coleta utilizado (no caso, chamaremos de processo de coleta A e processo de coleta B). O objetivo é rejeitar ou aceitar a seguinte hipótese: O índice de correlação entre os dados das métricas deve se manter alto quando esses são coletados através do mesmo processo e através de processos diferentes. Se a hipótese for aceita, ou seja, se o índice de correlação entre os dados de medições se manter alto, independente do processo de coleta, então significa que o comportamento dos dados se mantém homogêneo. Por outro lado, se a hipótese for rejeitada, ou seja, se o índice de correlação diminuir conforme o processo de coleta se altera, então é possível inferir que a qualidade dos dados no que diz respeito a confiabilidade está dependendo do seu método de coleta, o que pode ser uma característica prejudicial para os gestores na tomada de decisão, pois esta fica dependente da eficiência da coleta dos dados das métricas e conseqüentemente dos dados coletados.

Para realizar essa comparação, foram definidos os seguintes passos:

1. Definir os indivíduos dos quais são utilizadas as medições para avaliação. No contexto de mensuração de PDS, os indivíduos podem ser de diferentes releases em que a equipe de projeto está envolvida.
2. Para o processo de coleta A:
 - (a) Obter uma medição (medição 1) da métrica de esforço de retrabalho pré-release para cada um dos indivíduos definidos.
 - (b) Obter outra medição (medição 2) da métrica de esforço de retrabalho pré-release para os mesmos indivíduos em (a).
 - (c) Calcular o índice de correlação entre os dados desses dois conjuntos de medições.
3. Para o processo de coleta B:

- (a) Obter uma medição (medição 3) da métrica de esforço de retrabalho pré-release para cada um dos indivíduos definidos.
 - (b) Obter outra medição (medição 4) da métrica de esforço de retrabalho pré-release para os mesmos indivíduos em (a).
 - (c) Calcular o índice de correlação entre os dados desses dois conjuntos de medições.
4. Calcular o índice de correlação entre os dados das medições obtidas através do processo de coleta A com os dados das medições obtidas através do processo de coleta B.

Através do passo 1 são definidos os objetos de medição de acordo com as especificidades do contexto, garantindo assim que ao longo de toda a avaliação são considerados os mesmos indivíduos. O objetivo dos passos 2 e 3 é avaliar se o comportamento dos dados é homogêneo quando se utiliza o mesmo processo de carga. Já no passo 4, o objetivo é avaliar se os dados, coletados de diferentes maneiras, apresentam homogeneidade e por conseqüência, confiabilidade.

3.2.3 Forma de comparação

Para realizar a comparação dos dados da métrica de retrabalho pré-release, coletados com os diferentes processos de coleta, foram efetuadas duas avaliações. Como pode ser visualizado na Tabela 3.1, para utilizar o método TESTE/RETESTE foram utilizados pares de medições da métrica em questão. A primeira avaliação (Avaliação 1) consiste em utilizar três releases do projeto P1, duas medições que foram realizadas através do processo de coleta A - as quais ocorreram nos meses X e Y - e, as duas medições que foram realizadas através do processo de coleta B - que ocorreram nos meses P e Q.

Tabela 3.1 – Tabela de avaliações

	AVALIAÇÃO 1	AVALIAÇÃO 2
RELEASES	10.05 10.07 11.00	10.05 10.07 11.00 11.05 12.00
MEDIÇÃO – PROCESSO A	MÊS X (Medição 1) MÊS Y (Medição 2)	MÊS Z (Medição 1) MÊS W (Medição 2)
MEDIÇÃO – PROCESSO B	MÊS P (Medição 3) MÊS Q (Medição 4)	MÊS P (Medição 3) MÊS Q (Medição 4)

Na segunda avaliação (Avaliação 2) são utilizadas as mesmas releases com a adição de duas novas (11.05 e 12.00) e por causa dessa adição foi necessário utilizar diferentes medições (no caso, mês Z e W) pois nas medições anteriores essas releases não tinham medições coletadas. As duas medições realizadas através do processo B são as mesmas utilizadas na Avaliação 1. Decidiu-se realizar a segunda avaliação para obter um conjunto maior de medições, visto que a Avaliação 1 foi feita baseada em 3 pares de medições, e ao mesmo tempo, confirmar ou levantar novos resultados obtidos com a primeira avaliação. Os dados das métricas para as avaliações (medições) foram extraídos da BO.

Para atingir o objetivo de aceitar ou rejeitar a hipótese de que “o índice de correlação entre os dados das métricas deve se manter alto quando esses são coletados através do mesmo processo e através de processos diferentes” efetuamos a análise dos resultados de acordo com as seguintes etapas:

- observação dos índices de correlação entre os dados obtidos com o mesmo processo de coleta;
- observação dos índices de correlação entre os dados obtidos com processos de coleta distintos;
- comparação dos índices de correlação quando há mudança no processo de coleta.

3.2.4 Resultados

Através da execução da Avaliação 1, obteve-se os seguintes índices mostrados na Tabela 3.2

Tabela 3.2 – Tabela de resultados avaliação 1

Avaliação 1		
Tipo de Processo de coleta	Medições	Índice de correlação
Igual	Medição 1 e 2 (A)	$r = 0,9491$
	Medição 3 e 4 (B)	$r = 0,9998$
Diferente	Medição 1 e 3	$r = 0,1516$
	Medição 1 e 4	$r = 0,1373$
	Medição 2 e 3	$r = 0,4552$
	Medição 2 e 4	$r = 0,4423$

Observando os índices de correlação calculados entre os dados obtidos com o “Tipo de processo de coleta” Igual, pode-se inferir, de acordo com o método teste/reteste, que a confiabilidade dos dados pode ser considerada alta uma vez que os índices de correlação apresentam valores bastante próximos de 1 e, como é sabido, o valor 1 corresponde ao índice máximo de correlação. Por outro lado, observando os índices de correlação entre os dados obtidos com o “Tipo de processo de coleta “ Diferente, nota-se que há diferença bastante significativa em relação aos índices de “Tipo Igual”, e além de diferentes, são muito baixos, onde os dois primeiros índices são quase próximos a 0 e, como é sabido, o valor 0 corresponde ao índice mínimo (ou até mesmo inexistente) de correlação. Apesar dos dois últimos índices do “Tipo Diferente”, não serem tão próximos de zero não significa que estes tenham índices altos se comparados com os índices de “Tipo Igual”.

Com base nesses resultados, pode-se rejeitar a hipótese inicial, pois o índice de correlação entre os dados das métricas não se manteve alto quando esses foram coletados através de processos distintos.

Através da execução da Avaliação 2, obteve-se os seguintes índices mostrados na Tabela 3.3:

Tabela 3.3 – Tabela de resultados avaliação 2

Avaliação 2		
Tipo de Processo de coleta	Medições	Índice de correlação
Igual	Medição 1 e 2 (A)	$r = 0,9730$
	Medição 3 e 4 (B)	$r = 0,9942$
Diferente	Medição 1 e 3	$r = 0,7192$
	Medição 1 e 4	$r = 0,6474$
	Medição 2 e 3	$r = 0,8188$
	Medição 2 e 4	$r = 0,7656$

Como o objetivo motivador para a execução da Avaliação 2 é de confirmar os resultados da Avaliação 1 ou até mesmo levantar novos fatores, os índices de correlação na Tabela 3.3 foram observados buscando identificar características que ratificam os primeiros resultados ou que mostram novos fatores.

Como foi feito na primeira avaliação, observando os índices de correlação calculados entre os dados obtidos com o “Tipo de processo de coleta” Igual, pode-se reafirmar que a confiabilidade dos dados é alta devido aos altos índices de correlação encontrados, de acordo com o método teste/reteste. Contudo, observando os índices de correlação entre os dados obtidos com “Tipo de processo de coleta “ Diferente, identificam-se valores de r mais baixos, no entanto não tão baixos quanto os encontrados na Avaliação 1. Entretanto, apesar de serem consideravelmente maiores, em outras palavras, mais próximos de 1, continuamos afirmando que os índices não são altos com base no que afirma Triola em [Triola, 1999].

Triola fala que eventualmente a interpretação do índice de correlação r com a utilização dos critérios “próximo de” 0 ou 1 é vaga, e em casos onde os pares de dados amostrais são pequenos esta interpretação pode ser mais complicada ainda. Com o intuito de facilitar a interpretação dos índices com amostras de pares pequenos, o autor construiu uma tabela com valores usuais de r calculados com amostras genéricas de dados e definiu para cada quantidade de pares um valor crítico para r , o qual pode ser considerado nas interpretações de cálculo de índice de correlação com amostras pequenas.

No contexto da Avaliação 2, onde são utilizados 5 pares de dados (duas medições para cada uma das cinco releases), o autor indica que um índice de correlação só será considerado alto se este for maior que 0,878 com uma precisão de 95% de chance de não ocorrer erros nesse cálculo. Para que se tenha uma precisão de 99% de chance de não ocorrer erros, a correlação será alta se for maior que 0,959. Portanto, como praticamente todos os índices de correlação entre os dados obtidos de processos diferentes foram menores que 0,878 (com uma exigência de exatidão menor) e menores que 0,959 (com uma exigência de exatidão maior), consideramos baixos e portanto com confiabilidade duvidosa.

Por outro lado, analisando um dos índices de correlação, entre os dados da Medição 2 (Processo A) e 3 (Processo B), o qual foi calculado em $r = 0,8188$ é possível identificar um outro tipo de comportamento dos dados quando o processo de coleta sofre alteração. Diferente do que aconteceu na Avaliação 1 ($r = 0,4552$), na Avaliação 2 os índices com dados de processos diferentes não foram tão baixos, o que indica que quando há modificação no processo de coleta, a qualidade do dado referente à confiabilidade pode até ser alta, mas este é um fator que pode apresentar aleatoriedade. Mesmo com este novo fator identificado no comportamento dos dados dessa métrica no projeto P1, não é possível inferir que os dados mantém sua qualidade preservada quando coletados de maneiras diferentes.

3.3 Considerações sobre o estudo realizado

Com a realização deste estudo apresentado no Trabalho Individual II [Berardi, 2007] evidenciou-se que mudanças no processo de coleta de dados de métricas da organização estão, em uma primeira análise refletindo negativamente na qualidade dos dados. Com isso, mostrou-se que realmente os dados dessa base podem ter sua qualidade duvidosa, pois a qualidade do dado está sendo sensível a fatores externos. Além disso, observou-se os dados armazenados no repositório central e inferiu-se que, em coletas do passado, existem distorções em algum destes quando comparados os valores da métrica na base fonte e no repositório central. Tendo em vista esta última consideração, identificou-se que no processo de coleta manual desses dados, são adicionados ruídos nos mesmos. Esses ruídos estão na forma de inconsistências, incompletudes e desatualizações.

Com a execução dessa avaliação preliminar pode-se concluir alguns aspectos com relação à qualidade dos dados. No entanto, a forma de avaliação utilizada tem característica estatística e genérica, não considera aspectos inerentes à medições que são de software e, além disso, não aponta potenciais causas para o problema encontrado de homogeneidade dos dados. Essa técnica estatística é eficiente para confirmar ou descartar hipóteses de problemas. No entanto, ela só aponta que pode existir o problema, mas não é capaz de mostrar onde e por quê ele ocorre.

Para a análise da qualidade de uma métrica igualmente importante, como esforço de trabalho, que é a medição do quanto está sendo investido para a construção de um produto de software novo, é essencial a consideração de vários aspectos e não somente estatísticos, sendo necessário então um outro mecanismo de avaliação que não apresente essas falhas comentadas. A seguir são mencionados e discutidos os principais aspectos para construir uma proposta de avaliação eficiente no que diz respeito a contexto e apontamento de causas de problemas de qualidade, quando evidenciados.

3.4 Requisitos da solução

Para sistematizar a busca por uma solução sobre a avaliação de qualidade de dados de métricas de esforço de software, optou-se por esquematizar alguns aspectos envolvidos nessa tarefa, os quais foram notados no ambiente real durante a avaliação descrita anteriormente, visto que alguns gestores participaram ativamente desse processo, por demonstrarem total interesse em avaliar a qualidade dos dados dos seus projetos. Esses aspectos envolvidos

são: 1) Granularidade, 2) Funcionalidade, 3) Não-dicotomicidade, 4) Objetividade e 5) Contextualização.

1. Granularidade: A solução deve ser capaz de prover uma avaliação da qualidade para cada lançamento realizado, pois esse nível de detalhe proporciona melhor identificação de problemas e, conseqüentemente, de melhorias que podem ser realizadas. Uma avaliação que aponta a qualidade de conjuntos de dados mas não aponta claramente quais os motivos para tal avaliação não é eficaz neste contexto. Além disso, não apontar quais dados oriundos de quais membros estão com problemas também é uma falha no processo de avaliação. Atualmente o processo de melhoria e conscientização é geral envolvendo todos os colaboradores da equipe por não se saber um ponto de foco, onerando assim membros que poderiam estar se dedicando às suas atividades.
2. Funcionalidade: A solução deve permitir praticidade na análise dos dados possibilitando rápido retorno aos usuários quanto à avaliação da qualidade dos mesmos. Um resultado de avaliação disponibilizado depois de um certo tempo pode não ser mais útil para o gestor e dificulta a tomada de ação para busca de melhorias. Atualmente, a avaliação é realizada no momento da coleta de dados pelos próprios gestores e quando algum dado é suspeito (baseado em simples suspeitas e suposições) é exigido um prazo de no máximo 24 horas para se confirmar ou descartar as suspeitas, apontando assim uma certa urgência no conhecimento sobre o estado dos dados.
3. Não-dicotomicidade: A solução deve ser capaz de evitar a dicotomicidade, onde uma classificação exclui diretamente outra. Afirmar que um dado é estritamente ruim pode estar sendo desconsiderado o fato de que ele é pouco ruim, por exemplo. A dicotomicidade inibe a observação da evolução da qualidade dos dados, uma vez que esse tipo de avaliação não permite se ter a noção de que um dado era ruim e que após certas ações ele está menos ruim. Atualmente no ambiente real, há uma demanda por parte dos gestores de equipes de desenvolvimento de software pelo acompanhamento da evolução de qualidade de dados.
4. Objetividade: a solução deve apontar os motivos que levaram o dado a obter determinada avaliação. Essa é uma característica crítica para o processo de melhoria da qualidade, pois é o que permite verificar o que faz com que o dado tenha tal avaliação. A gestão de equipe de software exige dinamicidade e praticidade uma vez que

o contato com o cliente é permanente e direto. Indicar que há um problema se torna um problema se as causas não são apontadas. Os gestores necessitam da visibilidade do seu foco de ação e assim acompanhar a evolução nesses aspectos, sem onerá-los com ações genéricas sem foco e com possibilidade de falha no sucesso.

5. Contextualização: a solução deve considerar as características inerentes ao ambiente de PDS, o processo de coleta desses dados e o processo de análise dos mesmos. A coleta de métricas é um dos aspectos específicos desse contexto por não possuir um mecanismo automático de medição, o que torna esse ambiente mais complexo e não trivial.

Todos os requisitos são de grande e igual importância para que o problema seja resolvido da melhor forma possível. Portanto uma solução não é aplicável caso não abranja todos os critérios. É fato que a definição desses critérios é baseada na análise de um único ambiente real de desenvolvimento de software, colocando em dúvida se em outros ambientes reais desse tipo seriam notados os mesmos requisitos. No entanto, acreditamos que na ausência de critérios (pois nenhum outro trabalho aborda esse problema dessa maneira), a definição de um conjunto de requisitos, mesmo que pequeno e restrito a um cenário, é benéfica pois permite direcionar os objetivos do trabalho.

Na observação de uma proposta quanto a esses requisitos é possível inferir que o requisito é totalmente atendido pela proposta, parcialmente ou definitivamente não atendido. A proposta atenderá totalmente o requisito se contém todas as características descritas do respectivo requisito e poderá atender parcialmente quando apenas algumas dessas características são atendidas e não atenderá se nenhuma das características descritas no requisito for observada.

4

Trabalhos relacionados

Este capítulo apresenta os trabalhos relacionados ao tema de pesquisa encontrados na literatura que propõem soluções para avaliar qualidade de dados em diversos contextos, dentre eles estão: AIQM (*A methodology for information quality assessment*) [Lee et al., 2002], ASM-IQS (*Assessment of software measurement an information quality study*) [Berry et al., 2004], DQA (*Data quality assessment*) [Francalanci and Pernici, 2004], DQMIM (*A data quality measurement information model*) [Caballero et al., 2007], ORME-DQ (*A framework and a methodology for data quality assessment and monitoring*) [Batini et al., 2007], MDB:PA (*Measuring data believability:a provenance approach*) [Nicolas Prat, 2008] e PEDI (*Process-embedded data integrity*) [Lee et al., 2004]. O estudo foi feito em duas etapas devido à possibilidade de classificação dos trabalhos em dois grupos distintos e ao final de cada etapa é apresentada uma tabela de comparação desses trabalhos observando a abrangência ou não deles com relação aos requisitos da solução apontados no Capítulo anterior.

4.1 Análise

De acordo com Lee et. al. em [Lee et al., 2006], para a avaliação de qualidade de dados existem diversas técnicas, onde as principais são: (a) pesquisa sobre a qualidade dos dados (*data quality survey*); (b) aplicação de métricas (*data quality metrics*) e (c) análise de integridade de dados em bancos de dados (*data integrity analysis*). Os grupos são os seguintes:

Primeiro grupo: Análise dos trabalhos que aplicam as técnicas apontadas como principais por Lee et. al. em [Lee et al., 2006]. Esse grupo é analisado focando a abrangência

(forte, fraca ou nenhuma) de todos os requisitos apontados para uma solução adequada, observando sua total utilização como solução para o problema.

Segundo grupo: Análise dos trabalhos que aplicam técnicas distintas às apontadas por Lee et. al. em [Lee et al., 2006]. Esse grupo é analisado focando a abrangência total do maior número de requisitos possível, observando sua adequação enquanto abordagem para a elaboração de uma solução para o problema.

4.1.1 Primeiro grupo

Trabalhos que aplicam *Data Quality Survey* são:

- Lee et. al. (AIMQ) em [Lee et al., 2002] propõem uma metodologia que se utiliza de questionários para avaliar a qualidade de dados em qualquer contexto. Essa metodologia é composta por três componentes: (1) definição das dimensões para avaliação (2) coleta de dados através de questionários e (3) técnicas de análise para identificar problemas de qualidade nas informações;
- Berry et. al. (ASM-IQS) em [Berry et al., 2004] propõem a aplicação de dois diferentes tipos de instrumentos de avaliação, onde um é um questionário genérico e o outro é um questionário voltado para dados de medição de software. O foco dessa avaliação é para melhoria do processo de medição e não para melhoria dos dados propriamente ditos.

Os trabalhos que aplicam a segunda técnica *Data Quality Metrics* como forma de avaliação são:

- Pipino et. al. (DQA) em [Francalanci and Pernici, 2004] propõem um conjunto de métricas baseadas nas dimensões de [Wand and Wang, 1996] formuladas através de três tipos de cálculos (razão, min./max. e média ponderada) que indicam a qualidade de um conjunto de dados genéricos. Os autores ainda apresentam como essas métricas podem ser combinadas a uma avaliação subjetiva para melhorar a qualidade dos dados;

- Caballero et. al. (DQMIM) em [Caballero et al., 2007] definem um modelo de medição que a partir das necessidades dos usuários define o que, como e por que deve ser medido e ainda, quem realiza a medição. Com base nessas definições é construído um conjunto de métricas para avaliar a qualidade de conjunto de dados genéricos;
- Batini et. al. (ORME-DQ) em [Batini et al., 2007] propõem uma metodologia para avaliação, composta de quatro fases (1) análise do ambiente, (2) seleção de conjunto de dados críticos, (3) avaliação qualitativa e quantitativa da qualidade e (4) avaliação constante de qualidade. Na fase 3 é onde são aplicadas as métricas, cujo formato depende da respectiva dimensão (simples comparação entre strings, cálculo de distância, etc.);
- Prat e Madnick (MDB: PA) em [Nicolas Prat, 2008] propõem uma abordagem para medir a confiabilidade de um conjunto de dados. Os autores definem um conjunto de métricas a ser aplicadas com dados armazenados em um modelo de Banco de Dados que guarda a proveniência dos dados. Tanto as métricas quanto o modelo de proveniência são genéricos.

Os trabalhos que aplicam a segunda técnica *Data Integrity Analysis* como forma de avaliação são:

A pesquisa de Lee et. al. (PEDI) em [Lee et al., 2004] propõe um processo iterativo de melhoria de qualidade através da adoção de regras de integridade onde, dependendo da regra, algumas dimensões de [Wand and Wang, 1996] são atacadas:

- integridade de coluna: acurácia e interpretabilidade,
- integridade de entidade: acurácia e completude,
- integridade de usuário: consistência

Constantemente as regras são avaliadas através de freqüentes definições, medições, análises e redefinições para um ciclo de melhorias.

4.1.1.1 Considerações sobre trabalhos do primeiro grupo

A análise dos trabalhos especificamente com relação à abrangência dos requisitos da solução buscada (Capítulo 3) está ilustrada na Tabela 4.1. Células com preenchimento de

cor mais escura significam que o trabalho abrange o requisito conforme desejado para a solução, a cor mais clara indica que o trabalho apresenta o requisito, mas não conforme o esperado para a solução e a ausência de preenchimento indica a não abrangência ao requisito.

Tabela 4.1 – Trabalhos Relacionados

	Data Quality Survey		Data Quality Metrics				Integrity Data Analysis
Trabalho / Requisito	AIQM (2002)	ASM-IQS (2004)	DQA (2002)	DQMI M (2007)	ORM E-DQ (2007)	MDB:PA (2008)	PEDI (2004)
Granularidade							
Funcionalidade							
Não-dicotomicidade							
Objetividade							
Contextualização							

Não foi encontrado trabalho que, de acordo com os requisitos selecionados, pudesse ser aplicado adequadamente na avaliação de qualidade de dados no contexto de métricas de esforço de software. Nesse sentido, é possível observar com essa análise que, para o problema abordado neste trabalho, a técnica de *survey* (*data quality survey*) não é suficientemente completa para a solução, pois apesar de abranger alguns requisitos, a forma de aplicação e de análise de resultados dificulta consideravelmente sua aplicação. O tempo despendido na coleta de respostas sobre a qualidade de dados prejudica a sua **funcionalidade**, assim como o tempo necessário para a análise dos resultados obtidos com as respostas. Apesar dos trabalhos referentes à técnica que se utiliza de métricas (*data quality metrics*) serem os que apresentam as maiores funcionalidades, o problema de **granularidade** dificulta sua utilização para a solução aqui buscada. As métricas geralmente apresentam uma avaliação **dicotômica** (bom/ruim) levando em consideração grupos de dados (por exemplo, 70% do conjunto de dados X apresenta problema). Já o trabalho de análise de integridade de dados (*data integrity analysis*) não apresenta grande aderência ao que se procura, apesar de utilizar uma granularidade mais detalhada (por exemplo, colunas em uma tabela de banco de dados). O mesmo não satisfaz as reais necessidades da solução uma vez que o alto detalhismo desconsidera outros fatores que podem afetar a qualidade dos dados no contexto estudado, pois de acordo com os próprios autores [Lee et al., 2004], a análise de integridade de coluna, por exemplo, não tem como garantir que os dados presentes são corretos.

4.1.2 Segundo grupo

O trabalho desenvolvido por Liebchen et. al. em [Liebchen et al., 2007] é utilizada a técnica de classificação para avaliar a qualidade dos dados. A falta de adequação dessa técnica para o atual problema está relacionada às manipulações que os dados sofrem na preparação prévia à aplicação do algoritmo de classificação. Não parece coerente avaliar dados que têm a qualidade afetada justamente por manipulações ao longo do seu ciclo, através de uma técnica que também está sujeita à inserção de erros nos dados. Além disso, mesmo que a preparação seja isenta de problemas, uma avaliação que desconsidere características de **contexto** (talvez subjetivas), muitas vezes não mensuráveis, também pode apresentar falhas.

A pesquisa desenvolvida por Caro [Angelica Caro, 2007] considera o conhecimento empírico de usuários de portais web para identificar critérios para avaliação de qualidade de dados obtidos através desses portais. Esse trabalho utiliza critérios sugeridos pelos usuários para modelar uma rede *Bayesiana* considerando a incerteza dos usuários no momento da avaliação, o que pode ser uma abordagem interessante. No entanto, a rede *Bayesiana* aponta uma probabilidade de o dado ser bom ou ruim, mas não aponta quão bom ou ruim o dado pode ser considerado.

Data Provenance é uma alternativa que tem sido utilizada na busca pela qualidade de dados, onde tem se defendido que, através do conhecimento do processo de obtenção dos dados e/ou de suas fontes, é possível inferir se o dado tem ou não qualidade [Buneman et al., 2001]. A abordagem se mostra interessante para ajudar a evidenciar problemas de qualidade de dados através da sua proveniência, e pode ser encontrada em diferentes domínios de aplicação. Em Chimera [Foster et al., 2002] é utilizado *Data Provenance* para aplicações nas áreas de Astronomia, Biologia e Física. Buneman [Buneman et al., 2001] também a adota com dados da Biologia. Em Poesia [Fileto et al., 2003], por sua vez, é aplicada tal abordagem no contexto de Agricultura, enquanto que, em Trio [Widom, 2005], no de Biologia. Até onde se sabe, não foi utilizado *Data Provenance* no contexto de métricas de software.

A análise dos trabalhos do segundo grupo com relação à abrangência dos requisitos da solução buscada (Capítulo 3) está ilustrada na Tabela 4.2. O preenchimento de cor mais escura significa que o trabalho abrange o requisito conforme desejado para a solução, a cor mais clara indica que o trabalho apresenta o requisito, mas não conforme o esperado para a solução e a ausência de preenchimento indica a não abrangência ao requisito.

Tabela 4.2 – Trabalhos relacionados - segundo grupo

Requisito \ Trabalho	Liebchen (2007)	Caro (2007)	Buneman (2006)
Granularidade			
Funcionalidade			
Não-dicotomicidade			
Objetividade			
Contextualização			

4.1.2.1 Considerações sobre trabalhos do segundo grupo

Fica claro na Tabela 4.2 que o trabalho de Buneman é o que mais atende os requisitos. Contudo, como discutido no Capítulo 3, tal trabalho tem pouca aplicabilidade por não atender os requisitos de **Não-dicotomicidade** e **Contextualização**. A propósito, o critério de **não-dicotomicidade** não é atendido por nenhum dos trabalhos analisados, o que caracteriza uma proposta inovadora e desafiadora da presente pesquisa. As questões de objetividade e contextualização também são críticas porque apenas um de cada três trabalhos preencheu o requisito.

4.2 Considerações sobre o Capítulo

Tendo identificado através dos trabalhos relacionados que o tema é promissor e não está totalmente resolvido, definiu-se a estratégia de pesquisa. De acordo com as características observadas, a pesquisa a ser desenvolvida é classificada como exploratória, onde o objetivo é examinar um tema ou problema de pesquisa pouco estudado ou que não tenha sido abordado da mesma forma anteriormente na literatura. Entende-se que essa estratégia é adequada, pois o levantamento bibliográfico mostrou que não existem propostas abrangentes na área, e os estudos realizados até o momento não aprofundam o problema de estudo.

5

Solução

Este capítulo descreve a solução proposta para avaliar a qualidade de dados de métrica de esforço de software. É uma arquitetura que comporta o rastreamento e armazenamento da proveniência dos dados de métricas de esforço e através dessa proveniência avalia por um sistema de inferência qual o nível de qualidade do dado variando de 0 a 1. Nesse sentido, as próximas seções contemplam: (i) descrição da arquitetura proposta, (ii) o processo de avaliação suportado pela arquitetura e (iii) descrição dos componentes que compõem a arquitetura juntamente com o processo de criação dos mesmos.

5.1 Arquitetura proposta

A arquitetura proposta para suportar a avaliação da qualidade dos dados de esforço, ilustrada na Figura 5.1 é composta por quatro principais blocos componentes de ação: (a) Componente de Proveniência, (b) Componente de Inferência, (c) Componente de Qualidade e (d) Componente de Análise.

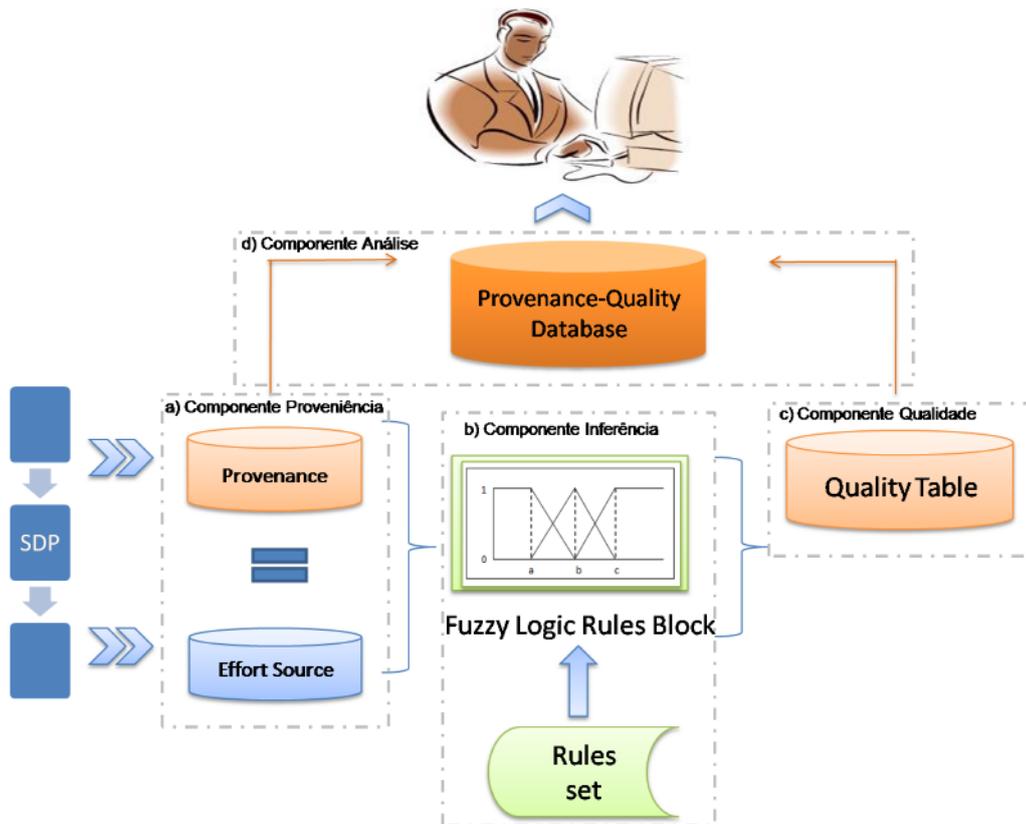


Figura 5.1 – Solução proposta

5.2 Processo de avaliação

O processo de avaliação que a arquitetura suporta está baseado no registro da métrica de esforço de trabalho, que é medida em horas, e ocorre durante o processo de desenvolvimento de software. Tomando como referência a Figura 5.1, o andamento do PDS está representado pelas caixas azuis da esquerda para direita e a base de dados chamada *Effort Source* é o banco de dados que armazena as horas trabalhadas de cada membro da equipe. Ao final de cada dia de trabalho o membro da equipe (podendo este ser gestor/desenvolvedor/analista/projetista, ou outro) registra seu esforço para cada atividade desenvolvida e paralelamente a esse registro ocorre o rastreamento da proveniência (representada na Figura 5.1 pelas barras paralelas entre a base *Effort Source* e *Provenance*). A base de dados chamada *Provenance* registra essa proveniência rastreada. A etapa seguinte do processo é quando um colaborador faz uma coleta de registros de horas referente a um período de interesse (podendo ser semanal/quinzenal/mensal, ou outro) para disponibilizar aos gestores para algum processo de tomada de decisão (estima-

tiva/análise/monitoração, ou outro). Antes que esses dados sejam mostrados aos gestores, o colaborador os submete a uma máquina de inferência (representada pela Figura 5.1) para receberem uma avaliação de nível de qualidade, cujo resultado é armazenado em um base de dados chamada *Quality Table*. Após, os dados podem ser disponibilizados ao processo de tomada de decisão com o nível de qualidade atribuído. A próxima e última etapa desse processo é a carga histórica da qualidade desses dados, que ocorre em um banco dimensional estruturado na forma de um Data Warehouse que tem as bases *Provenance*, *Effort Source* e *Quality Table* como fonte desses dados (Figura 5.1 d).

O processo acima descrito é considerado bem comportado, podendo variar alguns aspectos principalmente no que diz respeito à periodicidade com que os membros registram o esforço. No entanto, a arquitetura e seus componentes comportam as diferentes possibilidades de mudança do processo nesse sentido. A seguir cada componente da arquitetura é descrito juntamente com o seu processo de construção.

5.3 Componentes da arquitetura

A composição dos quatro blocos é o que permite à organização ter conhecimento do que está acontecendo no presente, mostrando os pontos de falha e de investimento de mudança para melhorias. Cada bloco tem sua importância na arquitetura, sendo que um bloco sozinho não traz tantos benefícios quanto o uso em conjunto dos mesmos.

5.3.1 Componente Proveniência

O bloco do Componente Proveniência, (a) na Figura 5.1, consiste no armazenamento da rastreabilidade de dados que dizem respeito às circunstâncias atuais de registro na base de dados *Effort Source*. Como foi discutido anteriormente (Subseção 4.2, Capítulo 4) existem várias técnicas para fazer uso de *Data Provenance (DP)*. Neste trabalho, o uso de DP foi através da construção de uma base de dados auxiliar convenientemente modelada e estruturada para capturar todas as informações consideradas pertinentes no processo de obtenção e derivação dos dados de esforço. A Figura 5.2 mostra o diagrama de classes UML do banco de dados desse componente de proveniência. As classes *Event*, *Project*, *Release*, *Phase*, *Resource* têm a função de registrar a proveniência do dado originalmente lançado, armazenando seu valor inicial (*Effort Value*), o dia em que o lançamento está

ocorrendo (*CurrentDay*) e o dia em que está sendo registrado o esforço (*RegisterDay*). A classe *UpdateEvent* tem a função de registrar a proveniência do dado quando este está sendo modificado por atualizações ou transformações, armazenando seu novo valor (*NewEffortValue*), quando isso está ocorrendo (*UpdateDay*) e quem está realizando essa alteração (*Id-custodian*). E a classe *Analysis* tem a função de registrar as datas de análises que os dados sofrem para observar eventuais mudanças de qualidade em um mesmo registro.

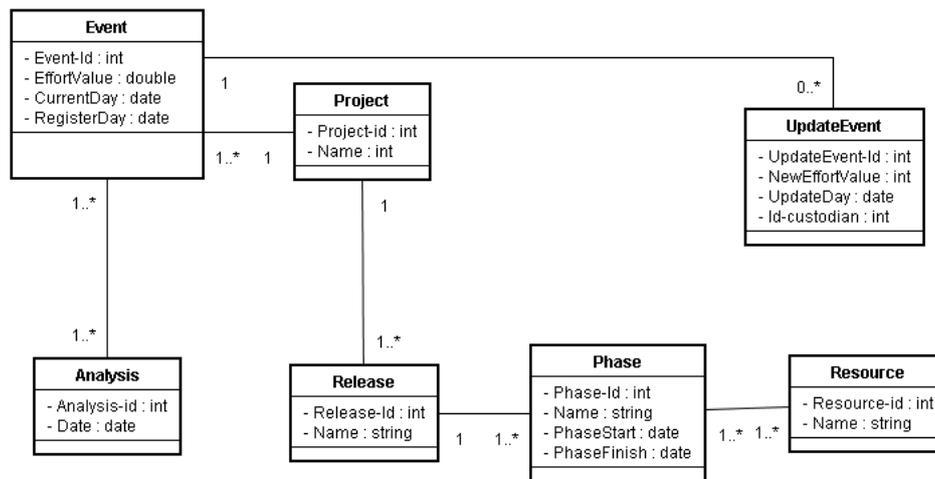


Figura 5.2 – Diagrama de classes UML do Componente de Proveniência

Para a construção desse modelo foram utilizados conceitos de DP e também conceitos consolidados na área de *Data Quality*, como o uso de dimensões de qualidade. A proposta deste trabalho é inovadora no sentido em que utiliza DP como variáveis na composição de dimensões de qualidade discutidas anteriormente na Fundamentação Teórica através das quais o dado é avaliado. A seguir são expostas as dimensões consideradas nesta arquitetura e como foi o processo de composição das mesmas através de DP.

5.3.1.1 Construção do componente através de dimensões de qualidade

A avaliação de qualidade de dados é composta pela avaliação de diversas dimensões e cabe a cada organização determinar quais dimensões são mais importantes de acordo com o contexto em que os dados estão inseridos. Além disso, deve-se também definir quais variáveis estão envolvidas na composição de cada dimensão [Lee et al., 2006]. Um aspecto novo deste trabalho é justamente compor cada dimensão com informações referentes à proveniência dos dados avaliados, pois acreditamos que essa composição já exista e é con-

hecida e utilizada no dia-a-dia das organizações, só que de forma empírica e intuitiva. Além disso, apesar de atualmente a avaliação ser subjetiva, os envolvidos se utilizam de critérios e também relacionam variáveis de acordo com esses critérios para julgar os dados que lhes são disponibilizados. Assim, para fazer essa composição é necessário que se consiga captar o conhecimento subjetivo e inerente às pessoas, e transformar em um conhecimento de posse da organização adequadamente estruturado e acessível para realização de avaliações de qualidade independente de quem está envolvido no processo.

Para isso, identificamos a necessidade de estudo e análise de um ambiente real de desenvolvimento de software, para se identificar quais dimensões e variáveis podem ser consideradas essenciais em um ambiente de desenvolvimento de software. Esse estudo e análise permitem que os componentes sejam construídos da melhor forma possível, refletindo as necessidades mais evidentes em um ambiente desse tipo.

É importante salientar que as dimensões refletem as necessidades dos envolvidos com a qualidade dos dados e, como não é possível abranger todas as necessidades, é preciso fazer escolhas [Olson, 2003]. Dessa forma, para a identificação de quais dimensões podem ser consideradas essenciais no ambiente de desenvolvimento de software, foram utilizados resultados da análise estatística de qualidade nos dados da operação de software parceira demonstrada no Capítulo 3, onde foram identificados problemas críticos durante o ciclo dos dados de esforço. Essa análise nos permitiu a definição de quais dimensões poderiam ser utilizadas dentre o conjunto proposto por [Wand and Wang, 1996]. Este trabalho tem o enfoque nas dimensões relativas a diferentes categoria estudadas no Capítulo 2, tais como *Intrinsic DQ* e *Contextual DQ*, pois identificamos que as questões abordadas nestas categorias vão ao encontro das necessidades de um dado de métrica de software, que precisa ter objetividade, confiabilidade e excelente precisão e reputação. A tabela 5.1 apresenta as dimensões definidas. A descrição de cada dimensão está relacionada ao contexto em que ela é aplicada. No contexto de análise de dados de esforço de software, as dimensões tem as seguintes interpretações:

Tabela 5.1 – *Dimensões de qualidade*

Dimensão	Descrição
Acurácia	Diz respeito ao quanto o dado está correto. Essa dimensão avalia se o dado contém o valor mais próximo do esperado possível.
Atualidade	Diz respeito a quão atualizado está o dado para o que ele vai ser analisado. Essa dimensão pode ter duas interpretações: 1) o dado não está atual porque o colaborador não o registrou no período que devia (está atrasado) ou 2) o dado não está atual porque ele foi coletado em um período e a análise está sendo feita muito tempo depois desse período. A interpretação adotada pela presente solução está relacionada à primeira (1).
Completude	Diz respeito à falta de informação, valores faltantes. Essa dimensão avalia se o dado não está completo devido à falta de registro ou o registro existente contém falhas com relação ao esperado.
Consistência	Diz respeito ao sentido do dado ao longo do ciclo. Essa dimensão avalia se o dado sofreu inserção de erro ao sair de sua origem (base fonte), passar por todo o processo de extração e transformação e não sofrer alteração (modificação) significativa em seu valor.

A proposta deste trabalho é que cada dimensão seja constituída de variáveis referentes à proveniência dos dados, isso porque se acredita que rastreando o dado de esforço durante todo seu ciclo é possível inferir níveis de qualidade. Para isso, a imersão em um ambiente real se faz, novamente, necessária e totalmente pertinente para a construção de componentes que refletem a realidade do contexto. Entretanto, para a definição das variáveis, não basta realizar análises sobre problemas históricos, como foi feito na definição das dimensões, já que esse é um conhecimento empírico baseado em conhecimentos e experiências acumuladas na organização. Portanto, nessa etapa foi necessário definir uma metodologia de pesquisa para aquisição de conhecimento.

5.3.1.2 Aquisição de conhecimento

Para a aquisição de conhecimento foi escolhida a metodologia de entrevista semi-estruturada (que pode ser acessada no Apêndice B através do Projeto exposto), uma vez que a metodolo-

gia que se utiliza de questionário não seria capaz de capturar detalhes necessários para a composição dos componentes da arquitetura [Rezende, 2003]. A estrutura da entrevista facilita a aquisição, pois a presença do entrevistador possibilita que novas questões sejam formuladas como conseqüência de uma resposta fornecida. O instrumento desenvolvido para apoio à entrevista (apresentado no Apêndice A) possui uma organização visando à construção dos componentes da arquitetura. Neste momento trataremos do aspecto da entrevista relacionado à construção do Componente de Proveniência, sendo os outros aspectos tratados na descrição de cada componente correspondente.

A parte da entrevista que contribuiu para a modelagem do Componente de Proveniência está ilustrada na Figura 5.3, onde foi disponibilizada uma lista de variáveis (informações de DP possíveis de serem rastreadas) para que o entrevistado selecionasse e relacionasse essas variáveis a cada uma das dimensões (acurácia, atualidade, completude e consistência).

- Acurácia**
- 1() o valor original lançado
 - 2() o dia do registro (dia escolhido para lançar a hora)
 - 3() dia do lançamento (dia atual)
 - 4() fase de lançamento
 - 5() fonte de lançamento (ferramentas de gerência)
 - 6() valor modificado (por transformações, adaptações, correções)
 - 7() diferença do valor original com um valor esperado (discrepância / quanto?)
 - 8() diferença do valor original com o valor modificado (discrepância)
 - 9() experiência
 - 10() _____

Figura 5.3 – Lista de variáveis de DP na entrevista

Para construir o modelo apresentado na Figura 5.2, foram utilizadas as variáveis que tiveram maior número de seleção por parte dos entrevistados e também variáveis que não estavam na lista inicialmente e que os entrevistados adicionaram. A Tabela 5.2 ilustra as variáveis envolvidas para cada dimensão de acordo com o que foi coletado com os entrevistados, onde :

- #DR-DL é a diferença entre o dia do registro (DR) e o dia do lançamento (DL), onde o dia do registro é o dia para o qual o colaborador quer lançar o esforço e o dia do lançamento é a data em que ele está fazendo isto;
- #DFD-DR é a diferença entre o dia em que finalizou a fase de lançamento (DFD) e o dia do registro (DR);

- #VO-VM é a diferença entre o valor original (VO) que é o primeiro registro do esforço e o valor modificado (VM) caso o registro seja alterado por motivo de atualização ou engano;
- #DC-DA é a diferença entre o dia da coleta dos dados (DC) e o dia da análise (DA).

Tabela 5.2 – Variáveis de DP para cada dimensão

Acurácia	Atualidade	Completude	Consistência
#DR-DL	#DR-DL	#DR-DL	#VO-VM
#DFF-DR	#DC-DA	#DFF-DR	
#VO-VM			

No momento, a proveniência contemplada pelo modelo diz respeito a circunstâncias de lançamento e coleta dos dados. Outros tipos de proveniência (fonte/cálculos) não são abordados porque a pesquisa está focada em dados de esforço que não são obtidos por mais de uma fonte e não sofrem cálculos.

5.3.1.3 Sobre o processo empregado nas entrevistas

Conforme indica a metodologia de aplicação de entrevistas estruturadas, antes da realização oficial das mesmas com os colaboradores em um ambiente real, foi realizado um pré-teste, cujos entrevistados são membros do grupo de pesquisa que possuem vivência na operação de software parceira e portanto conhecem a realidade dos reais entrevistados. Além disso, para o pré-teste foram escolhidos esses membros por terem um certo conhecimento do contexto de dados de esforço podendo assim colaborar com a melhoria do instrumento de auxílio à aquisição de conhecimento e com o próprio processo de condução da entrevista.

Após a aplicação do pré-teste, foram convidados 10 colaboradores da operação de software, com 3 diferentes perfis: membro de equipe de desenvolvimento, gestor e organizador dos dados, conforme indica [Lee et al., 2006]. Com a participação de diferentes perfis na aquisição de conhecimento é possível entender melhor os problemas relacionados a dados de esforço considerando diferentes tipos de papéis e responsabilidades com relação aos dados. Além disso, a abrangência de diferentes papéis é um fator importante na conscientização da organização como um todo quanto à importância da qualidade dos dados [Lee et al., 2006].

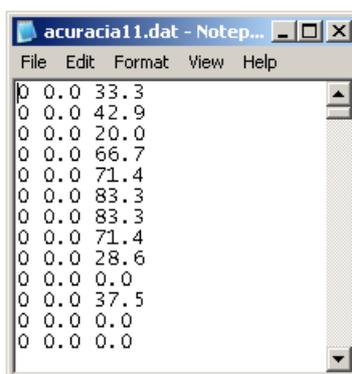
Dentre esses 10 participantes, 4 são gestores, 4 são membros de equipe de desenvolvimento e 2 são organizadores de dados para análise. Acreditamos que apesar de não ser

um número muito alto de entrevistados, para o tipo de uso de respostas a amostra é satisfatória. O propósito não é realizar análises estatísticas, mas sim construir uma agregação de conhecimentos de acordo com diferentes pontos de vista para alcançar uma adequada abrangência de critérios a serem considerados na arquitetura.

Nesse sentido, a escolha dos entrevistados foi rigorosa e não aleatória, buscando colaboradores que possuíssem alto grau de experiência em PDS, alto nível de envolvimento e comprometimento com dados de esforço e alto grau de responsabilidade. Detectar as necessidades dos envolvidos é uma tarefa muito difícil, mas é a melhor forma de entender o domínio e traz resultados bem interessantes de acordo com [Society, 1998].

5.3.2 Componente de Inferência

O bloco do Componente de Inferência, (b) na Figura 5.1, apresenta uma máquina de inferência munida de uma base de regras previamente construída. Com a frequência definida pela organização (semanal/ quinzenal/ mensal), a máquina de inferência recebe como entrada os dados de proveniência, referentes ao período desejado e, através do conjunto de regras, avalia a qualidade de cada lançamento considerando-o desde seu primeiro registro e todas as modificações sofridas durante o período (se existirem) até a última alteração. A Figura 5.4 mostra um exemplo de arquivo de entrada com os dados de proveniência para o processamento da máquina de inferência para a dimensão acurácia onde o arquivo mostra para cada registro (representado por cada linha) na primeira coluna os valores da variável #DR-DL, na segunda coluna os valores da variável #DFF-DR e na terceira coluna os valores da variável #VO-VM.



#DR-DL	#DFF-DR	#VO-VM
0	0.0	33.3
0	0.0	42.9
0	0.0	20.0
0	0.0	66.7
0	0.0	71.4
0	0.0	83.3
0	0.0	83.3
0	0.0	71.4
0	0.0	28.6
0	0.0	0.0
0	0.0	37.5
0	0.0	0.0
0	0.0	0.0

Figura 5.4 – Exemplo de entrada do Componente de Inferência

A máquina possui dois níveis de processamento, onde no primeiro (*1st Inference Level result* na Figura 5.5) ela processa, através das regras e da função de pertinência, o grau de aderência do dado com relação a cada dimensão, cujos valores estão no intervalo [0,1] de acordo com a lógica fuzzy. O segundo nível de processamento (*2nd Inference Level result* na Figura 5.5) utiliza o resultado de grau de aderência calculado para cada dimensão (*Quality fuzzy rule base* na Figura 5.5) para calcular o grau de aderência do dado a diferentes graus de qualidade (*Low, Reasonable, Acceptable ou High*), conforme ilustrado na Fig.5.5.

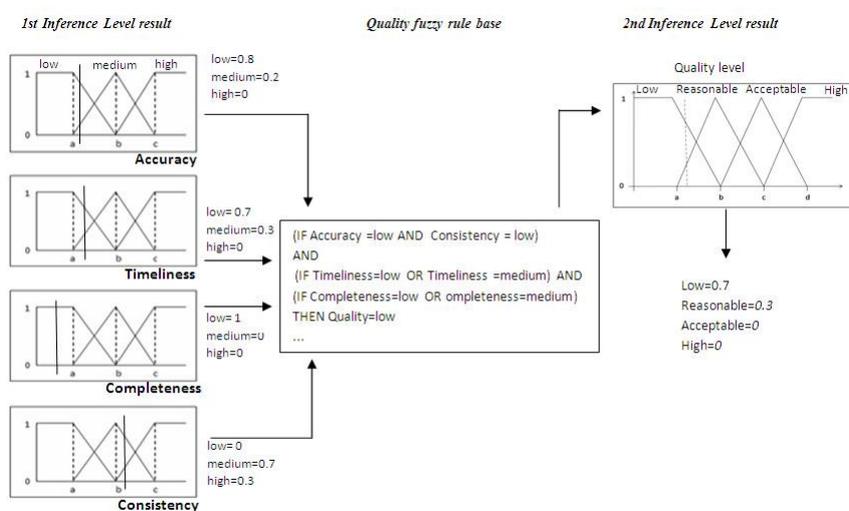


Figura 5.5 – Componente de Inferência Fuzzy

A Figura 5.6 mostra um exemplo de resultado de avaliação com o processamento da máquina de inferência onde cada linha (identificado por ID) representa um lançamento de esforço, a coluna grupo mostra a fase de lançamento, a coluna chv_ati é a atividade correspondente ao esforço mostrado na coluna hora_at. O restante das colunas diz respeito às avaliações, já defuzzyficadas, com as respectivas variáveis e o respectivo resultado, sendo diferenciadas pelas cores. A última coluna (quali) apresenta o resultado de avaliação final defuzzyficado. Para conhecer a avaliação linguística a Figura 5.37 pode ser acessada na seção seguinte.

ID	grupo	chv_ati	complemen	hora_at	ACURÁCIA			ATUALIDADE			COMPLETUDE			CONSISTÊNCIA			quali	
					#DR-DL	#DFF-D	#VO-VI	Acur	#DR-DL	#DC-DA	Atu	#DR-DL	#DFF-D	Com	#VO-VM	Con		
1	PROJETO	85066	OS-1520		0,83	0	0,0	0,0	8,31	0	2	8,31	0	0,0	8,31	0,0	8,31	7,3
2	PROJETO	85263	OS-1520		1,83	0	0,0	0,0	8,31	0	2	8,31	0	0,0	8,31	0,0	8,31	7,3
3	PROJETO	85265	OS-1520		2,17	0	0,0	0,0	8,31	0	2	8,31	0	0,0	8,31	0,0	8,31	7,3
4	PROJETO	86337	OS-1520		1,00	0	0,0	0,0	8,31	0	2	8,31	0	0,0	8,31	0,0	8,31	7,3
5	PROJETO	86341	OS-1520		4,00	0	0,0	33,3	6,86	0	2	8,31	0	0,0	8,31	33,3	1,74	6,6
6	PROJETO	86390	OS-1520		1,67	0	0,0	0,0	8,31	0	2	8,31	0	0,0	8,31	0,0	8,31	7,3
7	PROJETO	86391	OS-1520		5,00	0	0,0	0,0	8,31	0	2	8,31	0	0,0	8,31	0,0	8,31	7,3
8	PROJETO	86503	OS-1520		0,50	0	0,0	84,2	5,15	0	2	8,31	0	0,0	8,31	84,2	1,37	5,4
9	SERVER	87060	OS-1520		1,00	0	0,0	66,7	5,15	0	2	8,31	0	0,0	8,31	66,7	1,37	5,4
10	SERVER	87119	OS-1520		1,07	0	0,0	73,3	5,15	0	2	8,31	0	0,0	8,31	73,3	1,37	5,4
11	CLIENT	87732	OS-1520		3,33	0	0,0	0,0	8,31	0	2	8,31	0	0,0	8,31	0,0	8,31	7,3
12	CLIENT	87733	OS-1520		4,17	0	0,0	0,0	8,31	0	2	8,31	0	0,0	8,31	0,0	8,31	7,3
13	CLIENT	87734	OS-1520		1,00	0	0,0	68,4	5,15	0	2	8,31	0	0,0	8,31	68,4	1,37	5,4
14	CLIENT	87735	OS-1520		4,17	0	0,0	7,4	8,31	0	2	8,31	0	0,0	8,31	7,4	8,09	7,3
15	CLIENT	87750	OS-1520		2,17	0	0,0	0,0	8,31	0	2	8,31	0	0,0	8,31	0,0	8,31	7,3
16	TESTE	88066	OS-1520		2,00	0	0,0	0,0	8,31	0	2	8,31	0	0,0	8,31	0,0	8,31	7,3

Figura 5.6 – Exemplo de resultado de avaliação do Componente de Inferência

5.3.2.1 Construção do componente através de entrevista

As regras e as respectivas funções de agregação foram consolidadas através das entrevistas. Primeiramente, o entrevistado relatou qual a importância (baixa, média ou alta) de cada dimensão para determinado grau de qualidade (Muito baixa qualidade, Razoável qualidade, Boa qualidade e Ótima qualidade), conforme pode ser visualizado na Figura 5.7.

2. De acordo com as definições de cada dimensão, como você caracteriza um dado de:

a. Muito baixa qualidade

Acurácia ()Alta ()Média (x)Baixa
 Atualidade ()Alta (x)Média (x)Baixa
 Completude ()Alta (x)Média (x)Baixa
 Consistência ()Alta ()Média (x)Baixa

b. Razoável qualidade

Acurácia ()Alta ()Média ()Baixa
 Atualidade ()Alta ()Média ()Baixa
 Completude ()Alta ()Média ()Baixa
 Consistência ()Alta ()Média ()Baixa

c. Boa qualidade

Acurácia ()Alta ()Média ()Baixa
 Atualidade ()Alta ()Média ()Baixa
 Completude ()Alta ()Média ()Baixa
 Consistência ()Alta ()Média ()Baixa

d. Ótima qualidade

Acurácia ()Alta ()Média ()Baixa
 Atualidade ()Alta ()Média ()Baixa
 Completude ()Alta ()Média ()Baixa
 Consistência ()Alta ()Média ()Baixa

3. Que outras dimensões você acha que é importante na avaliação de qualidade de dados de métricas de esforço de PDS?

Figura 5.7 – Parte da entrevista

Por exemplo, se um entrevistado relata que: “se o dado possui baixa acurácia e consistência esse será considerado como de muito baixa qualidade, não importando se possui atualidade e completude baixa ou média, pois não adianta o dado ser atual e completo se não possuir adequada acurácia e consistência”, nesse caso uma regra é constituída da seguinte maneira:

$$Se\ acurácia=baixa\ E\ Consistência=baixa\ E$$

Se atualidade=baixa OU atualidade=media E

Se completude=baixa OU completude=media

Então Qualidade=Muitobaixa.

Esse conhecimento é adquirido através da marcação do entrevistado na folha (Figura 5.7) e de anotações realizadas durante a entrevista.

Com as variáveis de DP selecionadas, o entrevistado as relacionou com cada dimensão através de regras, definindo o que caracteriza um dado de baixa, média ou alta acurácia/atualidade/completude/ consistência.

Por exemplo, se o entrevistado relata que: “*se o membro de equipe (quem lança o dado) não lançou suas horas ao final do dia de trabalho, então provavelmente o dado tem acurácia duvidosa e dependendo de quanto tempo ele demorou para registrar, essa dúvida tende a aumentar*, nesse caso as variáveis de proveniência utilizadas são dia do lançamento (x), dia do registro(y) e a diferença entre x e y (z), e o bloco de regras a ser criada é o seguinte:

Se z = grande então acurácia=baixa

Se z = médio então acurácia=média

Se z = pequeno então acurácia=alta

A Figura 5.8 mostra um exemplo de marcação por um entrevistado.

Acurácia

1()o valor original lançado
 2(x)o dia do registro (dia escolhido para lançar a hora)
 3(x)dia do lançamento (dia atual)
 4()fase de lançamento
 5()fonte de lançamento (ferramentas de gerência)
 6()valor modificado (por transformações, adaptações, correções)
 7()diferença do valor original com um valor esperado(discrepância / quanto?)
 8()diferença do valor original com o valor modificado(discrepância)
 9()experiência
 10(x) diferença entre 2 e 3

Figura 5.8 – Exemplo de marcação pelo entrevistado

Em seguida, cada entrevistado definiu, de acordo com a sua experiência e conhecimento empírico, o que pode ser considerado um valor para z grande, médio e pequeno e a intersecção de níveis, como pode ser mostrado no exemplo de gráfico de uma função de pertinência na Fig.5.9.

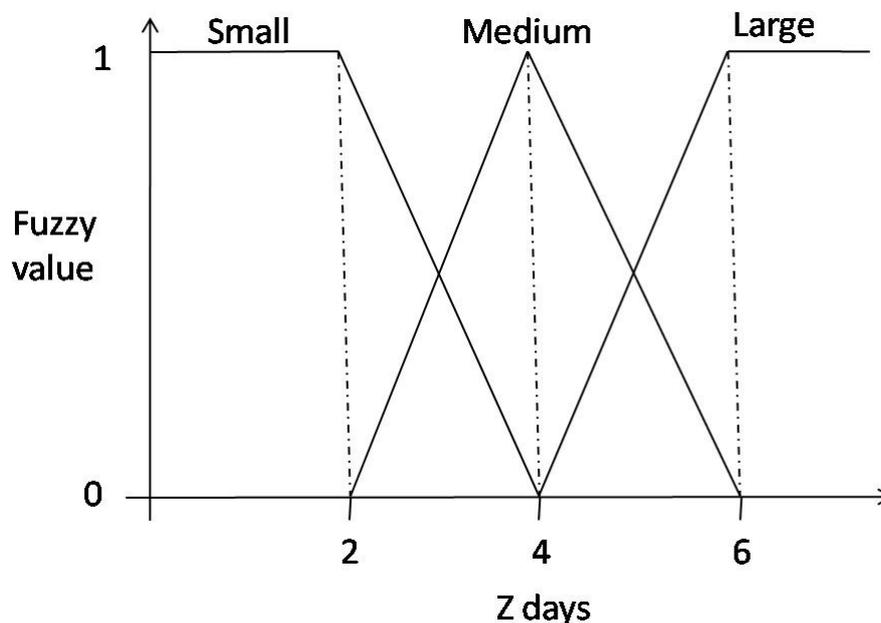


Figura 5.9 – Exemplo de função de pertinência para acurácia

A Fig.5.9 ilustra um exemplo de função de pertinência para uma declaração do tipo: “até uns 2 dias de diferença pode-se considerar um valor pequeno, no entanto até o 4º dia é uma diferença média, mas de 6 dias em diante é considerada uma diferença grande.” A principal vantagem do processamento em regras através de funções de pertinência que resultam valores entre $[0,1]$ é a flexibilidade na classificação do dado podendo fazer uso da incerteza utilizada no processo de tomada de decisão humana [Altrock, 1995]. No caso da métrica de esforço, os valores fuzzy indicam quão aderente está um dado com cada nível de qualidade, e não somente indica uma probabilidade de o dado ser classificado em cada nível. Por exemplo, a informação de que o dado é de baixa qualidade, mas que ele é quase de boa qualidade é uma informação mais rica e útil do que simplesmente dizer que o dado é de baixa qualidade. Além disso, a avaliação fuzzy se mostra mais adequada ao contexto de PDS, cuja complexidade e dinamismo são características que não podem ser ignoradas no momento de uma avaliação, tornando assim uma avaliação dicotômica mais passível de erros. É interessante observar que não foram encontradas pesquisas que utilizam lógica fuzzy para avaliação de qualidade de dados no contexto de métricas de software. Entretanto, lógica fuzzy tem sido amplamente utilizada nesse contexto em outras áreas, como estimativas de software e gerenciamento de qualidade de software [[Ebert, 1996], [Witold Pedrycz, 1998], [Andrew Gray, 1997]].

5.3.2.2 Implementação da máquina de inferência

Para a implementação da máquina de inferência foi utilizado a ferramenta Matlab versão 12 que possui um simulador fuzzy onde é possível implementar as funções de pertinência e o bloco de regras. Através da construção de um script com sufixo .m o recurso lê como entrada um arquivo com os dados a serem avaliados e grava em arquivo a avaliação de cada dado, conforme pode ser visualizado abaixo o script implementado.

```
diary resultado.txt;
diary on;
a=readfis('nome-do-arquivo');
matriz=load('arquivo-entrada.dat');
l=length(matriz); % l contem o numero de linhas da matriz
for i=1:l
prov=matriz(i,:);
evalfis([prov],a) end;
diary off;
```

A seguir é descrita a implementação de cada dimensão de acordo com os conceitos de sistemas de inferência fuzzy estudados na Fundamentação Teórica (Cápítulo 2), onde devem ser definidos: (i) variáveis lingüísticas, (ii) termos e (iii) funções de pertinência.

Acurácia

A Figura 5.10 mostra o modelo de inferência para o cálculo da dimensão Acurácia com as variáveis de entrada nas caixas amarelas e a variável de saída na caixa azul. A caixa branca representa o passo da inferência onde as regras são executadas.

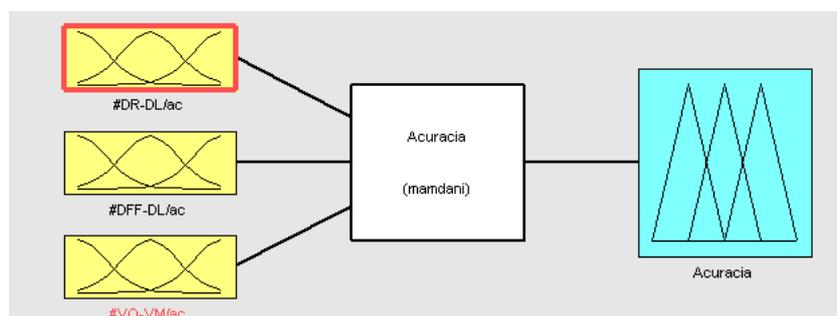


Figura 5.10 – Modelo Mamdani da dimensão Acurácia

- Variáveis lingüísticas e termos:

$$\#DR - DL \in \{baixa, média alta\}$$

$$\#DFF - DR \in \{baixa, média, alta\}$$

$$\#VO/ - VM \in \{baixa, média, alta\}$$

- Funções de pertinência

As funções de pertinência das variáveis envolvidas no cálculo da dimensão acurácia, implementadas no Matlab, são expostas a seguir.

A Figura 5.11 ilustra a implementação da variável #DR-DL para a dimensão Acurácia. Nesta figura a variável está descrita na forma de “#DR-DL/ac” porque essa variável também é considerada no cálculo de outras dimensões e o “/ac” indica que esta função é especificamente para acurácia. Assim também ocorre com a nomenclatura das outras variáveis “#DFF-DL/ac” e “#VO-VM/ac”.

A função de pertinência na Figura 5.11 é representada no eixo x por “dias” e no eixo y por “valor fuzzy”. Neste caso, a função permite uma variação de até “10 dias”, o que pode ser alterado dependendo do contexto de aplicação ou de teste. Essa representação também ocorre com a variável #DFF-DR na Figura 5.13. Diferentemente, a função de pertinência da variável #VO-VM/ac é representada no eixo x por “porcentagem” e no eixo y por “valor fuzzy”. Essa porcentagem pode ter uma variação de até 100%.

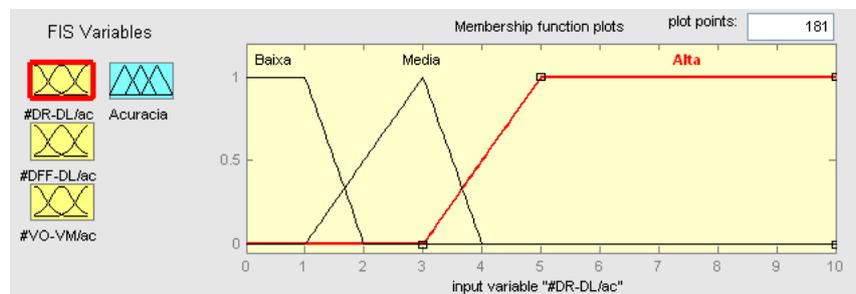


Figura 5.11 – Função de Pertinência da Variável #DR-DL

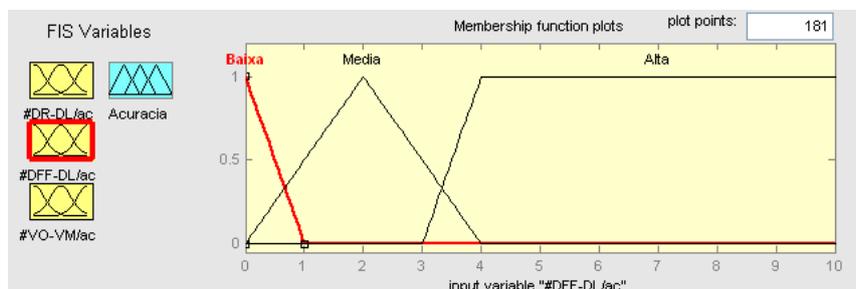


Figura 5.12 – Função de Pertinência da Variável #DFF-DR

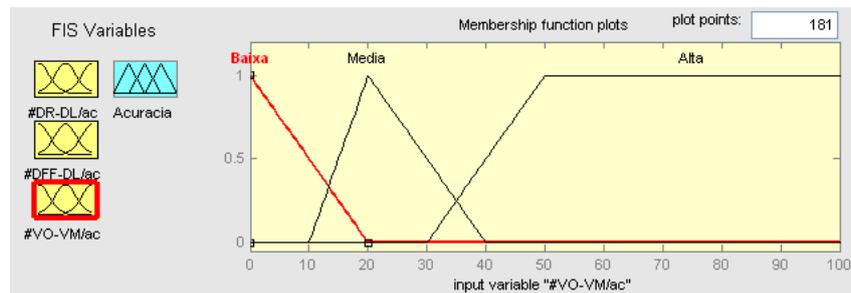


Figura 5.13 – Função de Pertinência da Variável #VO-VM

A variável de saída Acurácia na Figura 5.14 possui uma função de pertinência onde o eixo y é o valor fuzzy eixo x é um valor defuzzyficado em um sistema de “nota” que varia de 0 a 10. Essa saída servirá de entrada para o segundo nível de processamento da máquina de inferência.

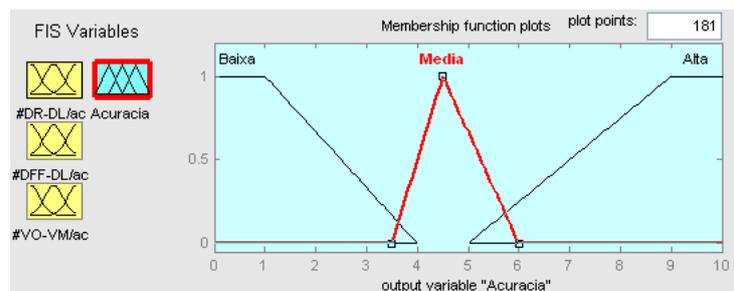


Figura 5.14 – Função de Pertinência da Variável de saída Acurácia

- Regras de produção

As regras de produção foram inseridas no Matlab através de um recurso dessa ferramenta e podem ser vistas na Figura 5.15. Essas regras são resultado do processo de aquisição de conhecimento já apresentado.

```

1. If (#DFF-DL/ac is Baixa) or (#VO-VM/ac is Baixa) then (Acuracia is Alta) (1)
2. If (#DFF-DL/ac is Media) or (#VO-VM/ac is Media) then (Acuracia is Media) (1)
3. If (#DR-DL/ac is Alta) or (#DFF-DL/ac is Alta) or (#VO-VM/ac is Alta) then (Acuracia is Baixa) (1)
    
```

Figura 5.15 – Regras de produção da dimensão Acurácia

Atualidade

A Figura 5.16 mostra o modelo de inferência para o cálculo da dimensão Atualidade com as variáveis de entrada nas caixas amarelas e a variável de saída na caixa azul. A caixa

branca representa o passo da inferência onde as regras são executadas.

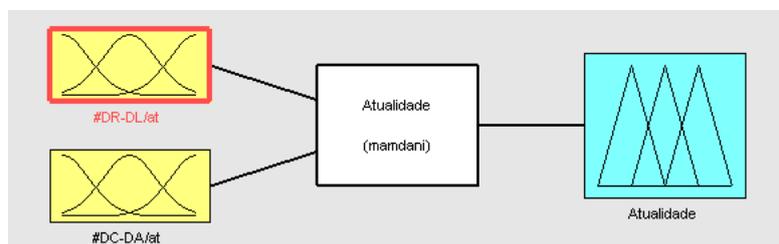


Figura 5.16 – Modelo Mamdani da dimensão Atualidade

- Variáveis linguísticas e termos:

$$\#DR - DL \in \{baixa, média alta\}$$

$$\#DC - DA \in \{baixa, média, alta\}$$

- Funções de pertinência

As funções de pertinência das variáveis envolvidas no cálculo da dimensão atualidade, implementadas no Matlab, são expostas a seguir.

A Figura 5.17 ilustra a implementação da variável $\#DR-DL$ para a dimensão Atualidade. Nesta figura a variável está descrita na forma de “ $\#DR-DL/at$ ” porque essa variável também é considerada no cálculo de outras dimensões e o “/at” indica que esta função é especificamente para atualidade. Assim também ocorre com a nomenclatura da outra variável “ $\#DC-DA/at$ ”.

A função de pertinência na Figura 5.17 é representada no eixo x por “dias” e no eixo y por valor fuzzy. Neste caso, a função permite uma variação de até “10 dias”, o que pode ser alterado dependendo do contexto de aplicação ou de teste. A representação da variável $\#DC-DA$ na Figura 5.18 também é no eixo x é por dias e no eixo y por valor fuzzy. Essa porcentagem pode ter uma variação de até 30 dias.

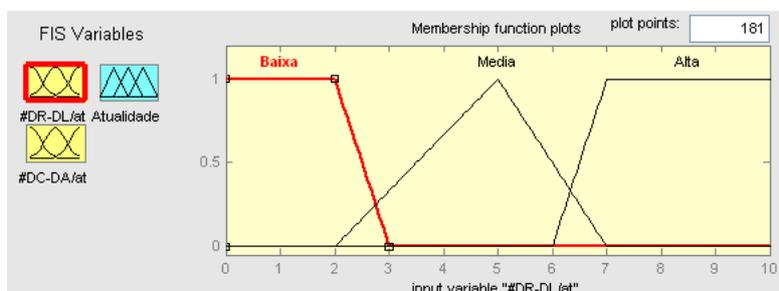


Figura 5.17 – Função de Pertinência da Variável $\#DR-DL$

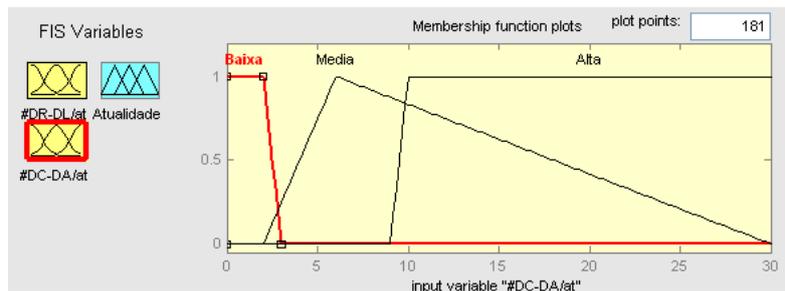


Figura 5.18 – Função de Pertinência da Variável #DC/DA

A variável de saída Atualidade na Figura 5.19 possui uma função de pertinência onde o eixo y é o valor fuzzy eixo x é um valor defuzzyficado em um sistema de “nota” que varia de 0 a 10. Essa saída servirá de entrada para o segundo nível de processamento da máquina de inferência.

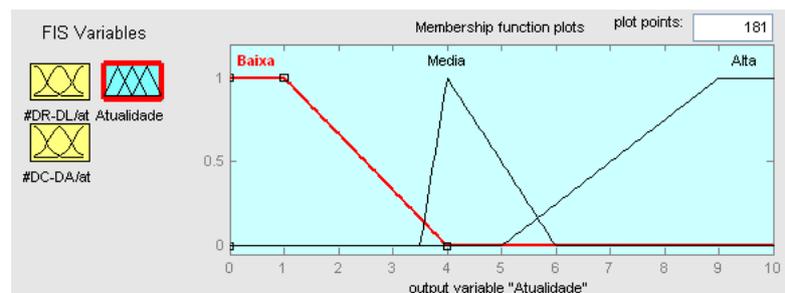


Figura 5.19 – Função de Pertinência da Variável de saída Atualidade

- Regras de produção

As regras de produção foram inseridas no Matlab através de um recurso dessa ferramenta e podem ser vistas na Figura 5.20. Essas regras são resultado do processo de aquisição de conhecimento já apresentado.

```

1. If (#DR-DL/at is Baixa) or (#DC-DA/at is Baixa) then (Atualidade is Alta) (1)
2. If (#DR-DL/at is Media) or (#DC-DA/at is Media) then (Atualidade is Media) (1)
3. If (#DR-DL/at is Alta) or (#DC-DA/at is Alta) then (Atualidade is Baixa) (1)
    
```

Figura 5.20 – Regras de produção da dimensão Atualidade

Compleitude

Figura 5.21 mostra o modelo de inferência para o cálculo da dimensão Compleitude com as variáveis de entrada nas caixas amarelas e a variável de saída na caixa azul. A caixa

branca representa o passo da inferência onde as regras são executadas.

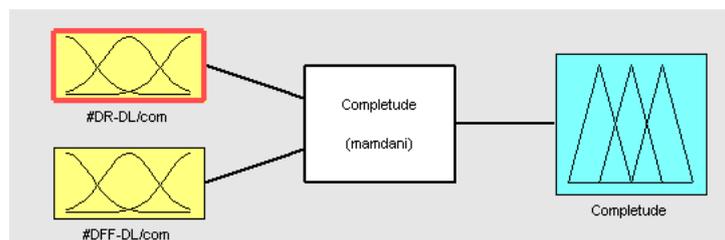


Figura 5.21 – Modelo Mamdani da dimensão Completude

- Variáveis linguísticas e termos:

$$\#DR - DL \in \{baixa, média alta\}$$

$$\#DFF - DR \in \{baixa, média, alta\}$$

- Funções de pertinência

As funções de pertinência das variáveis envolvidas no cálculo da dimensão completude, implementadas no Matlab, são expostas a seguir.

A Figura 5.22 ilustra a implementação da variável $\#DR-DL$ para a dimensão Completude. Nesta figura a variável está descrita na forma de “ $\#DR-DL/com$ ” porque essa variável também é considerada no cálculo de outras dimensões e o “/com” indica que esta função é especificamente para completude. Assim também ocorre com a nomenclatura da outra variável “ $\#DFF-DR/at$ ”.

A função de pertinência na Figura 5.22 é representada no eixo x por “dias” e no eixo y por valor fuzzy. Neste caso, a função permite uma variação de até “10 dias”, o que pode ser alterado dependendo do contexto de aplicação ou de teste. A representação da variável $\#DFF-DR$ na Figura 5.23 também é no eixo x é por dias e no eixo y por valor fuzzy. Essa porcentagem pode ter uma variação de até 10 dias.

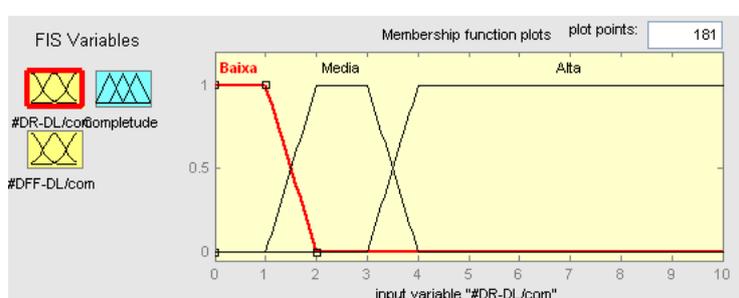


Figura 5.22 – Função de Pertinência da Variável $\#DR-DL$

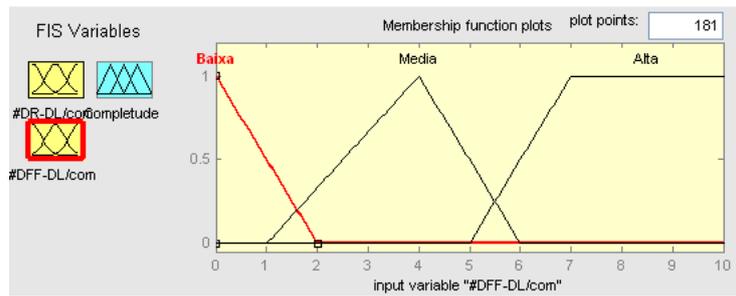


Figura 5.23 – Função de Pertinência da Variável #DFF-DR

A variável de saída Completude na Figura 5.24 possui uma função de pertinência onde o eixo y é o valor fuzzy eixo x é um valor defuzzyficado em um sistema de “nota” que varia de 0 a 10. Essa saída também servirá de entrada para o segundo nível de processamento da máquina de inferência.

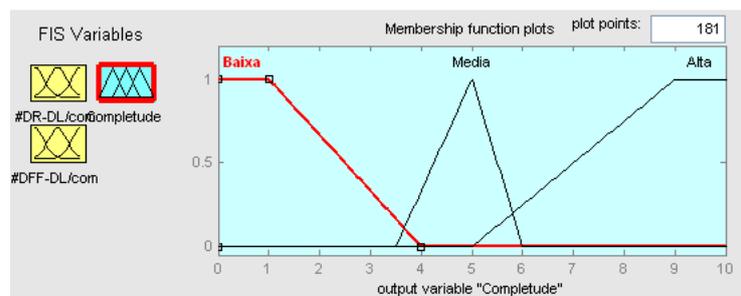


Figura 5.24 – Função de Pertinência da Variável de saída Completude

- Regras de produção

As regras de produção foram inseridas no Matlab através de um recurso dessa ferramenta e podem ser vistas na Figura 5.25. Essas regras são resultado do processo de aquisição de conhecimento já apresentado.

```

1. If (#DR-DL/com is Baixa) or (#DFF-DL/com is Baixa) then (Completude is Alta) (1)
2. If (#DR-DL/com is Media) or (#DFF-DL/com is Media) then (Completude is Media) (1)
3. If (#DR-DL/com is Alta) or (#DFF-DL/com is Alta) then (Completude is Baixa) (1)
    
```

Figura 5.25 – Regras de produção da dimensão Completude

Consistência

Figura 5.26 mostra o modelo de inferência para o cálculo da dimensão Completude com as variáveis de entrada nas caixas amarelas e a variável de saída na caixa azul. A caixa

branca representa o passo da inferência onde as regras são executadas.

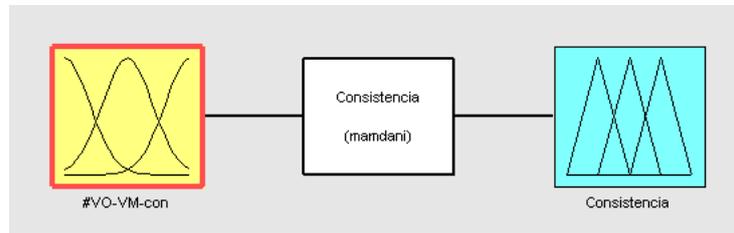


Figura 5.26 – Modelo Mamdani da dimensão Consistência

- Variáveis linguísticas e termos:
 $\#VO - VM \in \{baixa, média, alta\}$
- Funções de pertinência

As funções de pertinência das variáveis envolvidas no cálculo da dimensão consistência, implementadas no Matlab, são expostas a seguir.

A Figura 5.27 ilustra a implementação da variável #VO-VM para a dimensão Consistência. Nesta figura a variável está descrita na forma de “#VO-VM/con” porque essa variável também é considerada no cálculo de outras dimensões e o “/con” indica que esta função é especificamente para consistência.

A função de pertinência na Figura 5.27 é representada no eixo x por porcentagem e no eixo y por valor fuzzy. Neste caso, a função permite uma variação de até 100 por cento, o que pode ser alterado dependendo do contexto de aplicação ou de teste.

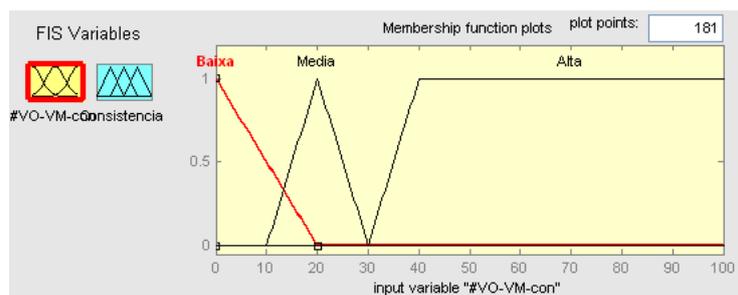


Figura 5.27 – Regra de Produção da Variável #VO-VM

A variável de saída Consistência na Figura 5.28 possui uma função de pertinência onde o eixo y é o valor fuzzy eixo x é um valor defuzzyficado em um sistema de “nota” que varia de 0 a 10. Essa saída também servirá de entrada para o segundo nível de processamento da máquina de inferência.

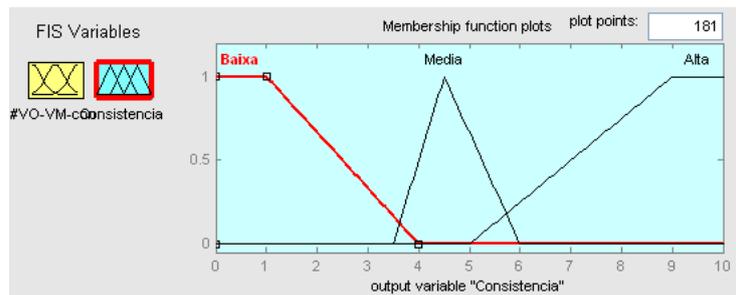


Figura 5.28 – Função de Pertinência da Variável de saída Consistêcia

- Regras de produção

As regras de produção foram inseridas no Matlab através de um recurso dessa ferramenta e podem ser vistas na Figura 5.29. Essas regras são resultado do processo de aquisição de conhecimento já apresentado.

```

1. If (#VO-VM-con is Baixa) then (Consistencia is Alta) (1)
2. If (#VO-VM-con is Media) then (Consistencia is Media) (1)
3. If (#VO-VM-con is Alta) then (Consistencia is Baixa) (1)
    
```

Figura 5.29 – Regras de produção da dimensão Consistêcia

Qualidade

Figura 5.30 mostra o modelo de inferência do segundo nível de processamento da máquina de inferência que calcula o resultado final da qualidade. Como entrada (caixas amarelas) é o resultado do primeiro nível de processamento, onde foi calculado o nível de pertinência de cada dimensão e como saída (caixa azul) é o resultado final da qualidade do dado (defuzzyficado).

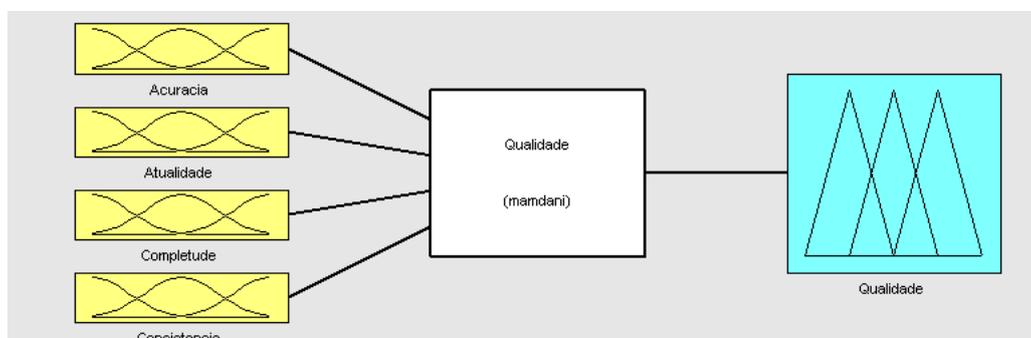


Figura 5.30 – Modelo Mamdani da Qualidade

- Variáveis linguísticas e termos:
 $Acurácia \in \{baixa, média, alta\}$
 $Atualidade \in \{baixa, média, alta\}$
 $Completude \in \{baixa, média, alta\}$
 $Consistência \in \{baixa, média, alta\}$
- Funções de pertinência

Na Figura 5.31 a entrada é um escalar defuzzyficado no primeiro nível de processamento, que está representado em um sistema de notas (de 0 a 10) como mostra o eixo x da figura. No eixo y é o valor fuzzy correspondente à entrada. O mesmo ocorre com as outras funções de pertinência do restante das dimensões Atualidade, Completude e Consistência.

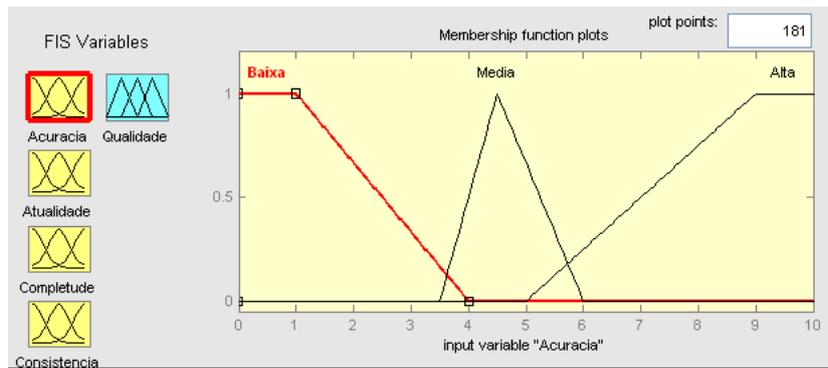


Figura 5.31 – Função de Pertinência da Variável Acurácia

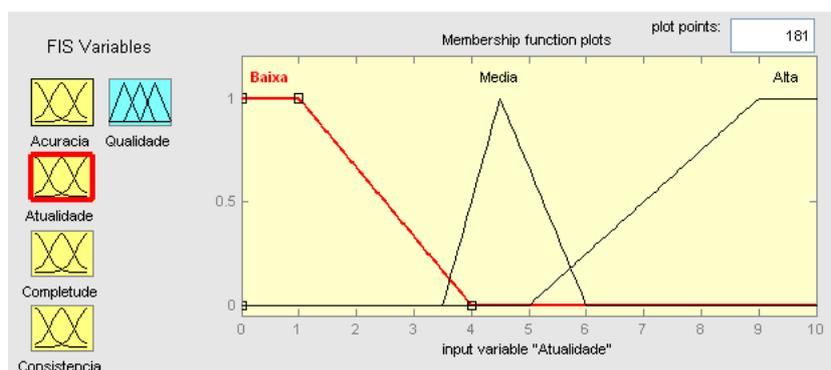


Figura 5.32 – Função de Pertinência da Variável Atualidade

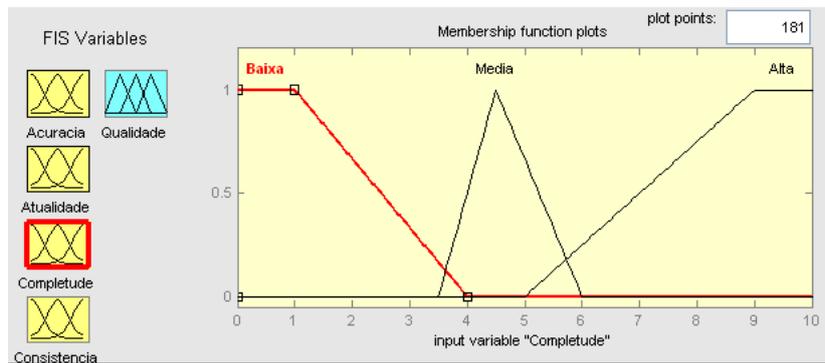


Figura 5.33 – Função de Pertinência da Variável Compleitude

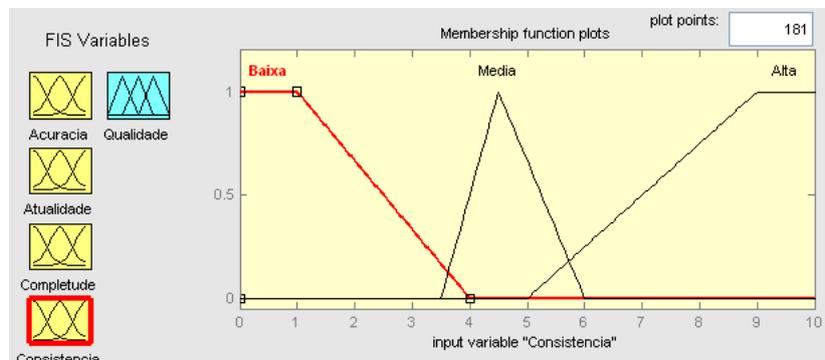


Figura 5.34 – Função de Pertinência da Variável Consistencia

A Figura 5.35 mostra a função de pertinência que calcula o valor final retornando um escalar representado num sistema de notas em que pode ser entendido de forma linguística através do acesso à Figura 5.37.

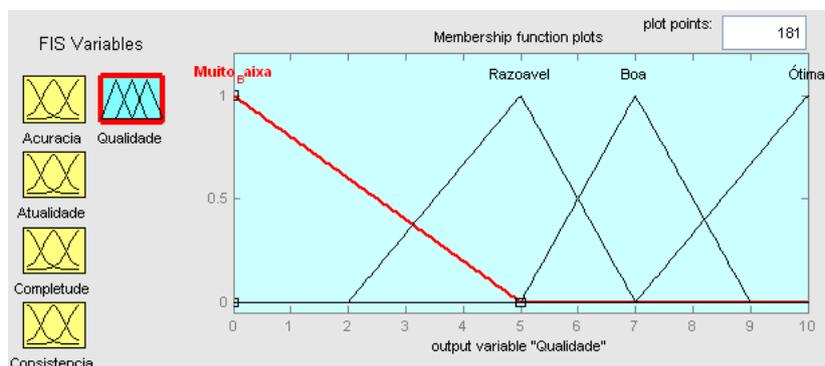


Figura 5.35 – Função de Pertinência da Variável de saída Qualidade

- Regras de produção

A implementação das regras de produção para a Qualidade foi realizada da mesma forma já descrita, através de um recurso da ferramenta Matlab. A Figura 5.36 mostra as regras aplicadas. É importante esclarecer que para o nível de qualidade Razoável foi possível identificar duas regras distintas através das entrevistas realizadas. A variação do critério de Completude e Consistência fez com que essas duas regras (2 e 3) fossem desenvolvidas. A máquina de inferência interpreta como uma operação OU no cálculo das duas, e de acordo com as operações fuzzy o OU é resultado de uma operação max entre as duas regras.

```

1. If (Acuracia is not Alta) and (Atualidade is Baixa) and (Consistencia is not Alta) then (Qualidade is Muito_Baixa) (1)
2. If (Acuracia is not Baixa) and (Atualidade is not Alta) and (Completude is not Baixa) and (Consistencia is not Baixa) then (Qualidade is Razoavel) (1)
3. If (Acuracia is not Alta) and (Atualidade is not Alta) and (Completude is not Alta) and (Consistencia is not Alta) then (Qualidade is Razoavel) (1)
4. If (Acuracia is not Baixa) and (Atualidade is not Baixa) and (Completude is not Baixa) and (Consistencia is not Baixa) then (Qualidade is Boa) (1)
5. If (Acuracia is Alta) and (Atualidade is Alta) and (Completude is not Baixa) and (Consistencia is not Baixa) then (Qualidade is Ótima) (1)

```

Figura 5.36 – Regras de produção para cálculo da Qualidade

Para a implementação da máquina de inferência foi utilizado o modelo Mamdani e para a etapa de defuzzyficação foi utilizado o centroid (nomenclatura do Matlab) que é o método de Centro de Gravidade, anteriormente discutido (Seção 4.1 do Capítulo 2).

Como a função de pertinência retorna um valor e não uma classificação linguística, a Figura 5.37 ilustra essa conversão para o uso da máquina de inferência como se propõe neste trabalho de acordo com a aquisição de conhecimento feita nas entrevistas.

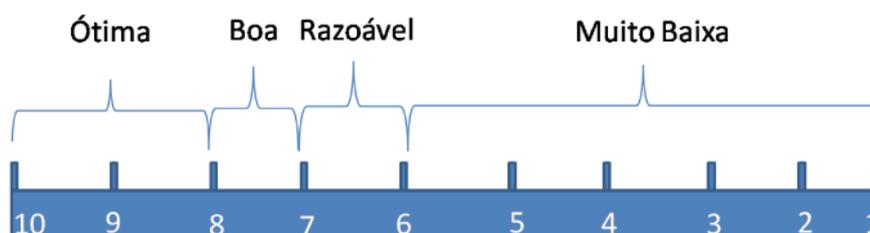


Figura 5.37 – Classificação linguística

5.3.3 Componente de Qualidade

O bloco do Componente de Qualidade, (c) na Figura 5.1, mostra uma Base de Qualidade que armazena a saída da máquina de inferência executada no bloco (b). A Tabela de Qualidade contém avaliação de lançamento (identificado pelo ID_lançamento) de acordo com cada dimensão de qualidade e ainda apresenta o grau de aderência do lançamento aos níveis de qualidade Muito Baixa (MB), Razoável (R), Boa (B) e Ótima (B).

Como resultado dos cálculos da máquina de inferência utilizando a proveniência dos dados, é gerada essa tabela para guardar a avaliação de cada dimensão e para cada nível de qualidade. Este é um componente bastante simples servindo como saída das regras e, também, como fonte para o Componente de Análise. A Tabela 5.3 exemplifica o registro de uma avaliação de qualidade de um dado de esforço identificado pelo ID=05, onde:

- Acur. é o valor defuzzyficado correspondente à avaliação do primeiro nível de processamento da máquina de inferência da dimensão Acurácia;
- At. é o valor defuzzyficado correspondente à avaliação do primeiro nível de processamento da máquina de inferência da dimensão Atualidade;
- Compl. é o valor defuzzyficado correspondente à avaliação do primeiro nível de processamento da máquina de inferência da dimensão Completude;
- Cons. é o valor defuzzyficado correspondente à avaliação do primeiro nível de processamento da máquina de inferência da dimensão Consistência;
- Ba, R, Bo e O dizem respeito à avaliação dos níveis de qualidade Baixo, Razoável, Boa e Ótima qualidade, resultado do segundo nível de processamento onde a entrada foi o valor defuzzyficado das dimensões e a saída é um valor defuzzyficado de qualidade, representado pela coluna Quali na Figura 5.3. Os valores Ba, R, Bo, O são valores fuzzyficados intermediários, utilizados entre o cálculo das dimensões e o cálculo final da Qualidade.

Tabela 5.3 – *Componente de Qualidade*

ID	Acur	At	Com	Cons	Ba	R	Bo	O	Quali
05	6,86	8,31	8,31	1,74	0	0.8	0.2	0	6,6

5.3.4 Componente de Análise

O bloco do Componente de Análise, (d) na Figura 5.1, diz respeito à construção de uma base analítica dimensionalmente modelada para proporcionar análises completas voltadas às necessidades dos gestores. A Base de Proveniência e Qualidade possibilita registrar o histórico da qualidade dos lançamentos de esforço em um projeto/organização. Isso possibilita análises não só do presente, mas também do passado e, conseqüentemente, de tendências futuras.

O componente que mantém o histórico das avaliações da organização e que possibilita análises de tendências para apontar possíveis pontos de melhoria é estruturado na forma de um data warehouse, modelo estrela [Kimball and Ross, 2002]. O componente, chamado aqui de DW-Q (5.38), é composto por uma tabela Fato (*Event*) que registra os valores de avaliação de cada dimensão de qualidade, também o resultado final de níveis de qualidade, além de medidas como *UpdateRate* (taxa de modificação dos dados), *RegisterLatency* (latência do registro dos dados, ou seja, o quanto os recursos demoram para registrar suas horas trabalhadas). Essa tabela fato é caracterizada pelas tabelas dimensão *When*, *What*, *Timeline*, *Who*, e *Where*.

A tabela dimensão *When* é responsável por possibilitar a análise de quando ocorrem os registros de esforço através da observação de ano, mês e data geradora do registro.

A tabela dimensão *What* mantém o valor original do esforço e informações de acontecimentos posteriores com o dado, quantas vezes ele foi modificado e o seu valor final que possibilita a análise de vulnerabilidade dos dados de esforço.

A dimensão *Who* disponibiliza a observação de colaboradores que possuem maior incidência de alteração de comportamento nos dados, permitindo aos gestores concluir se a equipe recebeu um bom treinamento antes de inserção no projeto e se existe alguém que pode estar deteriorando a qualidade dos dados, por exemplo.

A dimensão *Where* podem apontar as versões mais críticas de um projeto, com relação à qualidade de seus dados. Através de uma análise mais detalhada das versões de um projeto é possível observar aspectos e ações das equipes e até mesmo de clientes que podem influenciar na qualidade dos dados. Por exemplo, pode ser observado que uma versão que apresenta muito paralelismo nas atribuições de atividades apresenta um alto índice de baixa qualidade nos dados.

Finalmente, a tabela dimensão *Timeline* mantém o registro histórico de análises dos dados, mostrando a freqüência de avaliação. O componente DW-Q usa como fonte o componente de proveniência (*Provenance*), componente de qualidade (*Quality DataBase*)

e a fonte de esforço do projeto (*Effort Source*), sendo disponibilizado para os gestores na frequência definida pela organização.

O DW é apropriado com a periodicidade que a organização define (semanalmente, quinzenalmente ou mensalmente). A única condição para essa definição é que os dados avaliados sejam referentes ao período corrente e que não sejam dados “ativos” no processo, ou seja, que não necessitem de atualizações. Isso porque o modelo proposto não dá suporte a avaliações parciais; cada dado deve ter finalizado o seu ciclo.

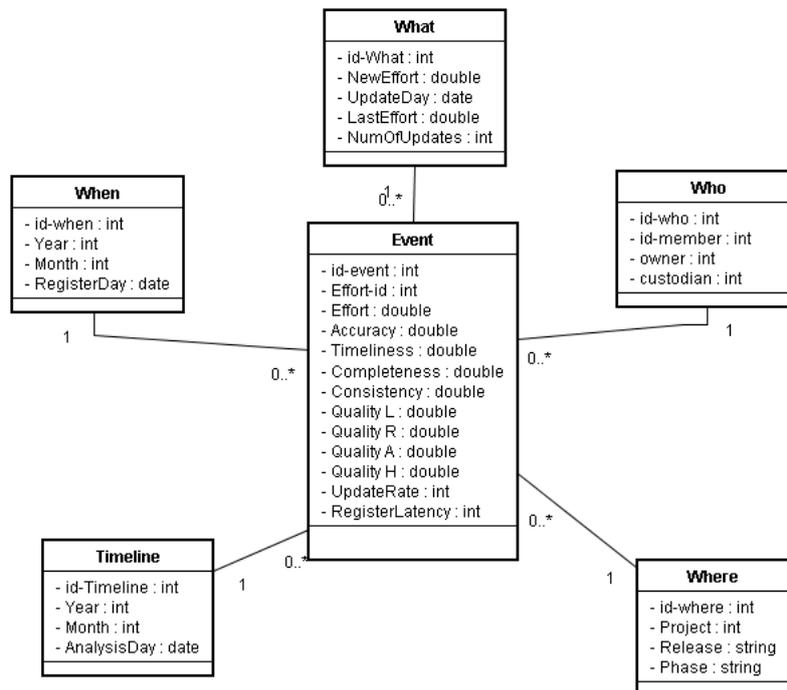


Figura 5.38 – DW-Q: Modelo do DW para a qualidade dos dados

5.4 Considerações finais sobre o Capítulo

Neste Capítulo foi apresentada a arquitetura proposta como solução juntamente com o detalhamento da construção de seus componentes. É importante ressaltar que a etapa de aquisição de conhecimento foi bastante trabalhosa e não trivial, exigindo bastante tempo e planejamento. No entanto, acreditamos que acrescentou significativamente valor à qualidade da solução por capturar um conhecimento real inerente às pessoas envolvidas em um ambiente de software. No próximo Capítulo a solução é testada com o intuito de observar sua utilidade e aplicabilidade em um ambiente real trazendo benefícios em algumas tarefas

relacionadas ao processo de tomada de decisão.

6

Testes da solução proposta

Este capítulo relata o conjunto de testes realizados sobre a solução proposta no Capítulo 5. Para tanto, apresenta: (i) os objetivos e procedimentos de teste; (ii) a execução dos testes; e (iii) e a análise dos mesmos.

6.1 Objetivos e procedimento

Os testes têm como objetivo principal averiguar se a avaliação esquematizada pela arquitetura consegue atender aos requisitos (Capítulo 3) inicialmente colocados como essenciais para uma solução adequada para o problema de avaliação de qualidade de dados de métricas de esforço de software. Para isso, simulamos a utilização da arquitetura em uma situação real de tomada de decisão que é o processo de estimativas através da utilização de algoritmos de mineração de dados.

Tendo em vista esse objetivo de uso de métricas de esforço, lançamos a seguinte hipótese: Os índices de erros diminuem quando os dados são previamente avaliados e preparados antes da construção do modelo preditivo. Se a hipótese for aceita, significa que a arquitetura contribui com o processo de tomada de decisão e ainda que a avaliação da arquitetura está calibrada para executar uma avaliação correta. Por outro lado, se a hipótese for rejeitada significa que a arquitetura pode até contribuir com o processo, mas ela ainda não está calibrada para executar uma avaliação correta. Para obter mais de uma possibilidade de observação da hipótese construímos três modelos preditivos, onde o primeiro serve de referência pois é construído sem submeter os dados a avaliação e portanto não sofreram preparação, no segundo e no terceiro os dados de entrada de cada um são previamente avaliados e preparados de acordo com critérios de níveis de qualidade diferentes.

Uma observação importante é com relação ao processo bem comportado que a arquitetura prevê e que não é considerado no teste devido a impossibilidade de testar alguns componentes da arquitetura separadamente. Basicamente no teste é considerado o componente de inferência que recebe como entrada a proveniência dos dados de esforço que são simulados e não rastreados efetivamente devido à impossibilidade de teste no ambiente inicialmente projetado.

Outro fator importante a ser considerado no teste é que a granularidade das informações avaliadas é diferente da granularidade utilizada nos modelos preditivos que se pretende construir. Em outras palavras, a construção dos modelos preditivos prevê a utilização do atributo Tamanho (que pode ser em linhas de código, pontos de função ou casos de uso) e não existe o controle de Tamanho por membro de equipe. Logo, para a construção dos modelos preditivos a qualidade é sumarizada por média ,para cada OS.

Tendo em vista estas restrições, para realizar essa análise, foram definidos os seguintes passos:

1. Definir uma amostra de dados de esforço coletados de um ambiente real a serem submetidos à avaliação;
2. Atribuir proveniência desses dados através de simulação (dedução/intuitivamente);
3. Para a construção do modelo preditivo 1:
 - (a) Sumarizar os dados por OS (sem submeter à avaliação);
 - (b) Executar o algoritmo de mineração de dados para a construção do modelo preditivo 1;
4. Para a construção do modelo preditivo 2:
 - (a) Submeter todos os dados à avaliação de qualidade em cada dimensão;
 - (b) Submeter todos os dados à avaliação de qualidade total;
 - (c) Preparar os dados que receberam nível de qualidade abaixo de Boa (Razoável e Muito Baixa) substituindo-os pela média dos bons, baseado em um dos processos

de preparação de dados apontados por Tan em [Tan et al., 2006] de acordo com a Figura 5.37;

- (d) Sumarizar os dados por OS;
- (e) Executar o algoritmo de mineração de dados para a construção do modelo preditivo 2;

5. Para a construção do modelo preditivo 3:

- (a) Submeter todos os dados à avaliação de qualidade em cada dimensão;
- (b) Submeter todos os dados à avaliação de qualidade total;
- (c) Preparar os dados que receberam nível de qualidade abaixo de Razoável (Muito Baixa) substituindo-os pela média dos bons, baseado em um dos processos de preparação de dados apontados por Tan em [Tan et al., 2006] de acordo com a Figura 5.37;
- (d) Sumarizar os dados por OS;
- (e) Executar o algoritmo de mineração de dados para a construção do modelo preditivo 3;

6. Realizar a comparação dos índices de erros entre o modelo preditivo 1 e modelo preditivo 2 ;

7. Realizar a comparação dos índices de erros entre o modelo preditivo 1 e o modelo preditivo 3.

6.2 Execução

Passo 1 do procedimento

Para a execução dos dois testes foi escolhido o mesmo grupo de dados para que os dados escolhidos não influenciassem no sucesso da avaliação. Por questão de coerência, os dados utilizados nesse teste são dados semelhantes aos envolvidos na análise estatística do Trabalho Individual II [5], ou seja, os registros de esforço do projeto P1 da versão 11.00, que contém ordens de serviço (OS) envolvidas apenas nesta versão. Houve um passo, então,

de filtragem buscando os dados de OSs que fazem parte apenas desta versão, pois uma OS pode estar envolvida em mais de uma versão. No total, foram selecionadas 40 OSs, fechando o valor de 1041 registros de esforço distribuídas entre elas.

Passo 2 do procedimento

Tendo definidos os dados (passo 1 do procedimento), foi executado o passo que trata da atribuição da proveniência dos dados. Abaixo são discutidas a forma de obtenção das variáveis de proveniência.

- #DR-DL

Como na base fonte original do projeto real não possui rastreamento do dia de lançamento (DL), optou-se por obter essa diferença através da observação do crescimento da identificação da atividade e o decrescimento da data de registro (DR). A Tabela 6.1 ilustra um exemplo onde as atividades crescem (1, 2, 3, 4, 5, 6, 7) e a DR cresce mas em alguns momentos que tenderia a crescer ela decresce e volta a crescer novamente, nesses casos a diferença é calculada de acordo com a tendência. No caso das atividades 1, 2, 3 e 4 a DR está respeitando uma ordem natural de registro então a #DR-DL é 0. Já no caso da atividade 5, pela tendência ela deveria ter a DR como 2/1/2007 ou maior e no entanto ela decresce para 1/1/2007 resultando então em 1 dia de diferença no mínimo. Assim como ocorre com a atividade 7 que pela tendência deveria ter a DR como 3/1/2007 ou maior e no entanto ela decresce para 1/1/2007 indicando então um atraso de registro de pelo menos 2 dias no mínimo.

Tabela 6.1 – Exemplo de #DR-DL

Identificação da atividade	DR	#DR/DL
atividade 1	1/1/2007	0
atividade 2	1/1/2007	0
atividade 3	1/1/2007	0
atividade 4	2/1/2007	0
atividade 5	1/1/2007	1
atividade 6	3/1/2007	0
atividade 7	1/1/2007	2

- #DFF-DR

Para a obtenção dessa variável foi feita a diferença entre a data em que a fase finalizou (DFF) e a data do registro (DR). Sobre essa diferença foi feita a seguinte análise:

Se $DFF - DR < 0$ Então RegistroForaDoPrazo

Se $DFF - DR \geq 0$ Então RegistroDentroDoPrazo

Essa análise é necessária porque a variável #DFF/DR busca captar quanto o dado está atrasado em relação ao término da fase, não interessando os dados registrados antes do término nem no dia do término. Nesse sentido, a Tabela 6.2 mostra um exemplo de atribuição dessa variável de acordo com a análise descrita acima. No caso da atividade 2 o dado foi registrado dentro do prazo de fechamento da fase (DFF), logo a variável #DFF-DR recebe o valor 0, o que igualmente acontece com a atividade 3. Já a atividade 4 mostra um caso em que o dado foi lançado em uma fase já finalizada (2 dias após) então a variável #DFF-DR recebe o valor 2 indicando que está 2 dias fora do prazo.

Tabela 6.2 – Exemplo de #DFF-DR

Identificação da atividade	DFF	DR	DFF-DR	#DFF/DR
atividade 2	2/1/2006	1/1/2007	1	0
atividade 3	1/1/2007	1/1/2007	0	0
atividade 4	31/12/2006	2/1/2007	2	2

- #VO-VM

Como na base fonte original do projeto real não possui rastreamento do valor original (VO), optou-se por obter essa diferença através da observação de uma tabela de log onde alguns registros da base oficial de lançamento tinham seus respectivos lançamentos nessa tabela log. Então consideramos o valor oficial de lançamento como o VM, ou seja, o valor definitivo. Não é confirmado que o projeto utiliza essa tabela para este fim, mas foi uma opção para simular essa variável de proveniência. A Tabela 6.3 ilustra essa simulação. No caso da atividade 1, não existe registro na log (VO) logo entende-se que o valor não sofreu modificação, recebendo então #VOM-VM o valor 0 porque essa variável tem o intuito de observar as modificações que o dado sofre. Na atividade 2 existe registro de log, logo é possível observar que o dado foi modificado em 0,5 hora, o que ocorre de forma similar com a atividade 3 onde o dado

foi modificado em 1 hora. A diferença foi para menos; contudo, a variável #VO-VM não faz essa distinção através de sinais positivo e negativo porque o interesse é saber o quanto foi modificado e não se foi para mais ou para menos.

Tabela 6.3 – Exemplo de #VO-VM

Identificação da atividade	VO	VM	#VO/VM
atividade 1		2 horas	0
atividade 2	1 hora	1,5 horas	0,5
atividade 3	4 horas	3 horas	1

- #DC-DA

A obtenção dessa variável foi feita através do cálculo entre a data de coleta dos dados (DC) e a data de análise (DA). A Tabela 6.4 ilustra essa atribuição.

Tabela 6.4 – Exemplo de #DC-DA

Identificação da atividade	DC	DA	#DC/DA
atividade 1	30/1/2007	31/1/2007	1
atividade 2	30/1/2007	31/2/2007	2

Passo 3 do procedimento

O modelo preditivo foi construído utilizando o algoritmo de mineração M5P, através da ferramenta Weka [Witten and Frank, 2005] utilizando os seguintes atributos: tamanho, pf ajustados, estimado projeto, estimado server, estimado client, estimado integração, real projeto, real server, real client, real integração, n docs, total estimado e total real. Esses atributos foram considerados porque são os principais atributos envolvidos no registro de métricas de esforço abrangendo o tamanho medido em kloc ou pontos de função, o valor estimado por um especialista para cada fase (estimado projeto, estimado client, estimado integração), o valor real de cada fase registrado pelos colaboradores (real projeto, real server, real client e real integração), número de documentos envolvidos em um requisito representado pelo atributo n docs e total estimado e total real que são somas dos estimados e reais de cada fase. O atributo Total estimado é o atributo classe. Os atributos referentes a valores estimados para as fases projeto, client, server e integração são considerados porque

eles melhoram as estimativas, o que mostra que o especialista que estima atualmente não pode ser eliminado com a utilização de algoritmos de mineração. Portanto, a mineração é utilizada para aprimorar a prática dentro da organização junto ao especialista.

Realizando o método de teste chamado avaliação cruzada, foram obtidos índices de erros através da análise estatística realizada pela própria ferramenta Weka. Esses índices de erros são mostrados nos passos 6 e 7 que também mostra os índices de erros do modelo preditivo 2 (com os dados com qualidade inferior a Boa substituídos pela média dos bons) e do modelo preditivo 3 (com os dados com qualidade inferior a Razoável substituídos pela média dos bons).

Passo 4 do procedimento

O modelo preditivo foi construído utilizando os mesmos atributos utilizados na execução do passo 3 do procedimento. Os dados foram submetidos à avaliação da máquina de inferência e obtiveram resultados de qualidade para cada dimensão e também receberam um nível de qualidade total baseado nesses resultados de dimensões. Conforme indica a etapa (c) do passo 4 os dados que receberam nível de qualidade abaixo de Boa (Razoável e Muito Baixa) foram substituídos pela média dos bons, conforme indica um dos processos de preparação de dados apontados por Tan em [Tan et al., 2006].

O processo utilizado está descrito abaixo e a Tabela 6.5 ilustra a execução do mesmo.

1. Seleciona a OS;
2. Seleciona o critério de qualidade;
3. Calcula a média de esforço dos registros com qualidade acima do critério para cada fase;
4. Substitui o valor de esforço dos registros com qualidade abaixo do critério para cada fase;
5. Observação sobre o processo: quando a fase não possuir registros com qualidade acima do critério definido e, portanto, não for possível calcular média, optou-se por manter os registros mesmo com qualidade prejudicada.

Tabela 6.5 – Preparação de dados com qualidade inferior a Boa (≤ 7)

Identificação da atividade	OS	Fase	Esforço (hr)	Qualidade	Média	Backup-esforço
atividade 1	A	Projeto	3,5	8	3,6	3
atividade 2	A	Projeto	2,5	7	3,6	-
atividade 3	A	Projeto	5	7	3,6	-
atividade 4	A	Projeto	3,6	4	3,6	1,5
atividade 5	A	Client	3,75	7	3,75	-
atividade 6	A	Client	3,75	2	3,75	0,25
atividade 7	A	Server	2	2	-	-
atividade 8	A	Teste	3	8	-	-
atividade 9	A	Teste	4	8	-	-

Na Tabela 6.5 podem ser visualizados exemplos das situações que podem ser encontradas na preparação. As atividades 1, 2, 3 e 4 da OS A são da fase Projeto onde um desses registros recebeu avaliação inferior a Boa. De acordo com o processo de preparação descrito anteriormente foi realizada uma média dos registros com qualidade acima do critério, no caso, com os registros das atividades 1, 2 e 3 resultando em 3,6 sendo assim, o registro da atividade 4 que tinha o valor 1,5 horas passou a ter 3,6 horas. Outra situação ilustrada ocorre na fase Client onde dos 2 registros, um recebeu qualidade inferior a Boa, sendo este substituído pelo valor do outro (3,75) já que só possui um registro com qualidade superior a Boa.

Na fase Server, só existe uma atividade e que recebeu qualidade inferior a Boa, neste caso, de acordo com a observação apontada no processo de preparação, o registro de esforço se mantém igual. Por último, na fase Teste não foi necessário realizar nenhuma substituição já que todos os dados foram avaliados com qualidade superior a Boa. Após essa preparação o algoritmo M5P foi executado e o modelo preditivo foi construído e pode ser acessado no Apêndice C. A Tabela com os resultados é mostrada no passo 6 do procedimento, conforme anteriormente descrito.

Passo 5 do procedimento

Para a construção do modelo preditivo 3 foram executados os mesmos passos do passo 4 com a mudança no critério de qualidade. Neste modelo preditivo os dados que receberam qualidade inferior a Razoável (≤ 6), ou seja Muito Baixa foram substituídos pela média dos registros de qualidade superior a Razoável em cada fase. A Tabela com os resultados é

mostrada no passo 7 do procedimento, conforme anteriormente descrito e pode ser acessado no Apêndice C.

Passo 6 do procedimento

A Tabela 6.6 mostra os índices de erros obtidos na construção dos dois modelos preditivos, o primeiro é sem submeter os dados a avaliação e o segundo submete os dados à avaliação e prepara os que tem qualidade inferior a Boa (≤ 7).

Tabela 6.6 – Comparação dos índices de erro do modelo preditivo 1 e do modelo preditivo 2

Índice	Modelo preditivo 1	Modelo preditivo 2
Correlation Coefficient	0,923	0,8963
Mean absolute error	37,508	51,14
Root mean squared error	57,0924	90,7205
Relative absolute error	45,9478	46,4197
Root relative squared error	40,165	44,2403

Como pode ser observado na Tabela 6.6, com a execução desse passo a hipótese foi rejeitada, pois todos os índices de erros não melhoraram com a submissão dos dados à avaliação e sua posterior preparação. Uma análise aprofundada sobre as causas e a interpretação desse resultado é apresentada no final deste capítulo.

Passo 7 do procedimento

A Tabela 6.7 mostra os índices de erros obtidos na construção dos dois modelos preditivos, o primeiro é sem submeter os dados a avaliação e o segundo submete os dados à avaliação e prepara os que tem qualidade inferior a Razoável (≤ 6).

Tabela 6.7 – Comparação dos índices de erro do modelo preditivo 1 e do modelo preditivo

3

Índice	Modelo preditivo 1	Modelo preditivo 3
Correlation Coefficient	0,923	0,9259
Mean absolute error	37,508	37,0781
Root mean squared error	57,0924	55,1948
Relative absolute error	45,9478	45,0035
Root relative squared error	40,165	38,6572

Como pode ser observado na Tabela 6.7, com a execução desse passo a hipótese foi aceita, pois todos os índices de erros melhoraram com a submissão dos dados à avaliação e sua posterior preparação, além do coeficiente de correlação (*correlation coefficient*) ter aumentado. Uma análise aprofundada sobre as causas e a interpretação desse resultado é apresentada no final deste capítulo.

6.3 Análise dos resultados

Observando a Tabela 6.6 os resultados não são animadores visto que os índices de erro aumentaram e o índice de correlação diminuiu. Uma interpretação mais detalhada sobre esses resultados mostra que isto aconteceu devido o critério de qualidade ser razoavelmente exigente (≤ 7), onde foram mantidos apenas os dados com qualidade superior a 7 e todos os outros foram manipulados pela preparação. Essa manipulação pode ter sido excessiva à medida que uma grande porcentagem dos dados foi avaliada como abaixo de 7 (22% dos dados), e assim refletindo na qualidade do modelo preditivo construído a partir deles.

Por outro lado foi bastante animador o resultado obtido com a segunda preparação já que todos os índices de erros foram melhorados e o índice de correlação aumentou. Assim, como no primeiro resultado, acreditamos que o sucesso foi devido à baixa exigência com relação aos dados, não se fazendo necessário grande manipulação já que foram preparados apenas os que foram avaliados como muito baixa qualidade (10% dos dados). Já que esse resultado se mostrou promissor com a análise realizada na ferramenta Weka, realizamos outro teste estatístico (Tabela 6.8) bastante indicado pelos autores da área de mineração de dados: MMRE (*Mean Magnitude of Relative Error*) e PRED (25) [Witten and Frank, 2005]. Ao mostrar o resultado desse teste estatístico esclarecemos a sua interpretação.

Tabela 6.8 – *Teste estatístico MMRE e PRED(25)*

	Modelo Preditivo 1	Modelo Preditivo 3
MMRE	0,625439	0,617775
PRED(25)	0,375	0,55

Quanto menor o MMRE melhor é a estimativa, logo o modelo preditivo 3 é o melhor modelo dentre esses dois por apresentar o menor MMRE. Além disso, os valores de PRED(25) indicam que apesar de ambos (modelo preditivo 1 e 3) errarem bastante em suas estimativas (modelo preditivo 1 erra 62% e modelo preditivo 3 erra 61%), os erros do modelo preditivo 3 estão mais dentro da faixa de erro (55%) enquanto que os erros do modelo preditivo 1 estão apenas 37% dentro da margem de erro permitida.

Nesse sentido, acreditamos que a hipótese inicial de que os índices de erros diminuem quando os dados são previamente avaliados e preparados antes da construção do modelo preditivo pode ser aceita uma vez que encontramos um caso de sucesso de confirmação da hipótese, que foi a execução do passo 5 do procedimento de teste com a construção do modelo preditivo 3.

6.4 Outras análises

Como os benefícios de uma avaliação de qualidade de dados não são apenas em mineração de dados, outras análises foram realizadas para observar a utilidade e funcionalidade da solução proposta por este trabalho. Nesse sentido foram realizadas análises que podem ser executadas pelos gestores para observar tendências e pontos que merecem atenção para melhorar a qualidade dos dados. Para isso foram construídos os gráficos mostrados a seguir.

6.4.1 Estado da qualidade dos dados por mês no ano de 2007

Esta visão de membro por equipe (Server) permite a observação do usuários que estão contribuindo com a baixa qualidade dos dados, podendo o gestor tomar atitudes em relação à treinamentos e até mesmo em processo de conscientização. Outro aspecto que é interessante é a abertura das causas da baixa qualidade (apontado na coluna quali) da Tabela 6.9 através da avaliação das dimensões onde, ac é a dimensão acurácia, at é dimensão atualidade, com é a dimensão completude e cons é a dimensão consistência.

Tabela 6.9 – Tabela mês Março, equipe Server

mês Março - Fase SERVER					
usuario	ac	at	com	cons	quali
a	4,46	6,98	4,19	7,05	5,8
b	5,15	5,61	1,49	8,31	5,4
c	5,94	6,77	5,64	8,31	6,3
d	8,31	8,31	8,31	8,31	7,3
e	8,02	8,25	8,22	7,66	7,1
f	8,31	8,31	8,31	8,31	7,3
g	7,78	8,31	8,31	6,89	6,98
h	7,04	8,31	8,31	5,52	7,3
i	7,93	8,23	8,19	7,28	7,05
j	7,48	8,31	8,31	4,75	7,2
l	8,25	8,31	8,31	8,31	7,3

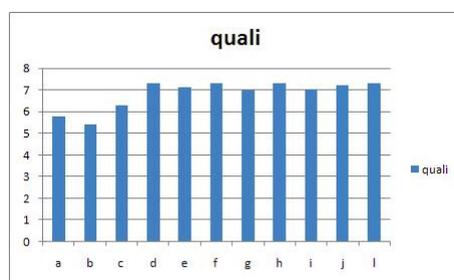


Figura 6.1 – Gráfico de qualidade de membro de equipe no mês Março na equipe Server

Observando o Gráfico 6.1 é possível notar que a maioria dos usuários contribui para uma qualidade considerada entre Razoável e Boa, com a exceção dos usuários a e b que apresentam a menor qualidade, abaixo de razoável, e o usuário a e c que apresentam como Razoável.

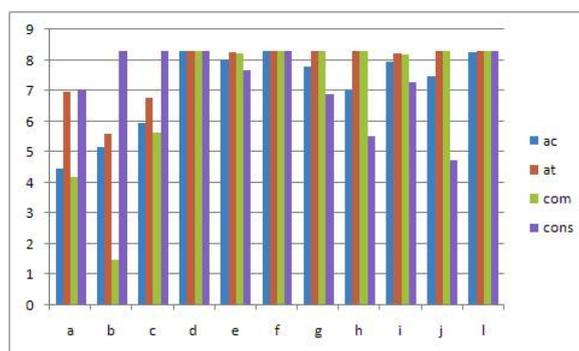


Figura 6.2 – Gráfico de qualidade de membro de equipe no mês Março na equipe Server por dimensões

Através da construção do Gráfico 6.2 o gestor pode ter a visualização de que forma o

membro de equipe está prejudicando a qualidade dos dados. Por exemplo, o membro b está apresentando o maior problema com relação à completude de seus registros, já o membro j está apresentando problemas com relação à consistência de seus registros. É importante salientar que essa análise por membro de equipe não tem o intuito de prejudicá-lo mas sim de conseguir contribuir para a sua melhoria dentro da equipe como um todo. Também é possível perceber grandes oscilações de qualidade nas dimensões, o que ocorre com os usuários h, i e j onde algumas dimensões são positivamente valoradas enquanto que outras deixam a desejar como na dimensão consistência.

Tabela 6.10 – Tabela de qualidade do mês Fevereiro, por projeto e dimensão

mês fevereiro				
equipe	media ac	media at	media com	media cons
projeto	8,28	8,31	8,29	8,18
client	7,8	8,26	8,16	7,23
server	8,17	8,31	8,31	7,96
teste	5,15	8,31	8,31	1,37

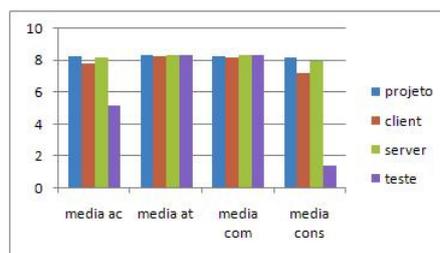


Figura 6.3 – Gráfico de qualidade do mês de Fevereiro por equipe e dimensão

Através da construção do Gráfico 6.3 com a Tabela 6.10 o gestor tem à sua disposição a visualização de equipes que apresentam problemas. Por exemplo, no gráfico do mês de fevereiro a equipe de teste apresenta grandes problemas com relação à acurácia e consistência. A utilização das dimensões facilita o apontamento de questões que necessitam de melhoria, e este gráfico mostra bem isto. Em uma análise geral as fases projeto e server não possuem grandes variações de qualidade nas dimensões. Enquanto que as outras fases, principalmente teste e client não possuem homogeneidade em seus registros de esforço.

Tabela 6.11 – Tabela de média de qualidade por equipe no mês de Fevereiro

mês fevereiro	
equipe	média quali
projeto	7,28
client	7
server	7,2
teste	5,4

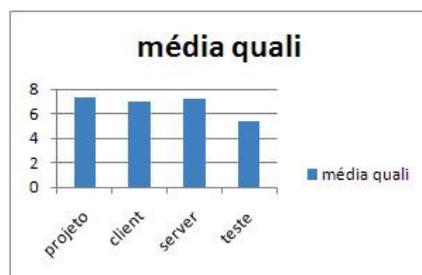


Figura 6.4 – Gráfico de média de qualidade por equipe no mês de Fevereiro

Com a construção do Gráfico 6.4 e da visualização de sua Tabela 6.4 é possível observar o estado da qualidade de forma geral entre as equipes no mês de Fevereiro. Confirmando o que outros gráficos já haviam demonstrado, no mês de fevereiro a equipe de Teste apresenta problemas. Mais uma vez as fases projeto e server são as que apresentam seus resultados de forma positiva.

6.4.1.1 Estado da qualidade dos dados por equipe ao longo do ano de 2007

Tabela 6.12 – Tabela de qualidade total do ano de 2007

Qualidade total	
	qualidade
jan	6,8
fev	7,2
marc	7
abr	7,06
maio	6,78
junho	6,74
julho	6,77
agosto	5,4

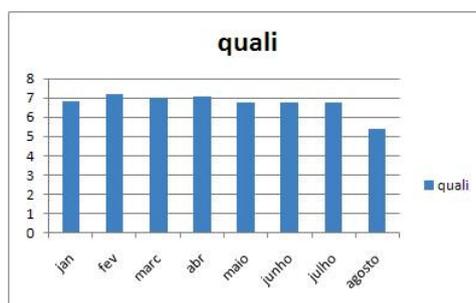


Figura 6.5 – Gráfico de qualidade geral no ano de 2007

A construção do Gráfico 6.5 e a visualização da sua Tabela 6.12 permitem ao gestor uma análise mais genérica observando a evolução da qualidade ao longo do ano. É possível

que seja notado algum mês como problemático e como estratégia para o ano seguinte, observar com maior atenção e cuidado aquele mês para que os dados possuam melhor qualidade e conseqüentemente as decisões tomadas não sejam prejudicadas pela falta de confiabilidade nos dados. Neste gráfico os meses janeiro e agosto podem ser notados como problemáticos, no entanto a falta de dados como pode ser observado também na Tabela 6.13, pode influenciar nessa análise. Entretanto, não é totalmente falso inferir que são meses problemáticos pois a única fase que possuem dados já apresentam uma qualidade não tão boa (6,8 e 5,4 respectivamente). A falta de dados é explicada pela escolha dos indivíduos que fazem parte do escopo deste teste. As fases aqui tidas como 0 (zero) possuem dados de esforço registrados em outras OS e outras versões que não foram escolhidas para fazer parte deste teste pelos motivos inicialmente expostos.

Tabela 6.13 – Tabela de qualidade total do ano de 2007 por dimensões

	Qualidade ao longo do ano							
	jan	fev	março	abril	maio	junho	julho	agosto
projeto	6,8	7,28	7,1	7,13	7,3	6,5	0	0
client	0	7	7,1	7,03	6,84	6,8	7,3	0
server	0	7,2	6,6	7,08	6,55	6,8	7,3	5,4
teste	0	5,4	7,04	7	6,86	6,8	6,7	0

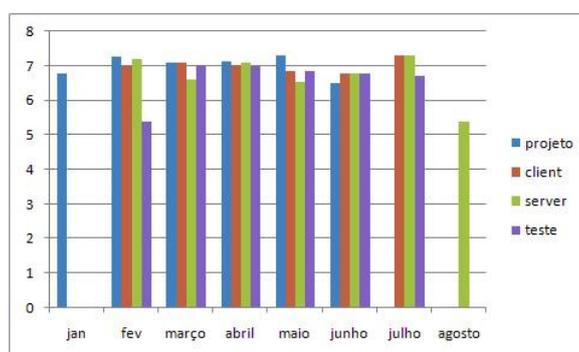


Figura 6.6 – Gráfico de qualidade de equipe no ano de 2007 por dimensões

Já com a construção do Gráfico 6.6 e com a visualização de sua Tabela 6.13 o gestor pode observar as causas de problemas em determinados meses e tomar ações preventivas para que isto não ocorra no ano seguinte.

6.5 Considerações sobre os testes

Foi atingido o objetivo principal dos testes de averiguar se a avaliação atende aos requisitos, simulando testes em um ambiente real e fazendo uso de dados reais. Os resultados

obtidos e as análises realizadas mostraram que a pesquisa apresenta resultados satisfatórios e é promissora. A etapa de preparação dos dados foi uma etapa que demandou tempo e trabalho mas que foi muito benéfica para mostrar a contribuição da pesquisa no processo de tomada de decisão no sentido de estimativas. No próximo e último Capítulo são apresentadas as conclusões e os trabalhos futuros tendo em vista o que foi alcançado até aqui.

7

Conclusão

7.1 Considerações finais

Este trabalho propõe uma arquitetura computacional para dar suporte à avaliação de qualidade de dados referentes à medições de esforço de software. Essa questão foi levantada através de uma análise realizada em um ambiente de desenvolvimento de software real, relatada no Capítulo 3, onde por técnicas estatísticas constatou-se que a qualidade de dados é uma questão em aberto nas organizações que coletam métricas desse tipo. Nesse estudo pôde-se observar que uma vez verificado o problema de qualidade de dados é necessária uma ação para melhorá-los. No entanto, sem uma avaliação adequada, essa ação se torna inócua.

Para verificar a existência de uma solução adequada foi realizado um estudo sobre o estado da arte em avaliação de qualidade de dados, onde foram encontrados vários trabalhos, relatados e comparados no Capítulo 4 de acordo com os requisitos de uma solução adequada definidos no Capítulo 3. Através dessa comparação, foi possível verificar que não há presente na bibliografia atual um trabalho que possa ser considerado como solução para o problema em questão. Os trabalhos são bem interessantes e se esforçam para colaborar com essa tarefa que é tão difícil: avaliar a qualidade dos dados. Contudo, os trabalhos não atendem às reais necessidades de gestão no contexto de desenvolvimento de software. A Tabela 7.1 mostra a comparação final entre os trabalhos relacionados estudados e o trabalho desenvolvido durante esta pesquisa e como pode ser observado, o presente trabalho preenche totalmente todos os requisitos. A Tabela 7.2 mostra de que maneira cada requisito é preenchido.

Tabela 7.1 – Comparação final entre os trabalhos

	Primeiro Grupo						Segundo Grupo			BERARDI (2008)	
	AIQM (2002)	ASM-IQS (2004)	DQA (2002)	DQMIM (2007)	ORME-DQ (2007)	MDB:PA (2007)	PEDI (2004)	Liebchen (2007)	Caro (2007)		Buneman (2006)
Granularidade											
Funcionalidade											
Não-dicotomicidade											
Objetividade											
Contextualização											

A arquitetura proposta como solução é composta por 4 principais componentes: (1) Componente de Proveniência, (2) Componente de Inferência, (3) Componente de Qualidade e (4) Componente de Análise. Os componentes foram construídos com base em uma aquisição de conhecimento realizada em um ambiente operacional utilizando como instrumento de coleta de dados uma entrevista semi-estruturada, cujo formato possibilitou a construção adequada de cada componente buscando atender de forma completa os requisitos da solução, como mostra a Tabela 7.2 .

Requisito	BERARDI 2008
Granularidade	Componente de Proveniência Componente de Análise
Funcionalidade	Composição dos componentes
Não-dicotomicidade	Componente de Inferência (lógica fuzzy)
Objetividade	Componente de Proveniência Componente de Qualidade
Contextualização	Componente de Proveniência Componente de Inferência

Tabela 7.2 – Análise da arquitetura com relação aos requisitos da solução

Para avaliar esta solução foi efetuado um conjunto de testes, descrito no Capítulo 6, seguindo uma seqüência de passos para aceitar ou rejeitar a hipótese inicial de melhoria. Com os resultados foi possível verificar que (i) a contribuição da avaliação com uma das técnicas utilizadas para realização de estimativas através de um dos testes, (ii) a efetividade da utilização de dimensões uma vez que a avaliação contribui com o processo de tomada de decisão através diferentes visões e granularidades que ela permite visualizar, facilitando assim mecanismo de treinamento e conscientização dentro da equipe de projeto, (iii) a eficácia da avaliação através da utilização da lógica fuzzy no componente de inferência

evitando eliminar dados com grau de qualidade não tão alto mas que pode ser útil no processo de tomada de decisão e (iv) o sucesso na composição das dimensões através de proveniência de dados, proposta inovadora deste trabalho.

Essa solução possui uma importante contribuição em relação a trabalhos anteriores de avaliação de qualidade de dados por apresentar uma arquitetura específica e adequada para avaliação de qualidade no contexto de métricas de esforço de software. Essa contribuição foi publicada na forma de pôster em um importante evento na área de *data quality* que acontece no MIT (*Massachusetts Institute of Technology*) em Cambridge, Massachusetts, USA, chamado *International Conference on Information Quality(ICIQ'08)* [Berardi and Ruiz, 2008].

7.2 Trabalhos futuros

A continuidade da pesquisa visa primeiramente implementar os componentes que ainda não foram testados individualmente. Em segundo lugar, a continuidade da pesquisa está em melhorar os componentes ou até mesmo analisar outras propostas para substituir algum deles. Nesse sentido, pretende-se:

- Agregar novas dimensões de qualidade para que atenda outros tipos de problemas que podem ser encontrados e assim agregar novas variáveis lingüísticas e funções de pertinência.
- Estudar outras alternativa para utilização no mecanismo de inferência como redes Bayesianas e redes neurais ou até mesmo sistemas NeuroFuzzy [Altrock, 1995].
- Possibilitar a análise de dados em termo real e assim permitir uma avaliação intermediária. Atualmente a arquitetura considera um período de coleta e após, análise, não permitindo análise de evolução ou involução individual dos dados.
- Integrar a arquitetura ao ambiente SPDW [Becker et al., 2006] desenvolvido no grupo de pesquisa.

Referências

- [Altrock, 1995] Altrock, C. V. (1995). *Fuzzy logic and neurofuzzy applications explained*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.
- [Ana R. Rocha, 2001] Ana R. Rocha, José C. Maldonado, K. C. W. (2001). *Qualidade de software: Teoria e Prática*. Prentice Hall, São Paulo.
- [Andrew Gray, 1997] Andrew Gray, S. M. (1997). Applications of fuzzy logic to software metric models for development effort estimation. In *Proceedings of the 1997 Annual meeting of the North American Fuzzy Information Processing Society (NAFIPS97)*, pages 394–399. IEEE.
- [Angelica Caro, 2007] Angelica Caro, Coral Calero, E. M. M. P. (2007). A probabilistic approach to web portal’s data quality evaluation. In *QUATIC '07: Proceedings of the 6th International Conference on Quality of Information and Communications Technology*, pages 143–153, Washington, DC, USA. IEEE Computer Society.
- [Batini et al., 2007] Batini, C., Barone, D., Mastrella, M., Maurino, A., and Ruffini, C. (2007). A framework and a methodology for data quality assessment and monitoring. In *Proceedings of the 12nd International Conference on Information Quality (ICIQ07)*, pages 85–99. MIT.
- [Becker et al., 2006] Becker, K., Ruiz, D. D., Cunha, V. S., Novello, T. C., and e Souza, F. V. (2006). Spdw: A software development process performance data warehousing environment. In *SEW '06: Proceedings of the 30th Annual IEEE/NASA Software Engineering Workshop*, pages 107–118, Washington, DC, USA. IEEE Computer Society.
- [Berardi and Ruiz, 2008] Berardi, R. and Ruiz, D. (2008). Evaluating data quality of software effort: A data provenance framework based on fuzzy-logic. In *Proceedings of the 13th International Conference on Information Quality*, pages 46–46, Cambridge, Massachusetts, USA. MIT Press.

- [Berardi, 2007] Berardi, R. C. G. (2007). *Qualidade de dados no contexto de Métricas de Software*. Monografia-PUCRS, RS, Porto Alegre.
- [Berry et al., 2004] Berry, M., Jeffery, R., and Aurum, A. (2004). Assessment of software measurement: An information quality study. In *METRICS '04: Proceedings of the Software Metrics, 10th International Symposium*, pages 314–325, Washington, DC, USA. IEEE Computer Society.
- [Buneman et al., 2006] Buneman, P., Chapman, A., and Cheney, J. (2006). Provenance management in curated databases. In *SIGMOD '06: Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pages 539–550, New York, NY, USA. ACM.
- [Buneman et al., 2000] Buneman, P., Khanna, S., and Tan, W.-C. (2000). Data provenance: Some basic issues. *Lecture Notes in Computer Science*, 1974:87–93.
- [Buneman et al., 2001] Buneman, P., Khanna, S., and Tan, W.-C. (2001). Why and where: A characterization of data provenance. *Lecture Notes in Computer Science*, 1973:316–330.
- [Buneman and Tan, 2007] Buneman, P. and Tan, W.-C. (2007). Provenance in databases. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 1171–1173, New York, NY, USA. ACM Press.
- [Caballero et al., 2007] Caballero, I., Verbo, E., Calero, C., and Piattini, M. (2007). A data quality measurement information model based on iso/iec 15939 (research-in-progress). In *Proceedings of the 12nd International Conference on Information Quality (ICIQ07)*, pages 119–141. MIT.
- [Chengalur-Smith et al., 1999] Chengalur-Smith, I. N., Ballou, D. P., and Pazer, H. L. (1999). The impact of data quality information on decision making: An exploratory analysis. *IEEE Transactions on Knowledge and Data Engineering*, 11(6):853–864.
- [Cui and Widom, 2000] Cui, Y. and Widom, J. (2000). Practical lineage tracing in data warehouses. In *ICDE '00: Proceedings of the 16th International Conference on Data Engineering*, page 367, Washington, DC, USA. IEEE Computer Society.
- [Ebert, 1996] Ebert, C. (1996). Fuzzy classification for software criticality analysis. *Expert Systems with Applications*, 11(3):323–342.

- [Fileto et al., 2003] Fileto, R., Medeiros, C. B., Liu, L., Pu, C., and Assad, E. D. (2003). Using domain ontologies to help track data provenance. In *Brazilian Symposium on Databases (SSDB)*, volume 18, pages 84–98.
- [Florac et al., 1997] Florac, W. A., Park, R., and Carleton, A. D. (1997). *Practical Software Measurement: Measuring for Process Management and Improvement*. Carnegie Mellon, New York.
- [Foster et al., 2002] Foster, I. T., Vöckler, J.-S., Wilde, M., and Zhao, Y. (2002). Chimera: Avirtual data system for representing, querying, and automating data derivation. In *SS-DBM '02: Proceedings of the 14th International Conference on Scientific and Statistical Database Management*, pages 37–46, Washington, DC, USA. IEEE Computer Society.
- [Francalanci and Pernici, 2004] Francalanci, C. and Pernici, B. (2004). Data quality assessment from the user’s perspective. In *IQIS '04: Proceedings of the 2004 international workshop on Information quality in information systems*, pages 68–73, New York, NY, USA. ACM.
- [Gopal et al., 2002] Gopal, A., Krishnan, M. S., Mukhopadhyay, T., and Goldenson, D. R. (2002). Measurement programs in software development: Determinants of success. *IEEE Transactions on Software Engineering*, 28(9):863–875.
- [Hulse et al., 2007] Hulse, J. D. V., Khoshgoftaar, T. M., and Huang, H. (2007). The pairwise attribute noise detection algorithm. *Knowledge and Information Systems*, 11(2):171–190.
- [Institute, 2002] Institute, S. E. (2002). *CMMI for Software Engineering (CMMI-SW, V1.1), Staged Representation*. Carnegie Mellon, Pittsburgh.
- [Institute, 2006] Institute, S. E. (2006). *CMMI for Development, Version 1.2*. Carnegie Mellon, Pittsburgh.
- [Jim Arlow, 2005] Jim Arlow, I. N. (2005). *UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design (2nd Edition)*. Addison-Wesley Professional.
- [Kan, 2003] Kan, S. H. (2003). *Metrics and Models in Software Quality Engineering Second Edition*. Project Management Institute Inc., Boston.

- [Kaner and Bond, 2004] Kaner, C. and Bond, W. P. (2004). Software engineering metrics: What do they measure and how do we know. In *In METRICS 2004. IEEE CS*, pages 1–12. Press.
- [Kanungo, 2004] Kanungo, S. (2004). *CMMI implementation : embarking on high maturity practices*. McGraw-Hill.
- [Khoshgoftaar et al., 2001] Khoshgoftaar, T. M., Allen, E. B., Jones, W. D., and Hudepohl, J. P. (2001). Data mining of software development databases. *Software Quality Control*, 9(3):161–176.
- [Kimball and Ross, 2002] Kimball, R. and Ross, M. (2002). *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling (Second Edition)*. John Wiley e Sons, New York, NY.
- [Kulpa and Johnson, 2003] Kulpa, M. K. and Johnson, K. A. (2003). *Interpreting the CMMI: A Process Improvement Approach*. Auerbach Publications, Boca Raton.
- [Lee et al., 2004] Lee, Y. W., Pipino, L., Strong, D. M., and Wang, R. Y. (2004). Process-embedded data integrity. *J. Database Manag.*, 15(1):87–103.
- [Lee et al., 2006] Lee, Y. W., Pipino, L. L., Funk, J. D., and Wang, R. Y. (2006). *Journey to Data Quality*. The MIT Press, London, England.
- [Lee et al., 2002] Lee, Y. W., Strong, D. M., Kahn, B. K., and Wang, R. Y. (2002). Aimq: a methodology for information quality assessment. *Inf. Manage.*, 40(2):133–146.
- [Liebchen et al., 2007] Liebchen, G., Twala, B., Shepperd, M., Cartwright, M., and Stephens, M. (2007). Filtering, robust filtering, polishing: Techniques for addressing quality in software data. In *ESEM '07: Proceedings of the First International Symposium on Empirical Software Engineering and Measurement*, pages 99–106, Washington, DC, USA. IEEE Computer Society.
- [Nicolas Prat, 2008] Nicolas Prat, S. M. (2008). Measuring data believability: A provenance approach. In *HICSS '08: Proceedings of the Proceedings of the 41st Annual Hawaii International Conference on System Sciences*, page 393, Washington, DC, USA. IEEE Computer Society.
- [Olson, 2003] Olson, J. E. (2003). *Data Quality: The Accuracy Dimension*. Elsevier, San Francisco, CA.

- [Palza et al., 2003] Palza, E., Furhman, C., and Abran, A. (2003). Establishing a generic and multidimensional measurement repository in cmmi context. In *Proceedings of the 28th Annual NASA Goddard Software Engineering Workshop (SEW'03)*, pages 12–22, Greenbelt (Maryland, USA). IEEE Computer Society.
- [Pfleeger, 2004] Pfleeger, S. L. (2004). *Engenharia de Software: teoria e prática. 2 ed.* Prentice Hall, São Paulo.
- [Redman, 2001] Redman, T. C. (2001). *Data Quality: The Field Guide*. Digital Press, Boston.
- [Rezende, 2003] Rezende, S. O. (2003). *Sistemas Inteligentes Fundamentos e Aplicações*. Manole, Barueri, SP.
- [Society, 1998] Society, I. C. (1998). *IEEE Std 1061-1998*. The Institute of Electrical and Electronics Engineers, Inc., New York.
- [Sommerville, 2004] Sommerville, I. (2004). *Software Engineering, 7. ed.* Addison Wesley.
- [Strong et al., 1997] Strong, D. M., Lee, Y. W., and Wang, R. Y. (1997). Data quality in context. *Communications of the ACM*, 40(5):103–110.
- [Tan et al., 2006] Tan, P.-N., Steinbach, M., and Kumar, V. (2006). *Introduction to Data Mining*. Addison Wesley.
- [Triola, 1999] Triola, M. F. (1999). *Introdução à Estatística*. Addison Wesley Longman, Inc., Rio de Janeiro.
- [Wand and Wang, 1996] Wand, Y. and Wang, R. Y. (1996). Anchoring data quality dimensions in ontological foundations. *Commun. ACM*, 39(11):86–95.
- [Widom, 2005] Widom, J. (2005). Trio: A system for integrated management of data, accuracy, and lineage. In *CIDR*, pages 262–276.
- [Witold Pedrycz, 1998] Witold Pedrycz, J. F. P. (1998). *Computational Intelligence in Software Engineering*. World Scientific, River Edge, NJ.
- [Witten and Frank, 2005] Witten, I. H. and Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques*. Elsevier, San Francisco, Ca.

- [Yogesh L. Simmhan, 2005] Yogesh L. Simmhan, Beth Plale, D. G. (2005). A survey of data provenance in e-science. *SIGMOD Rec.*, 34(3):31–36.
- [Zubrow, 2001] Zubrow, D. (2001). *The Measurement and Analysis Process Area in CMMI*. ASQ Software Quality Newsletter.

Apêndice A

Resumo da Norma 1061 de 1998

O padrão oferece um framework para estabelecer esse conjunto de métricas IEEE Computer Society (1998).

Primeiramente, neste framework, tem-se a função de estabelecer os requisitos da qualidade através da atribuição de vários atributos, os quais são utilizados para descrever a qualidade do sistema X. Vários fatores de qualidade são estabelecidos em conjunto com a equipe de projeto para que suas definições sejam estabelecidas. Cada fator de qualidade tem associada, ao menos, uma métrica direta, a qual é uma representação quantitativa do fator. Se necessário, subfatores de qualidade são atribuídos para cada fator de qualidade, os quais conteram as métricas coletadas do sistema X. O padrão IEEE 1061, além do framework, também define uma metodologia para as métricas de qualidade de software, para estabelecer os requisitos de qualidade bem como identificar, implementar, analisar e validar as métricas de processo e produto de software. Esta metodologia é composta por cinco passos em seqüência: (1) Estabelecer requisitos de qualidade, (2) Identificar métricas de qualidade, (3) Implementar as métricas (4) Analisar os resultados das métricas e (5) Validar as métricas.

1. Estabelecer os requisitos de qualidade - O resultado desse passo é um conjunto de fatores de qualidade.

(a) Determinar a lista de requisitos, ordenados por importância: Após conversas com os stakeholders, organizar o resultado das mesmas em uma lista de requisitos de qualidade, que podem ser mapeados como fatores de qualidade. Escolher a lista tomando o cuidado para que os requisitos sejam tecnicamente praticáveis, complementares, alcançáveis e ver-

ificáveis. É importante que todas as partes estejam em acordo. Exemplo de um requisito de qualidade mapeado como fator de qualidade é "Alta eficiência".

(b) Quantificar cada fator de qualidade: Para cada fator, atribuir uma ou mais métricas diretas para representar o fator de qualidade e atribuir valores para servir como requisito quantitativo para o fator de qualidade. Por exemplo, para o fator de qualidade "Alta eficiência", as métricas diretas que podem ser atribuídas são "UtilizacaoAtualRecursos" e "UtilizacaoRecursosAlocada", as quais vão representar o fator AltaEficiencia. Um valor associado como requisito quantitativo serve como limite mínimo ou máximo para a análise da métrica.

$$\text{AltaEficiencia} = \text{UtilizacaoAtualRecursos} / \text{UtilizacaoRecursosAlocada}$$

2. Identificar métricas de qualidade - O resultado desse passo é um conjunto de métricas aprovado.

(a) Aplicar o framework de métricas de qualidade: Usando o framework, decompor os subfatores em métricas. Para cada métrica, atribuir um valor como meta e um valor crítico. Após, definir como esses valores devem ser alcançados durante o desenvolvimento do software.

(b) Revisar a aplicação do framework: Use somente métricas validadas, ou seja, aquelas que já passaram pela avaliação estabelecida pelo padrão (essa avaliação é discutida em detalhes no Capítulo 2 deste trabalho). Além disso, usar somente métricas que estão associadas aos requisitos de qualidade do projeto.

3. Implementar as métricas de qualidade - O resultado desse passo é a descrição dos dados, uma matriz de rastreabilidade, um plano de treinamento e cronograma.

(a) Definir o procedimento para coleta dos dados: Para cada métrica no conjunto de métricas, determinar o dado a ser coletado e suposições que serão feitas sobre o dado. Mostrar o fluxo do dado do ponto de coleta até a avaliação da métrica. Identificar as ferramentas e descrever como elas serão usadas. Estabelecer uma matriz de rastreabilidade entre as métricas e os dados. Identificar as entidades organizacionais que participarão da coleta de dados, inclusive aquelas que monitorarão a coleta.

(b) Prototipar um processo de mensuração: Fazer uma análise para determinar se os dados são coletados uniformemente e se as instruções têm sido interpretadas consistente-

mente. Conferir se descrições e instruções sobre os dados coletados estão claras o suficiente para garantir resultados de interpretação uniformes. Usar os resultados coletados para melhorar descrições das métricas e dos dados.

(c) Coletar os dados e calcular os valores das métricas: Coletar e armazenar os dados em uma base de dados do projeto. Examinar o dado quanto à acurácia e à unidade de medida apropriada. Monitorar a coleta de dados. Verificar a uniformidade dos dados se mais de uma pessoa está fazendo a coleta. Calcular as métricas com os dados coletados.

4. Analisar os resultados das métricas- O resultado desse passo é um conjunto de análises sobre valores das métricas que estão, de alguma forma, discrepantes.

(a) Identificar valores de métricas de qualidade que estão fora de um intervalo de tolerância de valores, para estudos posteriores.

(b) Valores inaceitáveis de métricas podem ser complexidade excessiva, documentação inadequada, falta de rastreabilidade ou outros atributos indesejáveis.

(c) Analisar com detalhe os componentes de software que tiverem valores que desviaram dos valores críticos.

5. Validar as métricas - O resultado desse passo é um conjunto de métricas validadas.

(a) Aplicar a metodologia de validação: Quando é possível medir valores de fatores de qualidade em um determinado ponto do ciclo de vida, métricas diretas devem ser usadas para avaliar qualidade do software. Validação não se refere a uma avaliação universal das métricas para todas as aplicações, mas sim em um sentido mais específico, validar a relação entre o conjunto de métricas e o fator de qualidade para uma dada aplicação. O foco da validação está na determinação de onde existe uma relação estatisticamente significativa entre os valores de métricas e valores de fator de qualidade.

(b) Aplicar os critérios de validação. O padrão IEEE 1061 define uma metodologia de validação que é discutida a seguir.

Metodologia de validação de métricas de software da Norma 1061

O padrão IEEE 1061 IEEE Computer Society (1998) define uma metodologia de validação de métricas. Para isso alguns critérios de validação também foram descritos, e para essa avaliação alguns limites (chamados de V, B, A, alfa e P) devem ser previamente estabelecidos:

- V: o quadrado do coeficiente de correlação linear;
- B: rank do coeficiente de correlação;

- A: erro de predição;
- alfa: nível de confiança e
- P: taxa de sucesso.

Para ser considerada válida, uma métrica deve ter um alto grau de associação com o fator de qualidade que esta representa, em conformidade com os critérios abaixo:

- Correlação: a variação no valor do fator de qualidade explicada pela variação no valor da métrica, a qual é dada pela raiz quadrada do coeficiente de correlação linear (R^2) entre a métrica e o valor do fator de qualidade correspondente, seu valor deve ser maior do que V . Esse critério avalia se há uma associação suficientemente forte entre a métrica e seu correspondente fator de qualidade;
- Rastreabilidade: se uma métrica M é diretamente relacionada a um fator de qualidade Q , para determinado produto ou processo, então uma mudança nos valores de Q em um tempo T deve ser acompanhada por uma mudança no valor da métrica M no mesmo T . Essa mudança deve ser no mesmo sentido, ou seja, a medida que Q cresce, M também cresce, ou o contrário. O coeficiente de "rank de correlação" r é calculado para n pares (métrica/fator). Cada par deve ser medido no mesmo tempo. Esse critério avalia se uma métrica está apta a acompanhar as mudanças de qualidade de processo ou produto ao longo do ciclo de vida;
- Consistência: se os valores do fator qualidade F_1, F_2, F_n , correspondendo a produtos e processos $1, 2, n$, tem um relacionamento $F_1 > F_2 > F_n$ então o valor de métrica também deve ter uma relação $M_1 > M_2 > M_n$. Este critério mede o grau de consistência do valor do fator de qualidade de um conjunto de componentes de software e, o grau da qualidade do valor da métrica para os mesmos componentes. Isso determina se uma métrica possui um alto grau de qualidade de precisão
- Predição: se uma métrica é usada em um instante T_1 para prever um fator de qualidade de dado produto ou processo, ele deve prever um fator de qualidade F_{pT_2} com uma precisão de $|F_{aT_2} - F_{pT_2} / F_{aT_2}|$, onde F_{aT_2} é o valor real de F no instante T_2 . Esse critério avalia se a métrica é capaz de prever um valor de fator de qualidade com a acurácia desejada;

- Poder discriminativo: uma métrica deve estar apta para discriminar um componente de software de alta qualidade de outro de baixa qualidade. Assim, é possível identificar componentes de software com uma qualidade inaceitável. Este valor deve ser acima do valor estabelecido para alfa. Esse critério avalia se a métrica é capaz de separar um conjunto de componentes de software de alta qualidade de um conjunto de componentes de baixa qualidade; e
- Confiabilidade: uma métrica deve ter as propriedades de correlação, rastreabilidade, consistência, previsibilidade e poder discriminativo para pelo menos P% das aplicações das métricas. Este critério garante que a métrica tenha passado por todo o procedimento de validação com um número suficiente de testes ou porcentagem de aplicações de modo que a métrica possa desempenhar a função para a qual foi adotada.

Procedimento de validação:

1. Identificar uma amostra de fatores de qualidade;

- Identificar amostra do mesmo domínio (e.g. mesmos componentes de software), obtidos do repositório de métricas;
- Executar as análises dos critérios acima descritos;
- Documentar os resultados das análises;
- Revalidar as métricas (executar todos o procedimento novamente).
- Avaliar a estabilidade do ambiente validando as métricas em um outro ambiente de desenvolvimento com as mesmas características do ambiente, inicialmente, implantadas as métricas.

Se uma métrica não passa em todos os critérios de validação, ela deve ser usada somente de acordo com o critério que ela passou. Por exemplo, se uma métrica é considerada válida quanto o critério Rastreabilidade, ela deve ser usada somente para acompanhar a evolução (ou involução) do processo ou produto em questão.

É importante ressaltar que a validação das métricas deve ser aplicado em ambientes estáveis de desenvolvimento, ou seja, onde não muda com frequência a linguagem de implementação, as ferramentas de desenvolvimento e outras especificidades do projeto. Além

disso, a aplicação e validação de métricas deve ser implementada durante as mesmas fases do ciclo de vida em diferentes projetos. Por exemplo, se uma métrica A é coletada durante a fase de design do projeto P e tem relação com o fator de qualidade F, então a métrica B seria coletada durante a fase de design do projeto Q e teria relação com o fator de qualidade F.

Apêndice B

Entrevista semi-estruturada

1. Entidade aplicadora da entrevista

A entrevista está inserida no contexto do Grupo de Pesquisa em Inteligência de Negócios sob coordenação do Prof. Dr. Duncan Dubugras Alcoba Ruiz.

O Grupo de Pesquisa está localizado na sala 628, 6° andar do prédio 32 no campus da PUCRS. Qualquer informação sobre a entrevista pode ser obtida com o coordenador:

E-mail: duncan@pucrs.br

Telefone: (0xx51) 3320 3611

Fax: (0xx51) 3320 3621

2. Objetivos

2.1. Tema

De acordo com o CMMI (SEI) as métricas de software auxiliam à organização no acompanhamento da saúde dos seus processos e projetos. Essas métricas são utilizadas no gerenciamento quantitativo e, conseqüentemente, no processo de tomada de decisão. Tendo em vista o papel que as métricas exercem no acompanhamento dos processos, é de grande importância que os dados dessas métricas ofereçam um grau de confiança para quem as utiliza. O tema desta entrevista está embasado nesses fatos, no entanto a tarefa de avaliar a qualidade de dados de métricas de software não é uma tarefa fácil, pois esse tipo de dado tem origem subjetiva e portanto sua avaliação também é altamente dependente dos critérios que cada usuário possui. De acordo com Pipino et.al. a qualidade de dados é avaliada de acordo com um número de dimensões, tais como acurácia, atualidade, completude e consistência. Além disso, os autores enfatizam que cada organização deve determinar quais dessas dimensões são importantes e ainda, definir variáveis que constituem essas dimensões.

2.2. Objetivo Geral

O principal objetivo desta entrevista é a obtenção de informações de pessoas que estão diretamente envolvidas na coleta e utilização das métricas de esforço de PDS. Através dessas informações busca-se identificar critérios que são utilizados no dia-a-dia para a avaliação subjetiva de qualidade desses dados, as quais constituem as dimensões mencionadas acima e as variáveis que compõem cada dimensão. O conjunto de dimensões é base para modelagem de regras que definem um grau de qualidade para dados de esforço.

2.3. Objetivos Específicos De acordo com Sellitz citado por Marconi et.al. o método de pesquisa "entrevista" possui seis tipos de objetivos: averiguação de fatos; determinação das opiniões sobre os fatos; determinação de sentimentos, descoberta de planos de ação; conduta atual ou do passado e motivos conscientes para opiniões, sentimentos, sistemas e condutas. No entanto, nesta entrevista existem alguns objetivos que são mais focalizados de acordo com o tema e com o objetivo geral descrito anteriormente. São eles:

- Determinação das opiniões sobre os fatos: Conhecer o que os entrevistados pensam sobre os fatos, que neste tema, são as avaliações de qualidade de dados.
- Descoberta de planos de ação: Descobrir, por meio das definições individuais dadas,

qual a posição adotada em determinadas situações, identificando os critérios para as ações tomadas, no caso, com relação à avaliação da qualidade de dados de esforço.

- Motivos inconscientes para opiniões, sentimentos e conduta e por quê: Descobrir quais fatores podem influenciar as opiniões, e qual o grau de importância de cada um desses fatores para a qualidade dos dados.

3. Justificativa da técnica de pesquisa

A técnica de entrevista possui algumas vantagens que estão de acordo com os objetivos propostos. Segundo Marconi, a entrevista é um encontro entre duas pessoas, com o intuito de que uma delas obtenha informações a respeito de um determinado assunto através de um procedimento de investigação para coleta de dados. Essa técnica fornece uma amostragem melhor da população alvo e além disso, possibilita maior flexibilidade, pois o entrevistador pode repetir ou esclarecer perguntas ou ainda, modificar a forma da questão para prover maior entendimento do que está sendo questionado.

O tipo de resposta esperada pelo entrevistado, no caso de qualidade de dados, dificulta a utilização de outra técnica de pesquisa que prevê um conjunto de perguntas rígidas e sem possibilidade de interferência do questionador. Com os propósitos do tema, uma resposta do entrevistado pode fazer surgir outra pergunta que não foi previamente definida.

Essa técnica possui algumas limitações como dificuldade de expressão e comunicação entre as partes (entrevistador e entrevistado); possibilidade de o entrevistado ser influenciado; disposição do entrevistado em dar as informações necessárias, entre outras. No entanto, frente às vantagens acredita-se que um bom planejamento e cuidado no momento da entrevista podem amenizar eventuais problemas que são inerentes à técnica.

4. Conhecimento do entrevistado

As informações referentes ao entrevistado visam a verificação do grau de familiaridade dele com o tema e com os objetivos da entrevista. Para melhor abrangência de opiniões e conhecimentos sobre qualidade de dados, o grupo de entrevistados serão de três tipos de perfis. De acordo com Pipino et. al.:

- Membro de equipe de desenvolvimento (quem registra o dado)
- Responsável por coleta e armazenamento (responsável pelo dado)
- Gestor (gerentes, líderes)

4.1. Confidencialidade das respostas

A coordenação da pesquisa garante a confidencialidade dessas informações. O preenchimento dos dados pessoais é de livre opção do entrevistado, no entanto para a interpretação mais eficaz dos dados é importante as informações relacionadas à experiência e o tipo de contato o entrevistado mantém com os dados das métricas de esforço.

4.2. Motivação ao entrevistado

A participação do entrevistado colabora com a captura de conhecimentos obtidos com a experiência e vivência com a coleta de métricas. Essa captura contribui com o maior entendimento do porque que alguns dados são considerados duvidosos ou inconfiáveis. Cada perfil de usuário contribui na sua área como é discutido abaixo:

- Membro de equipe de desenvolvimento (quem registra o dado): Com as declarações sobre o que é importante e o que influencia na qualidade de um dado auxilia aos outros perfis a entenderem por que às vezes um dado correto, aparentemente parece conter problemas. A falta de conhecimento da realidade de outros pontos de vista pode prejudicar o entendimento do que realmente caracteriza um dado de qualidade.
- Responsável por coleta e armazenamento (responsável pelo dado): Com as declarações do ponto de vista de quem é responsável pela coleta, correção e adaptação podem possibilitar um esclarecimento de por que às vezes um dado mudou o comportamento em um ponto no seu ciclo.
- Gestor (gerentes, líderes): O gestor é um dos maiores interessados para que os dados da organização tenham alto grau de qualidade, pois de acordo com suas decisões a organização pode ter sucesso ou não. Se a decisão é tomada embasada em dados de baixa qualidade, as decisões também o serão.

5. Metodologia

Para atingir os objetivos propostos com a entrevista, algumas etapas são executadas:

Planejamento das perguntas a serem realizadas

Para uma organização e otimização do tempo disponível do entrevistado, é preparado uma seqüência de perguntas a serem realizadas na entrevista. Devido o tipo de entrevista escolhido, como justificado anteriormente, as definição prévia das perguntas não impede

que no momento da entrevista outras sejam formuladas para enriquecer o entendimento dos fatos buscados.

Definição do(s) perfil(is) dos entrevistados

De acordo com Pipino et al , para uma organização obter uma avaliação produtiva sobre a qualidade de seus dados é importante que fatores que influenciam nessa qualidade estejam claros para os três tipos de perfis envolvidos na obtenção, coleta e utilização dos dados. Entendeu-se que a melhor abrangência dos fatores de qualidade será atingida com a entrevista aos membros de equipes, os responsáveis pelos dados e os gestores que se utilizam desses dados no gerenciamento quantitativo.

Pré-teste

Realizar uma entrevista com alguém que não fará parte da pesquisa final pode acrescentar na qualidade da entrevista efetiva, fazendo críticas à postura do entrevistador, às questões abordadas ou outros aspectos da entrevista.

Refinamento

Realizar melhorias e adaptações de acordo com as opiniões e sugestões registradas no pré-teste.

Agendamento das entrevistas

O agendamento prévio das entrevistas fica incumbido a um facilitador na organização onde a entrevista acontecerá. O agendamento deve reservar no mínimo 60 minutos de disponibilidade do entrevistado. Esta etapa é importante na motivação do entrevistado em participar da entrevista. O facilitador pode disponibilizar o documento de proposta para que o entrevistado seja contextualizado sobre o tema, objetivos e importância da pesquisa.

Relatório de análise

Ao final das entrevistas será elaborado um relatório com as conclusões e interpretação dos dados respondidos. O ideal é que neste relatório estejam identificadas as variáveis que constituem cada dimensão de qualidade e que tanto os fatores quanto as dimensões esteja apropriadamente claras para a avaliação de qualidade dos dados de métricas de esforço.

6. Referências bibliográficas

Software Engineering Institute. (2006). CMMI for Development, Version 1.2. Pittsburgh, PE: Carnegie Mellon.

Lee, Y. W., Pipino, L. L., Funk, J. D., and Wang, R. Y. (2006). *Journey to Data Quality*. The MIT Press.

Marconi, M., Lakatos, E. 2003 *Fundamentos de Metodologia Científica*. Editora Atlas S. A. 5 edição.

7. Entrevista

A entrevista consiste em 3 principais partes. A primeira parte consiste no conhecimento do perfil do entrevistado. A segunda parte busca a definição da qualificação das dimensões para diferentes níveis de qualidade de dados de esforço. Na terceira parte são discutidas as variáveis que classificam cada uma das dimensões definidas na parte anterior. Na entrevista realizada no pré-teste existirá a parte iv onde serão registradas sugestões pelos entrevistados.

O formato das respostas esperadas nas partes 2 e 3 são de causa conseqüência. Se determinados aspectos acontecem, então se julga o dado com determinada qualidade.

PARTE I - CONHECIMENTO DO PERFIL DO ENTREVISTADO -

1. Qual o seu nome?

2. Qual o seu papel com relação aos dados de esforço da organização onde você trabalha?

() Membro de equipe de desenvolvimento (quem registra o dado)
() Responsável por coleta e armazenamento (responsável pelo dado)
() Gestor (gerentes, líderes)
3. O que você entende por qualidade de dados de métricas de esforço de PDS?

4. É importante a avaliação de qualidade de dados de métricas de esforço de PDS? Por quê?

5. Você concorda que algumas informações referentes à proveniência do dado podem contribuir com a avaliação de qualidade de dados de esforço de PDS? Por quê?

6. Você concorda que suas respostas sejam utilizadas na pesquisa de mestrado da aluna?
()sim ()não

PARTE I I - IDENTIFICAÇÃO DAS DIMENSÕES DE QUALIDADE -

Definição de cada dimensão de acordo com Pipino et. Al.(2006)

Acurácia: diz respeito ao quanto o dado está correto.

Atualidade: diz respeito a quão atualizado está o dado para o que ele vai ser analisado.

Completude: diz respeito à falta de informação, valores faltantes.

Consistência: diz respeito ao sentido do dado ao longo do ciclo.

De acordo com as definições de cada dimensão, como você caracteriza um dado de:

- Muito baixa qualidade

Acurácia ()Baixa ()Média ()Alta

Atualidade ()Baixa ()Média ()Alta

Completude ()Baixa ()Média ()Alta

Consistência ()Baixa ()Média ()Alta

- Razoável qualidade

Acurácia ()Baixa ()Média ()Alta

Atualidade ()Baixa ()Média ()Alta

Completude ()Baixa ()Média ()Alta

Consistência ()Baixa ()Média ()Alta

- Boa qualidade

Acurácia ()Baixa ()Média ()Alta

Atualidade ()Baixa ()Média ()Alta

Completude ()Baixa ()Média ()Alta

Consistência ()Baixa ()Média ()Alta

- Ótima qualidade

Acurácia ()Baixa ()Média ()Alta

Atualidade ()Baixa ()Média ()Alta

Completude ()Baixa ()Média ()Alta

Consistência ()Baixa ()Média ()Alta

2. Que outras dimensões você acha que é importante na avaliação de qualidade de dados de métricas de esforço de PDS?

PARTE I I I - IDENTIFICAÇÃO DAS VARIÁVEIS DAS DIMENSÕES -

Nesta parte são identificadas as variáveis que compõem cada dimensão relatada na parte anterior.

1. Que variáveis de proveniência você acredita que interferem de alguma maneira na acurácia?

Acurácia

- ()o valor original lançado
- ()o dia do registro (dia escolhido para lançar a hora)
- ()dia do lançamento (dia atual)
- ()fase de lançamento
- ()fonte de lançamento (ferramentas de gerência)
- ()valor modificado (por transformações, adaptações,correções)
- ()diferença do valor original com um valor esperado(discrepância / quanto?)
- ()diferença do valor original com o valor modificado(discrepância)
- ()experiência
- ()outro. qual?

Atualidade

- ()o valor original lançado
- ()o dia do registro (dia escolhido para lançar a hora)
- ()dia do lançamento (dia atual)
- ()fase de lançamento
- ()fonte de lançamento (ferramentas de gerência)
- ()valor modificado (por transformações, adaptações,correções)

- diferença do valor original com um valor esperado(discrepância / quanto?)
- diferença do valor original com o valor modificado(discrepância)
- experiência
- outro. qual?

Completude

- o valor original lançado
- o dia do registro (dia escolhido para lançar a hora)
- dia do lançamento (dia atual)
- fase de lançamento
- fonte de lançamento (ferramentas de gerência)
- valor modificado (por transformações, adaptações, correções)
- diferença do valor original com um valor esperado(discrepância / quanto?)
- diferença do valor original com o valor modificado(discrepância)
- experiência
- outro. qual?

Consistência

- o valor original lançado
- o dia do registro (dia escolhido para lançar a hora)
- dia do lançamento (dia atual)
- fase de lançamento
- fonte de lançamento (ferramentas de gerência)
- valor modificado (por transformações, adaptações, correções)
- diferença do valor original com um valor esperado(discrepância / quanto?)
- diferença do valor original com o valor modificado(discrepância)
- experiência
- outro. qual?

PARTE IV - OBSERVAÇÕES DE PRÉ-TESTE -

Esta parte relata as experiências no pré-teste. Pontos positivos e relativos, de melhora ou adaptação levantados pelos entrevistadores no pré-teste.

Você gostaria de mencionar pontos de melhoria, sugestionando como melhorar?

Apêndice C

Modelos preditivos

Modelo Preditivo 1

M5 pruned model tree: (using smoothed linear models)

Total-Lancado ≤ 110.32 : LM1 (31/13.494)

Total-Lancado > 110.32 : LM2 (9/21.85)

LM num: 1

Total Estimado =

0.9434 * tamanho

+ 0.1725 * fp_ajustados

+ 1.1869 * N-Doc

+ 0.9072 * Total-Lancado

- 20.1335

LM num: 2

Total Estimado =

1.7974 * tamanho

+ 0.8332 * fp_ajustados

+ 2.2749 * N-Doc

+ 1.0912 * Total-Lancado

- 136.0446

Modelo Preditivo 2

M5 pruned model tree: (using smoothed linear models)

Total-Avaliado ≤ 112.44 : LM1 (31/13.736

Total-Avaliado > 112.44 : LM2 (9/21.673

LM num: 1

Total Estimado =

0.9724 * tamanho

+ 0.1743 * fp_ajustados

+ 1.1591 * N-Doc

+ 0.886 * Total-Avaliado

- 18.517

LM num: 2

Total Estimado =

1.6737 * tamanho

+ 0.8093 * fp_ajustados

+ 2.2215 * N-Doc

+ 1.0948 * Total-Avaliado

- 128.9232

Modelo Preditivo 3

M5 pruned model tree: (using smoothed linear models) LM1 (40/17.52

LM num: 1

Total Estimado =

0.7118 * tamanho

+ 34.4075 * fp_ajustados=11.66,33.92,25.44,12.72,184.44,31.8,22.26,152.64,74.2,
48.76,171.72

+ 323.1045 * fp_ajustados=48.76,171.72

+ 5.2307 * N-Doc

+ 0.3152 * Total-Avaliado-preparado

- 2.8225