

Revista da Graduação

Vol. 5

No. 1

2012

19

Seção: Faculdade de Informática

UMA ABORDAGEM DE DATA WAREHOUSE PARA GESTÃO DE MÉTRICAS DE SOFTWARE COM ANÁLISE DE VALOR AGREGADO

Graziele Marques Mazuco

Este trabalho está publicado na Revista da Graduação.

ISSN 1983-1374

<http://revistaseletronicas.pucrs.br/ojs/index.php/graduacao/article/view/11416>

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
FACULDADE DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

GRAZIELE MARQUES MAZUCO

**UMA ABORDAGEM DE *DATA WAREHOUSE* PARA GESTÃO DE MÉTRICAS DE
SOFTWARE COM ANÁLISE DE VALOR AGREGADO**

Porto Alegre

2011

GRAZIELE MARQUES MAZUCO

**UMA ABORDAGEM DE *DATA WAREHOUSE* PARA GESTÃO DE MÉTRICAS DE
SOFTWARE COM ANÁLISE DE VALOR AGREGADO**

Trabalho de conclusão de curso de
bacharelado em Ciência da Computação da
Faculdade de Informática da Pontifícia
Universidade Católica do Rio Grande do Sul,
como requisito parcial para obtenção do grau.

Orientador: Prof. Dr. Duncan Dubugras Alcoba Ruiz

Porto Alegre

2011

DEDICATÓRIA

*Dedico esta conquista à minha família e ao meu noivo
especialmente à minha mãe, que sorriu quando me viu sorrir
chorou quando me viu chorar
entendeu meus medos, rezou por mim
mas acima de tudo, acreditou.*

AGRADECIMENTOS

Agradeço primeiramente a Deus, por ter me proporcionado as oportunidades que me trouxeram até aqui e também por ter me dado força e sabedoria para aproveitar estas oportunidades.

Agradeço a minha família, por acreditar e estar sempre ao meu lado principalmente quando eu não acreditava que seria capaz de dar o próximo passo. Por terem me dado coragem para trilhar meu caminho sozinha, mas sempre por perto quando eu precisei de apoio. Lisiane, Jocelir, Orecy, Adalberto, Lauriane, João Leandro, obrigada por serem meu alicerce. Especialmente ao grande culpado, aquele que deu um computador apenas com MS-DOS para uma menina de 11 anos para ela brincar, recompensando o interesse e a vontade de aprender com upgrades e novos joguinhos, obrigada Mano! Ainda tenho guardada a folha em que anotei os primeiros comandos que aprendi.

Agradeço ao meu noivo, meu amor, não encontro palavras para expressar o quanto o teu amor é essencial para mim. Sempre me ajudando a ver as coisas pelo lado racional, colocando meus pés no chão quando eu achava que nada daria certo. Tua dedicação e carinho foram fundamentais para que eu completasse esta etapa.

Ao meu orientador, professor Duncan Ruiz, por toda dedicação, ensinamentos, conselhos e incentivo. Tu és mais que um mestre, és um paizão, por ouvir todas as minhas lamentações e minhas paranóias de que nada estava bom e que não conseguiria terminar a tempo. Obrigada por ter acreditado na minha capacidade.

Às minhas amigas Duda e Daiane. À Duda pelas revisões gramaticais deste trabalho, e às duas por todos os trabalhos que fizemos juntas, pelas horas estudando, pelas bobagens ditas, pela amizade construída.

Obrigada aos meus colegas de trabalho Anderson e Marcelo pelo suporte técnico do Oracle, pela paciência.

É impossível citar todos que merecem um agradecimento aqui, mas obrigada a todos que colaboraram de alguma forma com esta etapa que chega ao fim.

EPÍGRAFE

*"Que nossos esforços desafiem as impossibilidades.
Lembrai-vos que as grandes proezas da história
foram conquistadas do que parecia impossível."
(Charles Chaplin)*

RESUMO

A busca pela qualidade é constante entre as empresas de desenvolvimento de *software*. A cada dia as empresas buscam aprimorar o seu processo de desenvolvimento de *software*, a fim de entregar produtos com mais qualidade, dentro do prazo e orçamento pré-definidos. Para isso, as organizações estão adotando modelos de referência de *software* tais como o *Capability Maturity Model Integration* (CMMI).

Uma das exigências para atender a este modelo de qualidade é a utilização de métricas para mensurar a qualidade dos seus processos. As métricas devem ser coletadas, consolidadas e armazenadas em um repositório implementado na forma de um *Data Warehouse* (DW). O processo de Extração, Transformação e Carga (ETC) é uma das tarefas mais dispendiosas durante a criação de um ambiente de *Data Warehousing*.

Este trabalho descreve um ambiente de *Data Warehousing* como solução para a gestão de métricas de *software*. A solução proposta apresenta um modelo analítico que fornece suporte à análise e monitoração, utilizando a técnica de *Earned Value Analysis* (EVA) para monitorar custos e prazos dos projetos. Os principais componentes da solução proposta são especificados, implementados e testados.

Os benefícios desta solução são: i) oferecer suporte à monitoração, permitindo a realização de cargas freqüentes e incrementais; ii) ser capaz de desonerar a tarefa de capturar os dados de maneira manual, diminuindo assim a chance de erros; e iii) garantir a consistência dos dados no repositório de métricas, possibilitando que dados de diferentes projetos possam ser comparados.

ABSTRACT

The search for quality is constant among the software development companies. Each day the companies look to improve their progress on software development, delivering quality products on time and pre-defined budget. For these, the companies have been adapting software references models like CMMI.

One need to attend this quality model is the use of metrics to measure the quality of their processes. These metrics should be collected, consolidated and stored in a repository. This repository can be implemented as a DW. The metrics process of extraction, transformations and ETC to be stored in the repository. These actions are the hardest during the establishment of the DW ambient.

This work describes a DW ambient as a solution to software metrics management. The solution proposed to introduce one analytic model of support to analysis and monitor as the EVA technique, responsible for monitoring costs and the time project. The main solution components are described, implemented and tested.

This solution has as benefits: i) give support to monitor, enable the execution of frequents and incremental load; ii) dispense the manual capture of the data, less errors chances; and iii) enable consistent information in the metrics repository to data of the different projects can be comparable.

LISTA DE FIGURAS

Figura 2.1 - Identificação das Variações no Gráfico de EVA.....	20
Figura 2.2 - Componentes do CMMI na Representação em Estágios..	22
Figura 2.3 - Componentes do CMMI na representação contínua.....	24
Figura 2.4 - Elementos Básicos de um DW	29
Figura 2.5 - Metodologia de um Projeto de Data Warehouse.....	33
Figura 3.1 - Organograma da organização	36
Figura 3.2 -Tipos de projetos definidos no OSSP da organização	39
Figura 3.3 - Ciclo de Vida dos Projetos	39
Figura 3.4 - Cronograma dos Projetos - Tarefas	40
Figura 3.5 - Cronograma dos Projetos - Componentes.....	40
Figura 3.6 – Acompanhamento do Projeto	42
Figura 3.7 – Planilha de Acompanhamento	43
Figura 4.1 - Modelo E-10 para Estimativas de Esforço	46
Figura 4.2 - Arquitetura Macro do Sistema.....	47
Figura 4.3 - Modelo Analítico do SPDW+	48
Figura 4.4 - Arquitetura do SPDW+	49
Figura 5.1 - Modelo Analítico.....	58
Figura 5.2 – Arquitetura do Sistema.....	61
Figura 5.3 – Estrutura da DSA.	62
Figura 6.1 – Fato Componente no AWM.....	68
Figura 6.2 – Fato Tarefa no AWM.....	69
Figura 6.3 – Fato Componente em 04/10/2011.....	70
Figura 6.4 – Fato Componente em 11/10/2011.....	71

LISTA DE TABELAS

Tabela 2.1 - Níveis de Maturidade CMMI - Representação em Estágios	22
Tabela 2.2 - Níveis de Competência CMMI - Representação Contínua	24
Tabela 2.3 - PAs e Respectivos Níveis de Maturidade	25
Tabela 3.1 - Plano de Métricas Atual	44
Tabela 5.1 - Programa de Métricas Proposto.....	54
Tabela 5.2 - Cronograma de um Projeto Fictício.....	55
Tabela 5.3 - Exemplo de Cronograma Monitorado com EVA.....	56
Tabela 5.4 - Fatos do Modelo Analítico.....	58
Tabela 5.5 - Dimensões do Modelo Analítico.....	58
Tabela 6.1 – Ajuste Métricas Sobre Prazo	67

LISTA DE SIGLAS

BI	<i>Business Intelligence</i>
CBO	Custo do <i>Baseline</i> Original
CBR	Custo do <i>Baseline</i> Revisado
CR	Custo Real
CMMI	<i>Capability Maturity Model Integration</i>
CR	Custo Real
DAT	<i>Data Access Tools</i>
DFBO	Data Final do <i>Baseline</i> Original
DFBR	Data Final do <i>Baseline</i> Revisado
DFR	Data Final Real
DGP	Desenvolvimento de Grande Porte
DIBO	Data Inicial do <i>Baseline</i> Original
DIBR	Data Inicial do <i>Baseline</i> Revisado
DIR	Data Inicial Real
DPA	<i>Data Presentation Area</i>
DPP	Desenvolvimento de Pequeno Porte
DS	Data de <i>Status</i>
DSA	<i>Data Staging Area</i>
DW	<i>Data Warehouse</i>
ETC	Extração Transformação e Carga
EVA	<i>Earned Value Analysis</i>
IDC	Índice de Desempenho de Custo
IDP	Índice de Desempenho de Prazo
MGP	Manutenção de Grande Porte
MPP	Manutenção de Pequeno Porte
MPSBr	Melhoria de Processos do Software Brasileiro
OLAP	<i>On-Line Analytical Process</i>
OSSP	<i>Organization's Set of Standard Process</i>
PA	<i>Process Area</i>
PDS	Processo de Desenvolvimento de Software
SC	Solicitação do Cliente
SG	<i>Specific Goal</i>

SP	<i>Specific Practice</i>
SQL	<i>Structured Query Language</i>
VA	Valor Agregado
VC	Variação de Custo
VCA	Variação de Custo Agregada
VCBO	Variação de Custo do <i>Baseline</i> Original
VCBR	Variação de Custo do <i>Baseline</i> Revisado
VE	Valor Estimado
VP	Variação de Prazo
VPA	Variação de Prazo Agregada
VPBO	Variação de Prazo do <i>Baseline</i> Original
VPBR	Variação de Prazo do <i>Baseline</i> Revisado

SUMÁRIO

1 Introdução	13
1.1 Motivação.....	13
1.2 Objetivos	14
1.3 Organização do Texto	14
2 Fundamentação Teórica	16
2.1 Métricas de Software.....	16
2.1.1 Programa de métricas	17
2.1.2 <i>Earned Value Analysis</i> (EVA)	17
2.2 Modelos de Maturidade de Software	21
2.2.1 CMMI	21
2.2.2 Process Areas (PAs)	25
2.2.2.1 Medição e Análise (MA)	26
2.2.2.2. Monitoração e Controle de Projetos (PMC)	27
2.2.2.3 Gerenciamento Quantitativo de Projetos (QPM)	27
2.3 Ambiente de Data Warehousing.....	28
2.3.1 Elementos de um ambiente de Data Warehousing	29
2.3.2 Modelagem Dimensional	30
2.3.3 Camada de Apresentação	31
2.3.4 OLAP	32
2.3.5 Metodologia do Desenvolvimento	33
2.4 Business Intelligence.....	35
3. Estudo de caso	36
3.1 Processo de desenvolvimento.....	36
3.2 Análise e Monitoração dos Projetos	41
3.2.1 Plano de métricas atual	43
3.3 Considerações Sobre o Estudo de Caso.....	44
4. Trabalhos Relacionados	45
4.1 Tenório [TEN10]	45
4.2 Rezende [REZ09].....	47

4.3 Silveira [SIL07]	47
4.4 Novello [NOV06].....	50
4.5 Considerações Sobre os Trabalhos Relacionados.....	51
5. Solução Proposta	52
5.1 Programa de Métricas	52
5.1.1 Programa de métricas com EVA	52
5.1.2 Monitoração de Custo e Prazo segundo EVA.....	55
5.2 Modelo Analítico.....	56
5.2.1 Questões de Negócio abordadas.....	56
5.2.2 Modelagem Dimensional	57
5.2.3 Validade dos Fatos.....	59
5.3 Processo de Extração Transformação e Carga (ETC)	60
5.3.1 Rotina de Extração	60
5.3.2 Data Staging Area (DSA).....	62
5.3.2.1 Estrutura da DSA.....	62
5.3.2.2 Rotinas de Limpeza e Transformação	63
5.3.2.3 Carga do Repositório de dados	63
5.4 Considerações Sobre a Solução Proposta.....	64
6. Teste da Solução.....	65
6.1 Objetivos	65
6.2 Descrição dos recursos computacionais utilizados	65
6.2.1 Oracle Analytic Workspace Manager (AWM)	66
6.3 Descrição dos dados utilizados no testes.....	66
6.4 Cubo OLAP	67
6.5 Teste de Monitoramento e Análise.....	69
6.6 Considerações Sobre os Testes	71
7. Considerações Finais	72
7.1 Trabalhos Futuros	72
Referências.....	74

1 INTRODUÇÃO

A busca pela qualidade é constante entre as empresas de desenvolvimento de *software*. A cada dia as empresas buscam aprimorar o seu processo de desenvolvimento de *software*, entregando produtos de qualidade, dentro do prazo e orçamento pré-definidos. Para isso, as organizações vêm adotando modelos de referência de *software* como o *Capability Maturity Model Integration* (CMMI) [SEI10] e a Melhoria de Processos do Software Brasileiro (MpsBr) [SOF09] visando certificar seus produtos e serviços.

1.1 MOTIVAÇÃO

Segundo Khoshgoftaarr [KAJ01 apud BER09], a qualidade do produto está diretamente relacionada à qualidade do processo pelo qual este é desenvolvido. Neste contexto, a qualidade do produto depende da melhoria contínua do processo de desenvolvimento, alcançada através do uso de modelos de maturidade que guiam as organizações no controle e na evolução de seus processos [SEI10].

A recente *Pesquisa de Qualidade no Setor de Software Brasileiro* realizada pelo Ministério da Ciência e Tecnologia [MCT11], revelou que há uma crescente preocupação por parte das organizações que contratam serviços de *software*, sobre a qualidade do sistema que estão comprando. Segundo a pesquisa, cerca de 60,6% das organizações entrevistadas exigem que a organização de *software* a ser contratada tenha algum tipo de certificação. Destas, 16,6% exigem que a certificação seja CMMI.

Os modelos de maturidade são um conjunto de boas práticas que, quando bem executadas, definem o grau de maturidade da organização. Para medir a qualidade do Processo de Desenvolvimento de Software (PDS), os modelos de referência indicam a coleta de métricas como meio de controle e análise dos seus processos. No contexto de organizações que desenvolvem *software*, podem ser armazenadas grandes quantidades de dados referentes às medições do processo de desenvolvimento. É necessário ter formas de analisar estes dados a fim de que possam fornecer informações úteis aos gestores, por exemplo, nas atividades de análise e monitoração das etapas de um projeto [SIL07].

Segundo Silveira [SIL07], as atividades de análise consistem no estudo sobre o histórico de métricas através de técnicas estatísticas, mineração de dados, geração de gráficos e verificação de indicadores de desempenho, tendo em vista a utilização dos seus resultados para um melhor entendimento do próprio projeto.

Já a monitoração tem por objetivo acompanhar o progresso do projeto durante a sua execução, para que possam ser tomadas ações corretivas apropriadas caso o projeto desvie significativamente do planejado [SEI10]. Esse acompanhamento pode ser realizado utilizando técnicas específicas de monitoração, como o *Earned Value Analysis* (EVA), (Seção 2.1.2).

Utilizando as métricas armazenadas pelos projetos de software, também é possível realizar a previsão de métricas para projetos futuros, baseando-se em experiências de projetos anteriores, por meio da coleta, armazenamento e análise de seus dados [TEN10].

1.2 OBJETIVOS

Este trabalho tem por objetivo facilitar a gestão de métricas de um ambiente de desenvolvimento de *software*, armazenando as métricas em um repositório estruturado na forma de um *Data Warehouse*. A solução proposta apresenta um modelo analítico que oferece suporte à análise e monitoração segundo a técnica de *Earned Value Analysis* (EVA) para monitorar custos e prazos dos projetos.

Pretende-se extrair os dados de maneira automatizada das bases de dados de origem. Com isso almeja-se contribuir com parte das exigências do CMMI quanto à medição e análise, e monitoração e controle de projetos de *software*.

1.3 ORGANIZAÇÃO DO TEXTO

O texto está dividido em sete capítulos. No segundo capítulo é desenvolvida a fundamentação teórica para este trabalho, abrangendo os seguintes conteúdos: (i) métricas de software, (ii) modelos de maturidade de software, (iii) ambiente de *Data Warehousing* e (iv) *Business Intelligence*. O terceiro capítulo apresenta o cenário de uma *Organização Exemplo*, avaliada CMMI nível 3. No quarto capítulo são apresentados os trabalhos relacionados. O quinto capítulo disserta sobre a proposta

de solução para o problema exposto no capítulo três. O sexto capítulo apresenta os testes realizados para avaliar se com a construção do cubo de métricas é possível monitorar projetos de *software*. E finalmente o sétimo capítulo discorre sobre as considerações finais sobre este trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são abordados os conceitos teóricos fundamentais para entendimento do trabalho: (i) métricas de software, (ii) modelos de maturidade de software, (iii) ambiente de *Data Warehousing* e (iv) *Business Intelligence*.

2.1 MÉTRICAS DE SOFTWARE

A medição é crucial ao progresso da ciência como um todo, o progresso científico é dado através de observações e generalizações baseadas em dados e medições [KAN03]. A melhor forma de aprimorar qualquer processo de medição é medir os atributos específicos do processo e, desenvolver um conjunto significativo de métricas baseadas nestes atributos. Prover indicadores utilizando métricas conduzirá a uma estratégia de melhoramento do Processo de Desenvolvimento de *Software* (PDS) [PRE01].

Uma métrica é uma função mensurável cujas entradas são os dados relacionados ao software, tendo como resultado corresponde a um único valor numérico, que pode ser interpretado como o grau de qualidade do software. O objetivo de “medir” um software é poder avaliá-lo ao longo do seu ciclo de vida, a fim de saber se os requisitos de qualidade estão sendo cumpridos [IEEE98].

Segundo o *Institute of Electrical and Electronics Engineers* (IEEE) as métricas de software são classificadas como diretas ou derivadas. As métricas diretas não dependem de nenhum atributo, nem necessitam ser validadas. As métricas derivadas são definidas a partir de outros atributos e necessitam de validação. Ambas as métricas possuem o objetivo de proporcionar uma informação quantitativa sobre um processo ou produto, sendo importantes às atividades de gerenciamento dos projetos [IEEE98].

Segundo Kan, as métricas podem ser classificadas da seguinte forma [KAN03].

- Métricas de produto: descrevem as características do produto tais como tamanho, complexidade, desempenho e nível de qualidade.
- Métricas de processo: são utilizadas para melhorar o desempenho e a manutenção do PDS, tais como indicadores de qualidade para defeitos

encontrados e o esforço despendido para sua correção e a eficiência da remoção de defeitos.

- Métricas de projeto: representam as características do projeto e a sua execução, tais como recursos, custo e cronograma.

Para permitir a institucionalização do uso das métricas de software, elas devem ser documentadas e organizadas por intermédio de um Programa de Métricas [SEI10]. O programa de métricas deve conter a definição das métricas adotadas pela organização e as suas respectivas unidades de medida. Tanto a norma 1061 [IEEE98], como os modelos de maturidade [CMMI] e MpsBr [SOF09] não definem um conjunto mínimo de métricas, apenas afirmam que métricas úteis devem ser estabelecidas conforme a necessidade dos projetos e da organização. Neste sentido, o maior desafio de um bom plano de métricas é saber qual o conjunto de métricas úteis e significativas devem ser considerados [SIL07].

2.1.1 Programa de métricas

Um programa de métricas é uma forma de documentar e organizar métricas, permitindo que o seu uso seja institucionalizado em uma organização. O programa de métricas deve conter a definição das métricas adotadas pela organização, as suas respectivas unidades de medida, bem como as equações utilizadas para o seu cálculo, no caso de métricas derivadas [SEI10]. Tanto a norma 1061 [IEEE98] quanto modelos de maturidade como o CMMI sugerem a definição e utilização de um programa de métricas que seja significativo para os projetos e para a organização, porém, não definem quais métricas devem ser utilizadas.

As métricas podem ser logicamente agrupadas em: métricas de produto, projeto e de processo segundo as necessidades de uma determinada organização [KAN03]. Desta forma é possível analisar os diferentes níveis de gerenciamento organizacional.

2.1.2 Earned Value Analysis (EVA)

O conceito de valor agregado tem origem em 1967, no Departamento de Defesa Americano (DOD – *Department of Defense*), com o objetivo de padronizar a

maneira como as empresas contratadas reportariam o andamento de custo e prazo de grandes projetos [FLE00]. A análise do valor agregado surgiu da necessidade de realizar previsões confiáveis relacionadas aos custos e prazos no decorrer do projeto, a fim de garantir que o projeto terminará dentro do prazo estipulado e com o custo definido.

Visualizando apenas os gastos previstos e as despesas reais, o verdadeiro desempenho de custo do projeto não pode ser determinado, pois não é possível determinar quanto do trabalho físico já foi realizado. É possível apenas determinar se até o momento o projeto está dentro do orçamento definido.

A técnica *EVA* é utilizada na medição do desempenho. A *EVA* é composta pelas medidas de escopo, custos e prazo que auxiliam o gerente de projeto a avaliar e medir o desempenho e progresso projeto. A técnica *EVA* desenvolve e monitora três dimensões chave para cada atividade do cronograma [PMI08]. Definidas como os seguintes parâmetros:

- Valor Estimado (VE): indica o orçamento autorizado designado para cada atividade do cronograma, segundo as estimativas realizadas no início do projeto (linha base), considerando as atividades e os recursos envolvidos. Segundo Silveira em [SIL07], o VE pode ser definido como: o valor total do orçamento planejado, dividido pelo tempo total e acumulado até a data atual do projeto ou data de *status*.
- Valor Agregado (VA): indica a parcela do orçamento que deveria ser gasta, considerando o trabalho até o momento e o custo da linha base das atividades e recursos. Este valor é obtido pelo produto entre o percentual de trabalho completado até a data de *status* e o valor total do orçamento estimado para determinada atividade.
- Custo Real (CR): é o custo total decorrido o trabalho já realizado por um recurso ou atividade até a data de *status* ou a data atual do projeto. O CR deve corresponder, em definição, ao que foi orçado para o VE e medido no VA, por exemplo, somente horas diretas, somente custos diretos, ou todos os custos inclusive os indiretos. O CR não deve ter limite superior, tudo que for gasto para atingir o VA será medido.

Uma vez determinados estes três parâmetros, a análise é obtida com base na correlação entre os valores encontrados para cada um deles em uma determinada data de *status*. A Figura 2.1 apresenta um possível conjunto de valores VE, VA e

CR. O eixo x do gráfico representa os valores de tempo e o eixo y o custo (valor monetário). As curvas representam o valor dos parâmetros ao longo do tempo, até uma determinada data de *status*. No gráfico a referência para análise é a data de *status*. No gráfico apenas VE ultrapassa esta data, uma vez que representa o valor estimado para toda a atividade ou projeto.

- Variação de Custo (VC): representa a diferença entre o valor agregado (VA) e o custo real (CR). Se o valor de VC for positivo, então o custo está abaixo do valor estimado e, se for negativo o custo encontra-se acima do orçamento estipulado.

$$VC = VA - CR$$

$$VC \geq 0 \therefore \text{dentro do orçamento}$$

$$VC < 0 \therefore \text{fora do orçamento}$$

- Variação de Prazo (VP): representa a diferença entre o valor agregado (VA) e o valor estimado (VE). Se o valor do VP for positivo, o projeto está adiantado e, se for negativo está atrasado.

$$VP = VA - VE$$

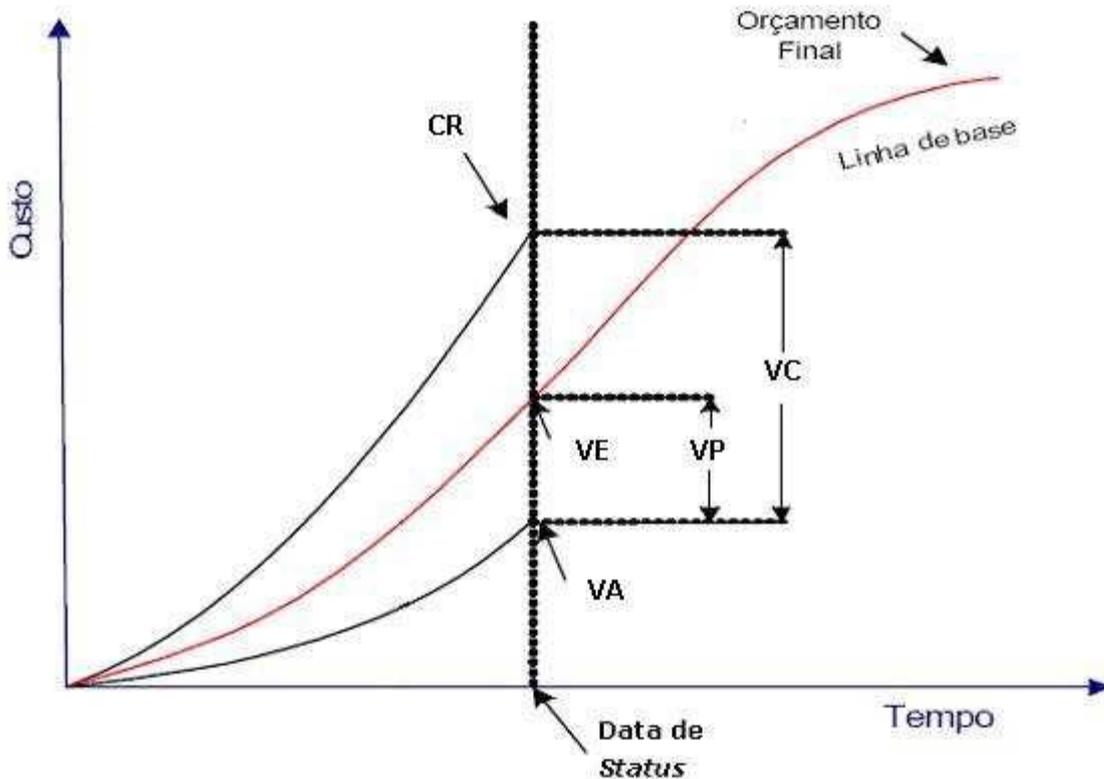
$$VP > 0 \therefore \text{adiantado}$$

$$VP = 0 \therefore \text{no prazo}$$

$$VP < 0 \therefore \text{atrasado}$$

A Figura 2.1 apresenta como as variações podem ser interpretadas no gráfico da EVA. Através da análise da Figura 2.1, é possível concluir que VC pode ser considerada como a diferença entre as curvas VA e CR e também, que VP é a diferença entre as curvas VA e VE. Através da análise é possível observar também que quanto mais distante a curva CR estiver das outras, maior será a variação do prazo e do custo para determinada data de *status*. Os valores VC e VP podem ser convertidos em indicadores de eficiência para refletir o desempenho de custos e prazos de qualquer projeto.

Figura 2.1 - Identificação das Variações no Gráfico de EVA



Fonte: Silveira (2007).

- Índice de Desempenho de Custo (IDC): é a medida que relaciona o custo do valor agregado com o custo real do projeto. O IDC é igual a razão entre o VA e o CR e demonstra a conversão entre os custos reais consumidos pelo projeto e os valores agregados em um determinado período. Se o valor de IDC for menor que 1, então o orçamento consumido foi maior que o planejado. E se o valor for maior que 1, então o custo foi abaixo do previsto. Por exemplo, um $IDC = 0,86$ indica que, para cada R\$1,00 de capital realmente consumido, apenas R\$ 0,86 estão sendo convertidos fisicamente em produto e que existe uma perda de R\$ 0,14.

$$IDC = \frac{VA}{CR} \quad \begin{array}{l} IDC \geq 1 \text{ :. menor que o planejado} \\ IDC < 1 \text{ :. maior que o planejado} \end{array}$$

- Índice de Desempenho de Prazo (IDP): é uma medida do progresso alcançado comparado ao progresso planejado em um projeto. O IDP representa a taxa de conversão do valor estimado (VE) em valor agregado (VA). O IDP pode ser utilizado em conjunto com o IDC para prever as

estimativas de término do projeto. Se o valor de IDP for menor que 1, então foi executado menos trabalho que o planejado. Se o valor for maior que 1, então foi executado mais trabalho que o planejado. Por exemplo, um IDP = 0,85 indica que 85% do tempo estimado no orçamento (VE) foi convertido em trabalho e que houve uma perda de 15% no tempo disponível.

$$IDP = \frac{VA}{VE} \quad \begin{array}{l} IDP \geq 1 \text{ .: menor que o planejado} \\ IDP < 1 \text{ .: maior que o planejado} \end{array}$$

A técnica EVA permite tomar ações corretivas e preventivas com antecedência, pois oferece aos gestores do projeto uma forma de acompanhar o andamento das atividades e verificar, com base nos indicadores apresentados, se estas estão sendo realizadas dentro do custo estimado.

2.2 MODELOS DE MATURIDADE DE SOFTWARE

Atualmente o CMMI é o modelo de maturidade mais difundido. O CMMI foi desenvolvido pelo *Software Engineering Institute* (SEI) da *Universidade Carnegie Mellon University* (CMU). O CMMI foi elaborado com o objetivo de prover um modelo capaz de avaliar o risco na contratação de empresas de *software* pela Força Aérea Norte-Americana. Esta desejava ser capaz de avaliar os processos de desenvolvimento utilizados pelas empresas que concorriam às licitações, tais como indicação da previsibilidade da qualidade, custos e prazos nos projetos contratados.

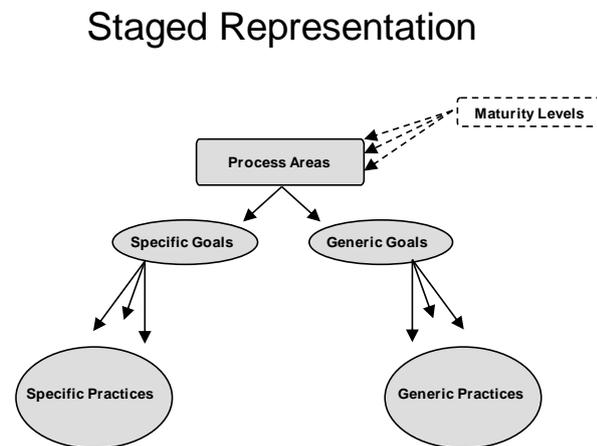
Este modelo de maturidade sugere, entre outras práticas, a coleta de métricas como uma das formas de controlar e analisar os processos e obter a melhoria contínua.

2.2.1 CMMI

O CMMI possui dois modos de representação: em estágios (*staged*) e contínuo (*continuous*), e para ambos, o elemento-chave é a *Process Area* (PA). Uma representação no CMMI é análoga à visão dos dados em um conjunto de informações providos por um banco de dados. Os dados visualizados são os mesmos para cada representação, mas o uso organização e representação dos dados são diferentes [CKS04].

Segundo o SEI a representação em estágios possui o foco na maturidade organizacional [SEI10]. Na representação em estágios, a melhoria de processo é medida por níveis de maturidade. Nível de maturidade é o grau de melhoria dos processos realizados através de um conjunto predefinido de áreas de processo. O processo de maturidade organizacional descreve o grau em que a organização implementa os comportamentos descritos em um conjunto de áreas de processo [CKS04]. Nesta representação é necessário que os níveis sejam atingidos em sequência para que a organização consiga alcançar determinado nível. A Figura 2.2 apresenta os componentes do CMMI na representação por estágios.

Figura 2.2 - Componentes do CMMI na Representação em Estágios



Fonte: SEI - Software Engineering Institute (2010).

Na representação em estágios, o CMMI considera cinco níveis de maturidade conforme apresenta a Tabela 1.

Tabela 2.1 - Níveis de Maturidade CMMI - Representação em Estágios

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

Fonte: SEI - Software Engineering Institute (2010).

- Nível 1 – Inicial: A organização não possui um processo de desenvolvimento definido e o projeto não é monitorado. Apesar da falta de planejamento, são produzidos produtos e serviços que funcionam, mas muitas vezes excedem o orçamento e o cronograma definidos.
- Nível 2 – Gerenciado: Os processos são planejados e executados em concordância com uma política organizacional. Os projetos são monitorados, controlados e revisados, e são avaliados quanto à aderência às descrições de seu processo. A disciplina do processo refletido pelo nível 2 de maturidade ajuda a garantir que os projetos são executados e geridos de acordo com seus planos documentados. Neste nível a organização começa a definir e a coletar algumas métricas alinhadas com os objetivos da organização e das suas necessidades de informação. A definição das métricas terá impacto sobre as análises que serão disponibilizadas e a conquista de níveis de maturidade posteriores.
- Nível 3 – Definido: Processo definido e fortemente institucionalizado. Os processos são geridos de forma mais ativa com a compreensão das inter-relações das atividades do processo. Neste sentido, toda a organização passa a ser regida por um conjunto de processos organizacionais (*Organization Defined Software Process – OSSP*). O OSSP contém processos fundamentais e obrigatórios e deve ser seguido por todos os projetos da organização.

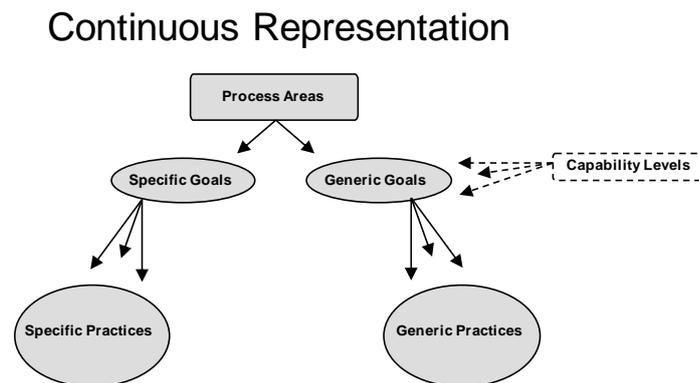
Para atingir o nível 3 de maturidade, a organização deve manter um repositório que armazene os processos organizacionais juntamente com as suas métricas, e que possibilite algumas análises sobre as métricas. O repositório deve conter as métricas dos processos previamente definidos pela organização e trabalhar com as informações de forma que possam ser utilizadas para avaliação quantitativa dos processos. Através destas métricas é possível estimar e planejar as atividades dos projetos, assim como obter uma visão unificada e quantitativa sobre a qualidade processo organizacional.

- Nível 4 – Quantitativamente Gerenciado: Prevê a utilização de técnicas estatísticas sobre as métricas para estabelecer o entendimento quantitativo dos processos. Esse entendimento é fornecido através do estabelecimento de modelos de desempenho de processo usados para representar comportamentos passados e atuais.

- Nível 5 – Otimizado: Prevê a melhoria contínua do processo através de melhorias incrementais e inovadoras de processos e tecnologia. A qualidade da organização e os objetivos de desempenho do processo são estabelecidos, e continuamente revisados para refletir os objetivos de negócios em mudança e desempenho organizacional, e utilizados como critérios na gestão de melhoria de processos.

A representação contínua possui o foco em “competência por área de processo” (*Process Area Capability*). Na representação contínua, a melhoria de processo é medida por níveis de competência. O nível de competência é o grau de melhoria dos processos realizados dentro de uma área de processo individual. Competência por área de processo representa o grau em que a organização implementa as práticas descritas em uma única área de processo [SEI10].

Figura 2.3 - Componente do CMMI na Representação Contínua



Fonte: SEI - Software Engineering Institute (2010).

Na representação contínua existem quatro níveis de competência, numerados de 0 a 3, conforme é apresentado na Tabela 2. Cada nível corresponde a um objetivo genérico e a um conjunto de práticas genéricas e específicas.

Tabela 2.2 - Níveis de Competência CMMI - Representação Contínua

Nível	Capacidade
0	Incompleto
1	Executado
2	Gerenciado
3	Definido

Fonte: SEI - Software Engineering Institute (2010).

O nível de competência 0 indica que o processo é incompleto, ou seja, ou o processo não é realizado, ou é realizado parcialmente. Alcançar o nível 1 de competência equivale a dizer que os processos associados a determinada PA são executados. No nível 2 de competência o processo pode ser planejado e controlado. Com o nível 3 de competência os processos da organização são melhor definidos e aplicados de forma consistente pois são baseados em processos organizacionais padronizados.

2.2.2 Process Areas (PAs)

Uma *Process Area* (PA) é um conjunto de práticas relacionadas a uma área que, quando implementadas coletivamente, satisfazem um conjunto de objetivos considerados importantes para melhorias nessa área. Os processos o que devem ser realizados ao invés de como fazer [SEI10].

Cada PA possui dois conjuntos de objetivos, um conjunto de objetivos específicos (*Specific Goal* - SG) e um de práticas específicas (*Specific Practice* – SP). Os objetivos específicos descrevem as características que devem estar presentes para satisfazer a PA, e são utilizados nas avaliações para ajudar a determinar se uma PA está satisfeita.

A Tabela 3 apresenta as PAs e os respectivos níveis de maturidade que estão relacionadas:

Tabela 2.3 - PAs e Respective Níveis de Maturidade

Process Area	Maturity Level
Causal Analysis and Resolution (CAR)	5
Configuration Management (CM)	2
Decision Analysis and Resolution (DAR)	3
Integrated Project Management (IPM)	3
Measurement and Analysis (MA)	2
Organizational Process Definition (OPD)	3
Organizational Process Focus (OPF)	3
Organizational Performance Management (OPM)	5
Organizational Process Performance (OPP)	4
Organizational Training (OT)	3

Product Integration (PI)	3
Project Monitoring and Control (PMC)	2
Project Planning (PP)	2
Process and Product Quality Assurance (PPQA)	2
Quantitative Project Management (QPM)	4
Requirements Development (RD)	3
Requirements Management (REQM)	2
Risk Management (RSKM)	3
Supplier Agreement Management (SAM)	2
Technical Solution (TS)	3
Validation (VAL)	3
Verification (VER)	3

Fonte: SEI - Software Engineering Institute (2010).

As PAs que definem boas práticas para análise e monitoração de projetos são: (i) Medição e Análise (MA), (ii) Monitoração e Controle de Projetos (PMC), e (iii) Gerenciamento Quantitativo de Projetos (QPM), detalhadas a seguir.

2.2.2.1 Medição e Análise (MA)

A medição e análise tem por objetivo desenvolver e sustentar a capacidade de medição utilizada para apoiar o gerenciamento. A PA de medição e análise envolve as seguintes atividades:

- Especificar objetivos de medição e análise alinhados com as necessidades de informação do projeto e os objetivos organizacionais.
- Especificar medidas, técnicas de análise e mecanismos para coleta e armazenamento de dados e geração de relatórios de *feedback*.
- Fornecer resultados objetivos que possam ser utilizados nas tomadas de decisão e auxiliar a tomar medidas corretivas apropriadas.

A integração das atividades de medição e análise com os processos de projetos suporta acompanhar o progresso e o desempenho real dos projetos contra os planos estabelecidos e fornecer uma base de medição incorporando processos adicionais no futuro.

O foco inicial para as atividades de medição é a nível de projeto. Entretanto pode ser útil a capacidade de medição a nível organizacional. Para suportar esta

competência, as atividades de medição devem apoiar as necessidades de informação em múltiplos níveis, como medições de processo, de produto e de projeto.

2.2.2.2. Monitoração e Controle de Projetos (PMC)

Possui o propósito de fornecer uma compreensão do progresso do projeto para que ações corretivas apropriadas possam ser tomadas quando o desempenho do projeto desvia do planejado.

Um plano de projeto bem documentado é a base para o monitoramento de atividades, comunicação de status do projeto e para tomar medidas corretivas. O progresso do projeto é determinado principalmente pela comparação do produto de trabalho real e os atributos das tarefas de esforço, custo e cronograma. A visibilidade adequada do progresso permite que ações corretivas sejam tomadas assim que o desempenho se desvia significativamente do plano. Um desvio é significativo se, quando deixado sem solução, desvia o projeto dos seus objetivos relacionados a prazo e custo.

Devem ser tomadas ações corretivas adequadas quando o status real se desvia significativamente dos valores esperados. Estas ações podem exigir um replanejamento, que pode incluir a revisão do plano original, o estabelecimento de novos acordos, ou incluir atividades de mitigação adicionais ao plano atual.

2.2.2.3 Gerenciamento Quantitativo de Projetos (QPM)

Possui o objetivo de gerenciar quantitativamente os processos definidos para a organização e para os projetos, de forma a alcançar os objetivos de desempenho e de qualidade de processos.

O projeto deve ser quantitativamente gerenciado, para isto os objetivos do projeto devem ser definidos e não podem contrariar os objetivos organizacionais. Junto com as práticas de PMC, o alcance destes objetivos deve ser revisado periodicamente, no andamento do projeto. A partir destes objetivos, devem ser escolhidos processos que serão gerenciados estatisticamente.

O desempenho dos processos selecionados para serem gerenciados estatisticamente devem ser monitorados periodicamente, como uma das atividades de monitoramento e acompanhamento do projeto (como descrito na PA de PMC).

Os resultados reais dos processos devem ser comparados com os objetivos estabelecidos utilizando-se técnicas estatísticas e elaboração de gráficos.

Aplicando estes conceitos ao gerenciamento dos projetos da organização, os resultados destas medições podem contribuir em diferentes aspectos para o desempenho da organização, fornecendo uma base para o equilíbrio e conciliar prioridades para lidar com um conjunto amplo de objetivos de negócios.

As PAs de Medição e Análise, e Monitoração e Controle de projetos, ambas do nível 2 de maturidade, fornecem um conjunto de práticas necessárias para o gerenciamento das métricas e monitoramento dos projetos. Este gerenciamento deve se tornar quantitativo para que a organização atinja o nível 4 de maturidade, onde a PA de Gerenciamento Quantitativo de projetos indica as práticas a serem seguidas a nível organizacional.

2.3 AMBIENTE DE DATA WAREHOUSING

No contexto de uma organização que busca certificação em alto nível de maturidade do CMMI, é necessário armazenar as métricas de software. Entretanto, os sistemas que capturam os dados operacionais de tempo e esforço, não foram projetados para apoiar a extração de informação gerencial com facilidade.

O processo de construção, acesso e manutenção de um *Data Warehouse* (DW) é denominado *Data Warehousing*. O DW pode ser considerado como uma fonte de consulta de dados, que visa disponibilizar os dados de maneira compreensível e otimizada, a partir de campos pré-calculados e de um modelo dimensional de dados. O DW é utilizado como uma implementação física do modelo de dados de suporte à decisão organizacional, armazenando informações cruciais à tomada de decisão estratégica [SIL07].

Segundo Inmon [INM05], um DW é uma coleção de dados orientados a assunto, integrado, não-volátil e variante no tempo. Segue o detalhamento de cada uma destas propriedades.

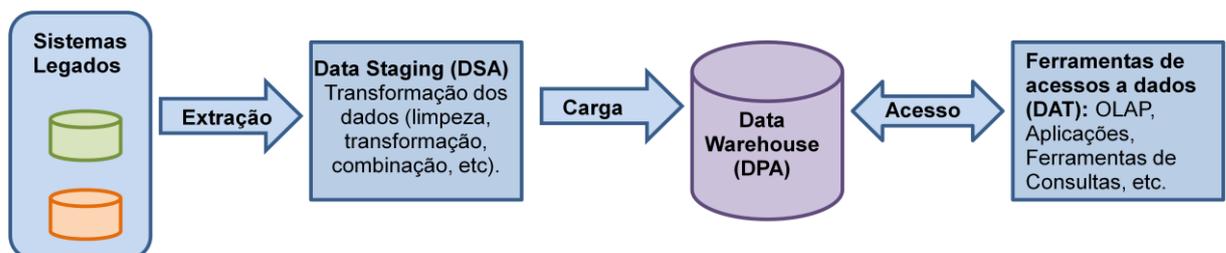
- Orientado a assunto: organiza os dados pelos principais elementos do negócio para que os dados sejam analisados. Utiliza uma visão simples, excluindo dados não úteis para a tomada de decisão.

- Integrado: fontes de dados múltiplas que devem ser limpas, convertidas, formatadas, e resumidas antes de serem armazenadas.
- Não volátil: relativo à forma como os dados são tratados no DW, os dados são carregados e acessados, mas as atualizações geralmente ocorrem no ambiente operacional.
- Variante no tempo: possibilita a manutenção de uma perspectiva histórica dos dados. Para realizar, por exemplo, o monitoramento dos projetos dia-a-dia a fim de analisar o cumprimento dos prazos.

2.3.1 Elementos de um ambiente de Data Warehousing

Um ambiente de *Data Warehouse* é composto por quatro grandes componentes genéricos: (i) sistemas legados; (ii) *Data Staging Area* (DSA); (iii) *Data Presentation Area* (DPA) e (iv) *Data Access Tools* (DAT) conforme apresentado na Figura 2.4. O primeiro componente é composto pelos sistemas legados (também chamados sistemas operacionais), que registram as informações do dia-a-dia da organização. Os objetivos dos sistemas legados diferenciam-se dos de um DW, os primeiros possuem como prioridades principais o seu desempenho no processamento das informações, a disponibilidade e normalmente não há preocupação quanto ao formato dos dados armazenados nestes sistemas. As consultas realizadas nestes sistemas são simples e fazem parte do fluxo normal de transações e restritas à demanda do sistema legado. Segundo Kimball [KIM02] se a organização possui um bom DW, o sistema legado pode ser aliviado de parte da responsabilidade de representar o passado.

Figura 2.4 - Elementos Básicos de um DW



Fonte: Kimball (2002).

O segundo elemento previsto, a *Data Staging Area (DSA)*, é uma área de armazenamento e conjunto de processos comumente denominado de Extração Transformação e Carga (ETC). Estes processos visam adequar os dados originais a través de transformações, assegurando que estes sigam os padrões determinados no DW, conforme as convenções de nomes, atributos físicos e unidades de medida [HAN01].

Após a etapa de transformação, os dados são carregados para a *Data Presentation Area (DPA)*. Nesta área os dados são organizados em um modelo dimensional projetado de acordo com as necessidades de uma organização, armazenados e disponibilizados para consultas por usuários e outras aplicações analíticas. Segundo Kimball [KIM02], a DPA é o *Data Warehouse* propriamente dito.

O último elemento ilustrado na Figura 2.4 é composto das ferramentas que provêem acesso aos dados, chamada de *Data Access Tools (DAT)*, também conhecida como camada de apresentação. Esta é definida por Kimball [KIM02] como o componente principal do DW. O termo ferramenta é empregado como uma variedade de recursos que podem ser fornecidos aos gestores de negócios para auxiliar a tomada de decisão.

2.3.2 Modelagem Dimensional

Um DW é constituído segundo uma estrutura multidimensional, a qual apresenta como componentes principais: fatos e dimensões.

As tabelas de dimensões armazenam as descrições textuais do negócio, por exemplo, a tabela dimensão de um projeto de *software* poderia armazenar dados sobre as características deste projeto como cliente, nome, categoria entre outras. Tabelas de dimensão caracterizam-se por possuir uma única chave primária.

Uma tabela Fato armazena geralmente as medidas numéricas do negócio, como por exemplo, esforço despendido por determinada tarefa. As tabelas Fato caracterizam-se por suas chaves primárias serem compostas de chaves estrangeiras das tabelas Dimensões [KIM02].

Segundo Kimball [KIM02] existem três principais esquemas para modelagem multidimensional de base de dados:

- Modelo estrela (*Star scheme*): uma tabela fato no meio ligado a um conjunto de tabelas dimensão. O modelo estrela é caracterizado por uma ou mais tabelas fato muito grandes e uma série de tabelas dimensão menores, onde cada uma contém informações sobre entradas para um atributo em particular da tabela fato. Uma tabela fato contém chaves e medidas.
- Modelo floco de neve (*Snowflake scheme*): refinamento do modelo estrela, onde a hierarquia dimensional é normalizada em um conjunto de tabelas dimensão menores, formando uma forma semelhante a um floco de neve.
- Modelo constelação de fatos: várias tabelas fato compartilham tabelas dimensão, visto como uma coleção de estrelas, por isso chamado de constelação de fatos. Este modelo é usado principalmente para tabelas fato agregadas ou quando deseja-se dividir uma tabela fato para melhor compreensão. A divisão da tabela fato é feita somente onde há a necessidade de focar na agregação de mais alguns fatos e dimensões.

2.3.3 Camada de Apresentação

O DW fornece dados integrados e históricos voltados desde a alta direção, que necessita de informações resumidas e estratégicas, até as gerências de níveis operacionais. Nos níveis operacionais, dados detalhados ajudam a observar aspectos mais táticos da organização ou de um processo específico [NOV06].

A forma como a camada de apresentação disponibiliza estas informações é muito importante, pois geralmente o usuário final não possui conhecimento suficiente para buscar a informação no formato dimensional. Por isso, as técnicas analíticas devem apoiar o uso do DW estabelecendo uma estrutura analítica consistente [KIM02].

2.3.4 OLAP

A tecnologia *On-line Analytical Processing* (OLAP) surgiu a partir da necessidade dos executivos e gerentes em dispor de informações de sua organização de forma sintetizada, através de comparações, visões personalizadas e análises históricas [KIM02].

As ferramentas de apresentação OLAP permitem derivar medidas envolvendo comparações entre períodos, percentual de diferença, médias, ou somas acumulativas. Segundo Novello [NOV06] as principais vantagens de uma ferramenta OLAP referem-se à capacidade de visualização de forma interativa das informações sob várias formas, conforme a necessidade de detalhamento. Esta capacidade de analisar as informações de várias perspectivas é denominada cubo. Algumas das principais características OLAP são apresentadas a seguir:

- Granularidade: nível de detalhe dos dados. Ou seja, quanto menor a granularidade, mais detalhada é a informação.
- *Drill Down*: permite aumentar o nível de detalhe da informação, diminuindo o grau de granularidade.
- *Drill Up*: ao contrário do *drill down*, possibilita aumentar o grau de granularidade diminuindo o detalhamento da informação.
- *Slice and Dice*: consiste em mudar a ordem das dimensões alterando assim a orientação segundo a qual os dados são visualizados. Altera linhas por colunas de maneira a facilitar a compreensão dos usuários.

A utilização de recursos analíticos, que apresentam as informações de maneira gráfica, permite que a informação organizacional torne-se compreensível para um número maior de usuários da organização. Entretanto, apenas a utilização de gráficos pode não ser suficiente. Um colaborador que não sabe quais as metas organizacionais, ao verificar um índice de 80% de previsibilidade pode não saber se este é um resultado bom ou ruim. Neste sentido, *dashboards* diferenciam-se de gráficos, pois disponibilizam as informações aos gestores baseados em indicadores pré-definidos pela organização [NOV06].

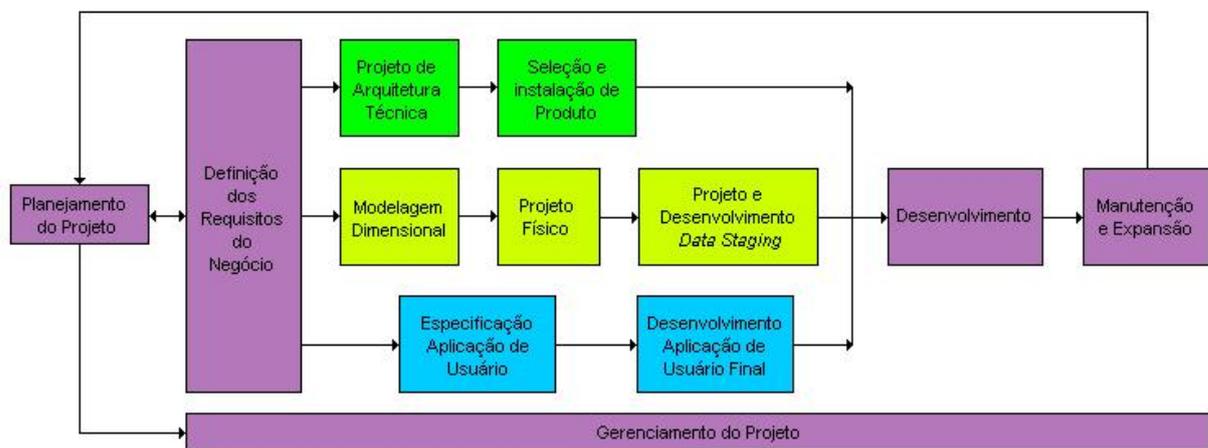
O acompanhamento efetivo do desempenho organizacional torna-se mais importante, para que desvios de comportamento sejam detectados o mais rápido possível. Para isto, a utilização de alertas acoplados a *dashboards* mostra-se um

importante recurso analítico, pois este alerta pode ser disparado sempre que houver um desvio de performance no processo organizacional [NOV06].

2.3.5 Metodologia do Desenvolvimento

O projeto de um DW depende do desenvolvimento e integração de várias tarefas e ferramentas. Considerando a grande dificuldade em coordenar estas múltiplas características, Kimball [KIM02] propõe uma metodologia para desenvolvimento de projetos de DW, cujas principais fases encontram-se na Figura 2.5.

Figura 2.5 - Metodologia de um Projeto de Data Warehouse



Fonte: Kimball (2002).

Conforme a metodologia proposta, o projeto inicia na fase de planejamento do projeto, onde é definido o escopo do projeto resultante. Nesta fase são avaliados os objetivos que a organização espera atingir com o projeto de *Data Warehousing*. O escopo é definido e inicia-se o projeto.

A *definição dos requisitos do negócio* é uma fase crucial ao desenvolvimento do projeto, pois nesta fase serão tomadas as decisões que guiarão o restante do projeto. Esta fase tem por objetivo alinhar o *Data Warehousing* com os requisitos do negócio.

O *gerenciamento do projeto* inicia-se com a definição dos requisitos e se estende durante todo o projeto. Esta atividade deve manter e gerenciar o plano de projeto, a fim de gerenciar as expectativas dos usuários.

Após a *definição dos requisitos do negócio* e iniciado o *gerenciamento do projeto* três fluxos de atividades podem ser executadas em paralelo:

- Atividades relacionadas aos dados (em amarelo na Figura 2.5): *modelagem dimensional, projeto físico e projeto e desenvolvimento de um Data Staging*.
- Atividades relacionadas à tecnologia (em verde na Figura 2.5): *projeto de arquitetura técnica e seleção e instalação do produto*. Estas atividades devem prover suporte ao projeto de *Data Warehousing*.
- Atividades relacionadas à aplicação (em azul na Figura 2.5): *especificação aplicação de usuário e desenvolvimento aplicação de usuário final*. Estas são as atividades de mais alto nível do projeto, correspondentes à camada de apresentação do DW.

A *modelagem dimensional* visa suportar os requisitos de negócio definidos junto aos usuários do DW a ser construído.

O *projeto físico* define a estrutura física necessária para suportar o banco de dados lógico. Este inclui também a definição de nomenclatura e configuração de ambiente para o banco de dados.

O *projeto e desenvolvimento de um Data Staging* é considerado uma das maiores dificuldades do projeto, pois envolve três atividades principais: extração, transformação e carga dos dados das bases operacionais para o DW, conforme foi detalhado na seção 2.3.1.

O *projeto de arquitetura técnica* estabelece suporte à integração de diversas tecnologias. Por isso, é preciso considerar os requisitos do negócio, o ambiente tecnológico atual e aspirações futuras. A *seleção e instalação de produtos* visa definir os componentes que integrarão a arquitetura técnica, selecioná-los e testá-los. Como exemplo podem ser citados: instalação do sistema gerenciador de banco de dados, bem como a definição da plataforma de hardware.

A atividade de *especificação e desenvolvimento da aplicação final do usuário final* especifica a maneira como os usuários irão acessar as informações armazenadas no DW. Os dados podem ser disponibilizados por intermédio de relatórios, *dashboards* ou outros mecanismos pré-definidos.

As atividades de tecnologia, dados e aplicação convergem para o desenvolvimento do projeto. Neste sentido, é necessário um grande a fim de garantir a união das atividades. A fase de *manutenção e expansão* envolve a constante

atualização do projeto e da aplicação a fim de continuar atendendo as expectativas do usuário.

2.4 BUSINESS INTELLIGENCE

Segundo Turban [TUR09], *Business Intelligence* (BI) é um termo que inclui arquiteturas, ferramentas, bancos de dados, aplicações e metodologias. BI é um termo bastante abrangente e pode ter diferentes significados dependendo do contexto em que for inserido. Os principais objetivos do BI são permitir o acesso interativo aos dados e fornecer aos gerentes e analistas de negócios a capacidade de realizar a análise adequada [TUR09]. O processo do BI baseia-se na transformação de dados em informações, depois em decisões e finalmente em ações.

Tipicamente as organizações recolhem informações a fim de avaliar o ambiente de negócio, e converter estas informações em vantagem competitiva. Neste sentido, as operações de *software* podem utilizar os seus próprios dados produzidos durante o desenvolvimento de um *software* para aprimorar o seu processo de desenvolvimento e assim agregar qualidade ao seu produto.

Segundo Silveira [SIL07], o DW pode ser considerado uma fonte de consulta de dados, que visa disponibilizá-los de maneira compreensível e otimizada, a partir de campos pré-calculados e de um modelo dimensional de dados. O DW é uma implementação física do modelo de dados de suporte à decisão organizacional, armazenando informações cruciais à tomada de decisão estratégica.

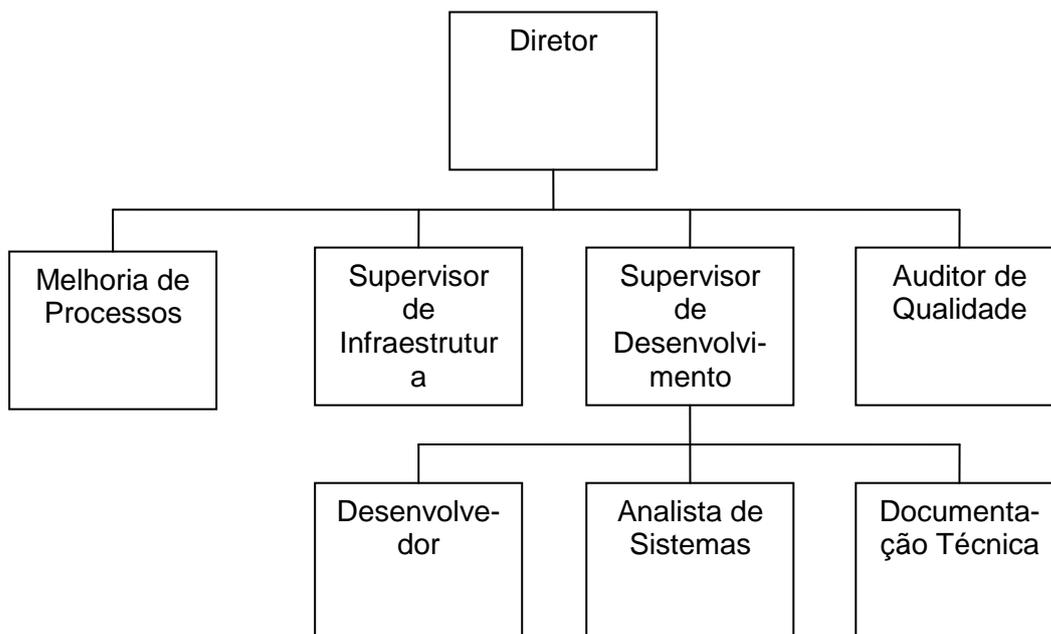
3. ESTUDO DE CASO

Neste capítulo é apresentado o cenário de uma *Organização Exemplo* que desenvolve *softwar*, avaliada CMMI nível 3. Inicialmente é descrito o processo de desenvolvimento adotado, logo após é relatada a solução computacional adotada denominada Sistema de Apoio e seus principais componentes. Posteriormente, são descritas as principais dificuldades encontradas neste cenário. Por fim é proposta uma solução a fim de automatizar a solução atual.

3.1 PROCESSO DE DESENVOLVIMENTO

De acordo com o definido na PA de OPD [SEI10], a operação de software possui um OSSP bem definido e institucionalizado. O OSSP está documentado e disponível a todos colaboradores através da intranet da organização. Os processos são documentados na forma de procedimentos, guias e fluxos que guiam o processo de desenvolvimento de *software*. A Figura 3.1 apresenta o organograma da *Organização Exemplo*.

Figura 3.1 - Organograma da Organização



Fonte: Adaptado do organograma da organização.

As responsabilidades de cada função são bem definidas e documentadas no OSSP da organização.

- Diretor – É responsável por prover recursos para o desenvolvimento dos projetos, para o ambiente de configuração e para o processo de desenvolvimento. Responsável ainda por analisar os indicadores de qualidade e definir ações preventivas e/ou corretivas quando necessário.
- Melhoria de Processos – responsável por documentar, treinar, desenvolver e melhorar os processos do OSSP da organização.
- Supervisor de Infraestrutura – responsável por manter disponível e atualizado o ambiente operacional para desenvolvimento dos projetos.
- Supervisor de Desenvolvimento – responsável por designar um Analista de Sistemas para o projeto, aprovar novos projetos autorizando o levantamento e especificação de requisitos, validar e aprovar os documentos relacionados ao projeto. O Supervisor de Desenvolvimento deve ainda garantir que o Analista de Sistemas tenha todos os recursos definidos no Plano do Projeto, supervisionar o acompanhamento do projeto feito pelo Analista de Sistemas e registrar no documento de acompanhamento do Supervisor de Desenvolvimento. Responsável ainda por medir, analisar e comunicar resultados dos indicadores qualidade.
- Auditor de Qualidade – responsável por elaborar o plano de auditorias de qualidade do projeto, a fim de garantir a qualidade do processo de desenvolvimento da organização. Responsável também por assegurar a medição, análise e publicação dos indicadores qualidade da organização.
- Desenvolvedor – responsável por executar as atividades determinadas pelo Analista de Sistemas conforme a especificação documentada no requisito do projeto, analisar os compromissos a serem assumidos e aprovar o plano de projeto. Responsável também por cumprir os prazos determinados e o tempo estimado para cada atividade conforme definido no plano do projeto.
- Analista de Sistemas – responsável por levantar e especificar requisitos técnicos e não técnicos do projeto baseado no entendimento da solução do cliente e na solução técnica adequada. Elaborar o documento de requisitos, estimar o esforço de cada atividade, negociar recursos e carga horária. O Analista de Sistemas ainda elabora o Plano de Projeto. Realiza a verificação

e validação dos componentes do projeto. É responsável por realizar o acompanhamento e supervisão do projeto, comparando os resultados obtidos em relação às estimativas. Se o desempenho desviar significativamente do planejado, deve comunicar ao Supervisor de Desenvolvimento para que seja realizada ação de contingência.

- Documentação Técnica – responsável por executar a atualização da documentação técnica do sistema, cumprindo prazos determinados e o tempo estimado para cada atividade conforme documentado no Plano do Projeto.

Antes de definir o processo de desenvolvimento da organização torna-se importante definir alguns termos que serão utilizados a seguir.

São consideradas funcionalidades quaisquer componentes do sistema, que podem ser transações, relatórios, procedurais, modelo ER, tabelas, visões, menu, documentação técnica e *scripts* de apoio.

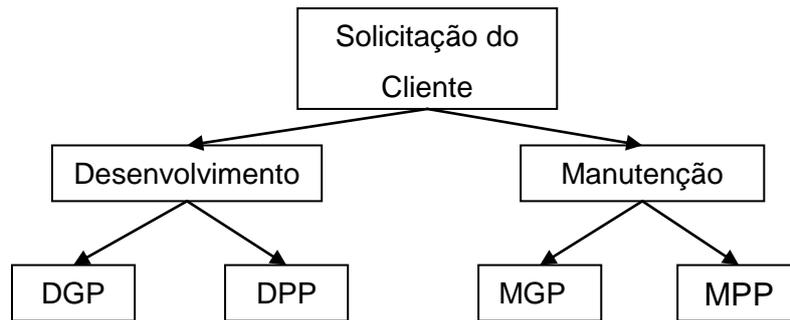
Entende-se por manutenção a correção de funcionalidade já existente e/ou adequação de funcionalidade existente aos requisitos aprovados. Implementação é o desenvolvimento de nova funcionalidade o que pode incluir a criação de novo artefato para implementação de melhorias.

Solicitação do Cliente (SC) é o registro no Sistema de Apoio do atendimento ao cliente, desde o primeiro contato do cliente até a entrega do produto, passando pelas fases de requisito, planejamento, execução e encerramento. A Solicitação do Cliente é representada por um número sequencial que identifica cada projeto.

Foram definidas quatro categorias de projetos: Manutenção de Pequeno Porte (MPP), Manutenção de Grande porte (MGP), Desenvolvimento de Pequeno Porte (DPP) e Desenvolvimento de Grande Porte (DGP). Os projetos são classificados de acordo com o número de horas do projeto. Sendo MPP e MGP projetos de manutenção e DPP e DGP projetos de implementação, como ilustra a Figura 3.2.

Todos os projetos da organização devem seguir o OSSP definido, entretanto os projetos de manutenção possuem menor documentação, quanto à análise de riscos, por exemplo, por se tratarem de projetos muito pequenos. Quanto ao acompanhamento, estes projetos também possuem um comportamento diferente visto que podem haver projetos de manutenção que duram apenas um dia, então o acompanhamento é feito apenas no encerramento da solicitação a fim de verificar se o projeto foi desenvolvido de acordo com o prazo e custo definido.

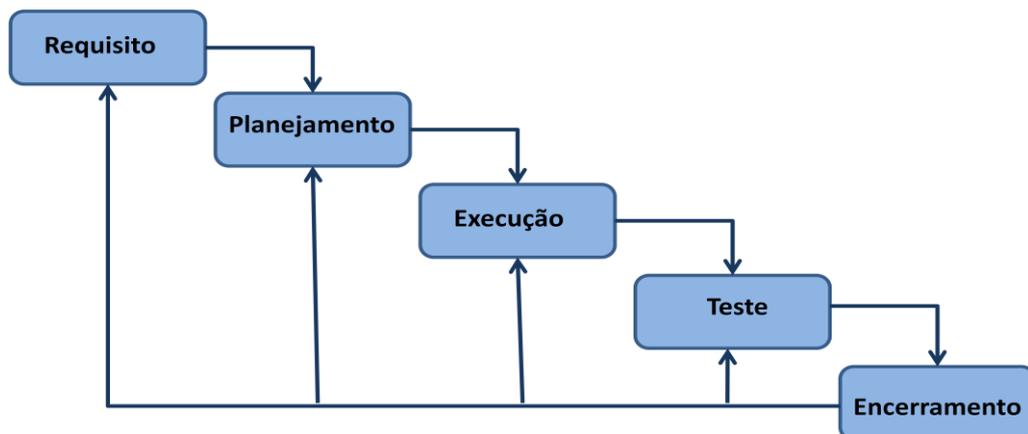
Figura 3.2 - Tipos de projetos definidos no OSSP da organização



Fonte: Mazuco (2011).

O ciclo de vida de um projeto dentro da organização é desenvolvido conforme o ciclo de vida em cascata, passa pelas fases de: requisito, planejamento, execução, acompanhamento e encerramento [SOM04]. A Figura 3.3 ilustra o ciclo de vida de um projeto da organização.

Figura 3.3 - Ciclo de Vida dos Projetos



Fonte: Adaptado de [SOM04].

Para realizar o controle das solicitações de atendimento da organização, foi desenvolvido internamente um sistema denominado Sistema de Apoio. Este sistema é responsável por conter as informações sobre os projetos tais como: o esforço de cada atividade do projeto, prazos e o registro diário das pessoas envolvidas em cada projeto. O Sistema de Apoio também elabora o cronograma do projeto.

Para os projetos MPP não há documento de requisitos por serem projetos muito pequenos, as alterações e o cronograma são documentados apenas no Sistema de Apoio.

Os projetos MGP e DPP possuem documento de requisito, entretanto o cronograma também é gerado pelo sistema de Apoio. A Figura 3.4 ilustra a tela do sistema que contém o cronograma do projeto. O cronograma do projeto é dividido

em componentes e tarefas, componentes são os programas que devem ser alterados ou criados dentro do sistema, tarefas são atividades de controle do projeto. A Figura 3.5 apresenta o cronograma das tarefas do projeto.

Nos projetos DGP as informações do cronograma também estão contidas no Sistema de Apoio, mas é necessário elaborar o documento cronograma utilizando o Microsoft Excel. A fonte das informações para preencher o cronograma, é o Sistema de Apoio, de onde o Analista de Sistemas deve copiar as informações do Sistema de Apoio para a planilha Excel, o que pode gerar erros.

Figura 3.4 - Cronograma dos Projetos - Tarefas

TareCod	Descricao da Tarefa	Dt Ini Prev	Dt Fim Prev
6	Auditoria de Qualidade	26/08/2010	21/01/2011
5	Atividades de Gerência	26/08/2010	28/01/2011
7	Planejamento	03/12/2010	11/01/2011
2	Documentação do Sistema	28/12/2010	04/01/2011
1	Teste	07/01/2011	14/01/2011
3	Encerramento	18/01/2011	18/01/2011
4	Nivelamento	07/02/2011	08/02/2011

Fonte: Adaptado do Sistema de Apoio da organização.

Figura 3.5 - Cronograma dos Projetos - Componentes

Nome	Seq	Ord	InterdComponente	Dt Ini Prev	Dt Fim Prev	Conflito
Recurso 1	1	1	SCRIPT01	08/12/2010	08/12/2010	NAO
Recurso 1	2	2	SCRIPT02	08/12/2010	09/12/2010	NAO
Recurso 1	3	3	SCRIPT03	09/12/2010	09/12/2010	NAO
Recurso 1	37	40	HLP	09/12/2010	09/12/2010	NAO
Recurso 2	4	4	TRS0010	09/12/2010	10/12/2010	NAO
Recurso 3	5	5	TRS0020	09/12/2010	10/12/2010	NAO
Recurso 2	7	7	TRS0030	10/12/2010	13/12/2010	NAO
Recurso 1	35	38	SCRIPT04	10/12/2010	13/12/2010	NAO
Recurso 3	9	9	TRS0040	10/12/2010	14/12/2010	NAO
Recurso 2	8	8	TRS0050	14/12/2010	14/12/2010	NAO
Recurso 1	12	12	TRS0060	14/12/2010	14/12/2010	NAO

Fonte: Adaptado do Sistema de Apoio da organização.

O Sistema de Apoio de também é responsável por armazenar as tarefas diárias de cada colaborador, onde cada colaborador registra em que atividade está trabalhando, a hora que iniciou e terminou a atividade. É importante ressaltar que estes registros são feitos em tempo real, ou seja, quando o colaborador inicia determinada atividade registra que neste momento iniciou a atividade. Esta é uma forma de garantir a qualidade nos dados relacionados a quanto tempo um colaborador demora para realizar uma atividade, uma vez que este não precisa lembrar posteriormente quanto tempo trabalhou em determinado projeto.

3.2 ANÁLISE E MONITORAÇÃO DOS PROJETOS

O acompanhamento dos projetos é realizado semanalmente. O Analista de Sistemas deve realizar o acompanhamento para cada projeto que estiver sob sua responsabilidade, informando a situação sobre cada projeto ao Supervisor de Desenvolvimento e este indicará se deve ser realizada alguma ação, referente aos prazos e custos de cada projeto.

Diariamente, o Supervisor de Desenvolvimento também realiza o acompanhamento verificando a situação de cada solicitação dos clientes.

De acordo com a categoria do projeto o acompanhamento pode ser realizado de maneira diferente. Para projetos de manutenção e os projetos DPP, o acompanhamento é realizado da seguinte forma: o Analista de Sistemas realiza o levantamento da situação do projeto em relação ao plano do projeto, ou seja, compara as atividades realizadas com as atividades previstas. Para verificar o andamento do projeto em relação ao planejado o Analista executa um relatório que lista todas as atividades previstas e realizadas de um projeto. Este relatório lista também qual o recurso que realizou cada atividade e quanto tempo trabalhou em determinada atividade.

Para os projetos DGP o acompanhamento é realizado de forma diferente do anterior. O Analista deve coletar as informações do Sistema de Apoio e atualizar as atividades realizadas na planilha Excel que contém o cronograma do projeto. Após atualizar o cronograma, o Analista deverá fazer o levantamento da situação do projeto e registrar as informações em um relatório de acompanhamento, que será encaminhado ao Supervisor de Desenvolvimento para análise da situação do

projeto. A elaboração do relatório ocorre da seguinte forma: após atualizar a planilha Excel do cronograma deve ser executado um macro da planilha que calcula os percentuais do andamento do projeto e atualiza a planilha de acompanhamento, conforme ilustrado na Figura 3.6.

Figura 3.6 - Acompanhamento do Projeto

Data de Status: 12/01/2010	Horas Previstas	Horas Realizadas	Percentual	
Atividade				
Requisito	78	78	100,00%	C
Atividades de Gerência	24,5	16,94	69,14%	
Auditorias Qualidade	7	3,64	52,00%	
Desenvolvimento - Tabelas	10	12,5	125,00%	
Desenvolvimento - Procedurais	118,5	41	34,60%	
Desenvolvimento - Transações	86,5	68,5	79,19%	
Desenvolvimento - Relatórios	4,5	3,5	77,78%	
Desenvolvimento - Menu	2	1,5	75,00%	
Documentação - Técnica	11,5	14	121,74%	
Testes	20	22,92	114,60%	
Encerramento	9	0	0,00%	
Total	375	264	70,40%	
				Situação do Projeto:

Fonte: Adaptado do *template* da organização.

O acompanhamento do Supervisor de Desenvolvimento é realizado da seguinte forma: uma planilha contém todas as solicitações que estão em aberto, onde contém o número da Solicitação do Cliente, categoria de projeto, o nome da pessoa, função no projeto e cada um dos componentes que serão criados ou alterados no projeto. São revisadas as solicitações na fase de desenvolvimento ou encerradas no dia anterior. Emite o relatório que contém a lista das atividades de cada projeto no período a ser verificado. Para cada componente registra a situação na Planilha de Acompanhamento, ilustrada pela Figura 3.7.

Figura 3.7 - Planilha de Acompanhamento do Supervisor de Desenvolvimento

Planilha de Acompanhamento de Projetos							
Projeto	Categoria	Recurso	Papel	Componente/Tarefa	Prazo	Custo	Status
10001	DGP	Recurso 3	Desenvolvedor	PROC0100	No Prazo	Abaixo do Custo	Encerrada
10001	DGP	Recurso 1	Analista	PROC0200	No Prazo	Abaixo do Custo	Encerrada
10001	DGP	Recurso 1	Analista	HLP	No Prazo	No Custo	Encerrada
10001	DGP	Recurso 2	Documentação	DOC0020	Adiantado	Abaixo do Custo	Encerrada
10001	DGP	Recurso 1	Analista	Planejamento	No Prazo	Abaixo do Custo	Encerrada
10001	DGP	Auditor	Qualidade	Qualidade		No Custo	Encerrada
10001	DGP	Recurso 1	Analista	Teste	No Prazo	Abaixo do Custo	Encerrada
10001	DGP	Recurso 1	Analista	Prep. Doc	No Prazo	No Custo	Encerrada
10001	DGP	Recurso 1	Analista	Encerramento	No Prazo	Acima do Custo	Encerrada
10001	DGP	Recurso 1	Analista	Gerência		Abaixo do Custo	Encerrada
10002	MPP	Recurso 2	Desenvolvedor	MEN0010	No Prazo	Acima do Custo	Execução
1002	MPP	Recurso 2	Desenvolvedor	TRS0030	Adiantado	No Custo	Execução
1002	MPP	Recurso 2	Desenvolvedor	PROC0350	No Prazo	No Custo	Execução
1002	MPP	Qualidade	Qualidade	Qualidade		Acima do Custo	Execução
1002	MPP	Recurso 2	Desenvolvedor	Planejamento	Adiantado	No Custo	Execução
1002	MPP	Recurso 2	Desenvolvedor	Teste	No Prazo	No Custo	Execução
1002	MPP	Recurso 2	Desenvolvedor	Encerramento	No Prazo	Acima do Custo	Execução
1002	MPP	Recurso 2	Desenvolvedor	Gerência		Abaixo do Custo	Execução

Fonte: Adaptado do *template* da organização.

Para realizar a análise de custo das atividades dos projetos, a técnica utilizada é linear, por exemplo, se uma atividade foi planejada para durar 1 hora e durou 1 hora e 10 minutos, a atividade está acima do custo, se levou 55 minutos, está abaixo do custo. Durante seu desenvolvimento, não há como prever se uma atividade irá ou não atrasar.

3.2.1 Plano de métricas atual

O plano de métricas atual da organização compreende métricas de previsibilidade de prazo e esforço dos projetos. Estas métricas são elaboradas na forma de indicadores, e disponibilizadas para os colaboradores mensalmente. Estes indicadores são usados como suporte a tomadas de decisão e definição do processo de melhoria contínua.

A Tabela 3.1 apresenta um resumo do plano de métricas atual da organização.

Tabela 3.1 - Programa de Métricas Atual

Métricas Derivadas	Métricas Diretas	Equação	Unidade
Prazo (P)	Numero de Dias Realizado (NR)	$P = \frac{NR}{NP} \times 100$	%
	Numero de Dias Previsto (NP)		
Esforço (E)	Horas Realizadas (HR)	$E = \frac{HR}{HP} \times 100$	%
	Horas Previstas (HP)		
Horas Excedentes (HE)	Horas Realizadas (HR)	$THE = HR - HP$	Horas
	Horas Previstas (HP)		
Reavaliação (R)	Horas Previstas (HP)	$R = \frac{HE}{HP} \times 100$	%

Fonte: Adaptado do plano de métricas da organização.

Estes indicadores são elaborados por nível de projeto e avaliados dentro de limites de especificação definidos no OSSP da organização.

3.3 CONSIDERAÇÕES SOBRE O ESTUDO DE CASO

Com base no que foi apresentado neste capítulo, percebe-se que o atual ambiente de coleta dos dados para monitoração de projetos exige muito trabalho manual de extrair os dados da base de dados para planilhas Excel a fim de obter indicadores sobre o cumprimento do prazo e custo de um projeto.

Com base nesta análise acredita-se que a organização precisa de um ambiente de monitoração dos projetos onde não seja necessária a extração manual dos dados. Uma característica do ambiente atual que facilita o desenvolvimento de um projeto de DW é os dados estarem centralizados em uma única base de dados o que facilita o processo de extração dos dados.

Portanto, conclui-se que as características e as dificuldades da organização em monitorar seus projetos serviram para enfatizar a necessidade de uma solução que contemple os aspectos computacionais mencionados no Capítulo 2.

4. TRABALHOS RELACIONADOS

Este capítulo apresenta as pesquisas relacionadas a este trabalho, com temas que buscam soluções para mensurar a qualidade de *software*, a partir de métricas e de um repositório central.

4.1 TENÓRIO [TEN10]

Este trabalho apresenta um modelo de esforço em manutenção de software denominado *Modelo-E10*, esta solução combina diferentes técnicas de estimativas a fim de alcançar boas aproximações do esforço de trabalho estimado sem depender da expertise de pessoas.

O *Modelo-E10* é um modelo com o propósito de estimar o esforço de trabalho de manutenção de software. O modelo pode trabalhar com dois tipos de fontes de dados: “Dados de Projetos” e “Dados de Outras Técnicas”. Os “Dados de Projetos” são aqueles coletados por um projeto de uma organização e armazenados em um ou mais repositórios de dados. Geralmente um banco de dados ou ferramentas diversas para acompanhamento do projeto. É sobre os “Dados de Projetos” que atuam as técnicas de estimativas para tentar encontrar um valor do esforço de trabalho de manutenção de software próximo ao esforço real.

A Figura 4.1 apresenta o *Modelo-E10*, onde o mesmo é constituído de oito fases, sendo elas:

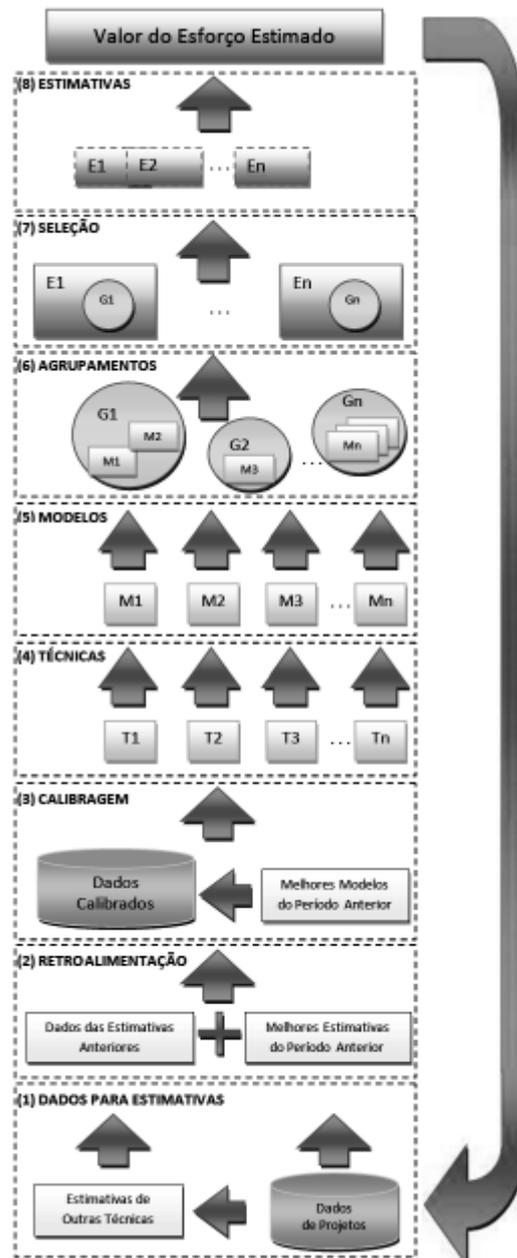
1. Dados para Estimativas – Nesta fase os Dados de Projetos devem ser devidamente extraídos e disponibilizados para o *Modelo-E10*;
2. Retroalimentação – Nesta fase, os dados das mais recentes e melhores estimativas são agregados aos dados das estimativas dos períodos anteriores;
3. Calibragem – Nesta fase os resultados da Retroalimentação são devidamente armazenados em um repositório de calibragem e uma avaliação dos melhores modelos preditivos é realizada;
4. Técnicas – Nesta fase diferentes técnicas de estimativas são utilizadas pelo *Modelo-E10*;
5. Modelos – Nesta fase são aplicados os modelos preditivos encontrados pelas técnicas utilizadas;

6. Agrupamentos – Nesta fase os valores gerados pelos modelos preditivos são agrupados conforme o Nível de Exigência, ou seja, o percentual parametrizável que indica a tolerância de erro do valor do esforço estimado em relação ao valor de esforço real das demandas definido;

7. Seleção – Nesta fase os agrupamentos com maior número de indivíduos são selecionados; e

8. Estimativas – Nesta fase os valores de um ou mais agrupamentos são devidamente combinados para comporem um valor único de esforço de trabalho de manutenção de software.

Figura 4.1 - Modelo E-10 para Estimativas de Esforço

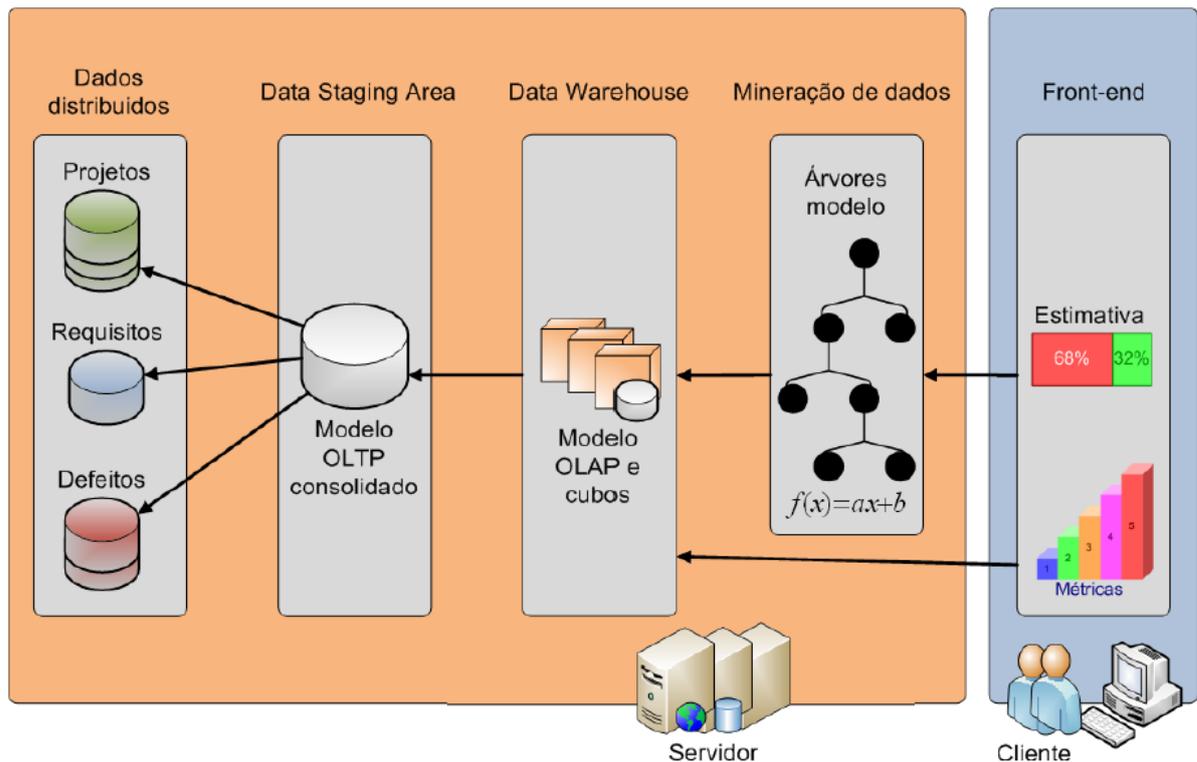


Fonte: Extraído de Tenório (2010).

4.2 REZENDE [REZ09]

Este trabalho consiste na execução de um processo de descoberta de conhecimento baseado em histórico com o objetivo de realizar a predição de métricas de software utilizando mineração de dados cuja arquitetura é apresentada na Figura 4.2.

Figura 4.2 - Arquitetura Macro do Sistema



Fonte: Extraído de Rezende (2009).

O sistema proposto por Rezende possui como requisitos funcionais: estimar as métricas de um *software*, consultar entidades por indicadores, ou seja, analisar a estimativa gerada pelo sistema, e gerar relatórios gráficos ou em modo textual (tabelado) [REZ09].

4.3 SILVEIRA [SIL07]

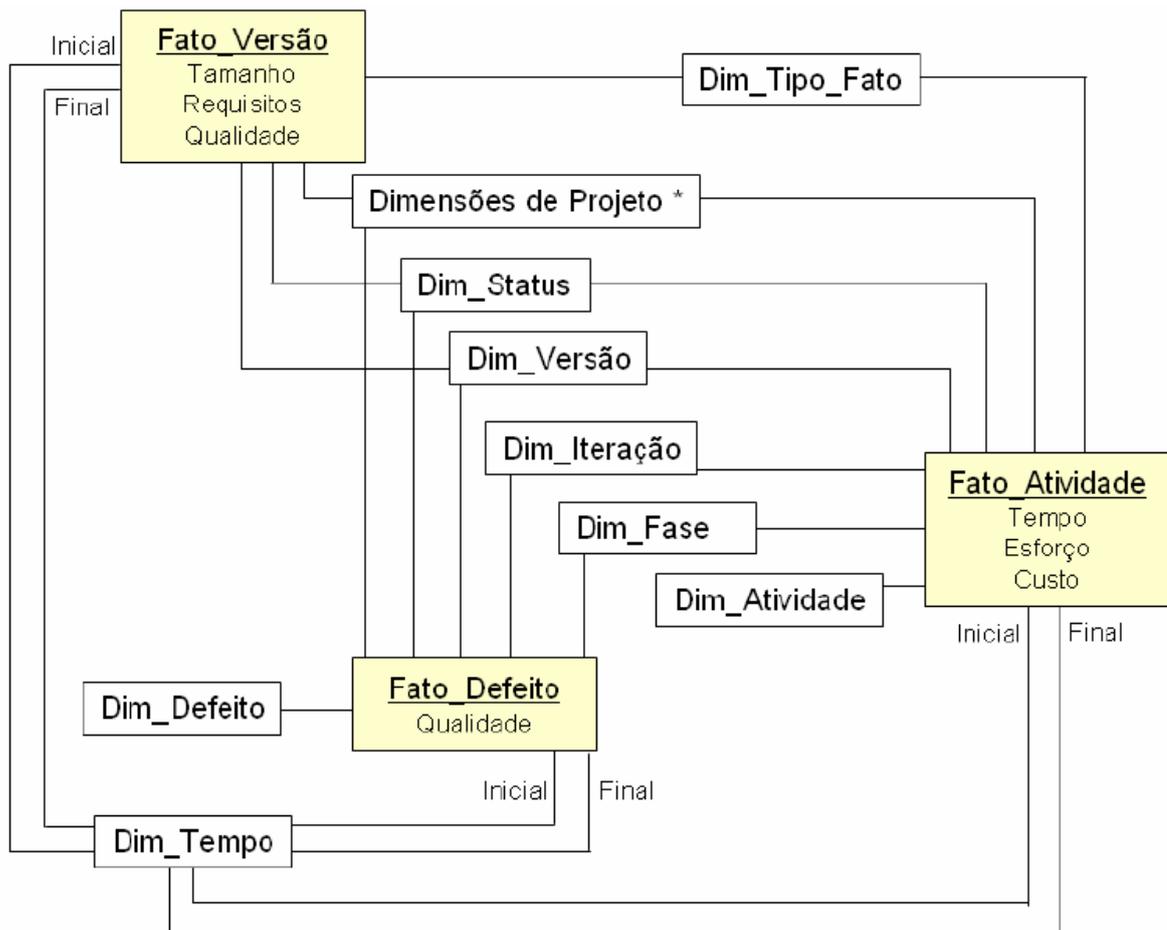
Este trabalho propõe um processo automatizado de ETC de métricas da qualidade de *software*, a serem armazenados em um Repositório de Dados. Silveira analisa a mensuração da qualidade a partir de três dimensões ortogonais: nível de

mensuração, atividades de supervisão e plataforma, para tanto é definido um instrumento de referência cujas informações detalhadas podem ser encontradas em [SIL07].

Silveira realizou um estudo de caso onde é apresentado o ambiente de desenvolvimento de *software* de uma organização de TI, considerada de grande porte na América Latina e certificada CMM3.

A solução proposta neste trabalho denomina-se *SPDW+* e consiste em um ambiente de *data warehousing*, que oferece suporte à análise e à monitoração da mensuração da qualidade de *software*. A mesma contempla um modelo analítico elegante e abrangente, apresentado pela Figura 4.3, que permite a monitoração segundo a técnica EVA, a partir no nível de atividade.

Figura 4.3 - Modelo Analítico do SPDW+

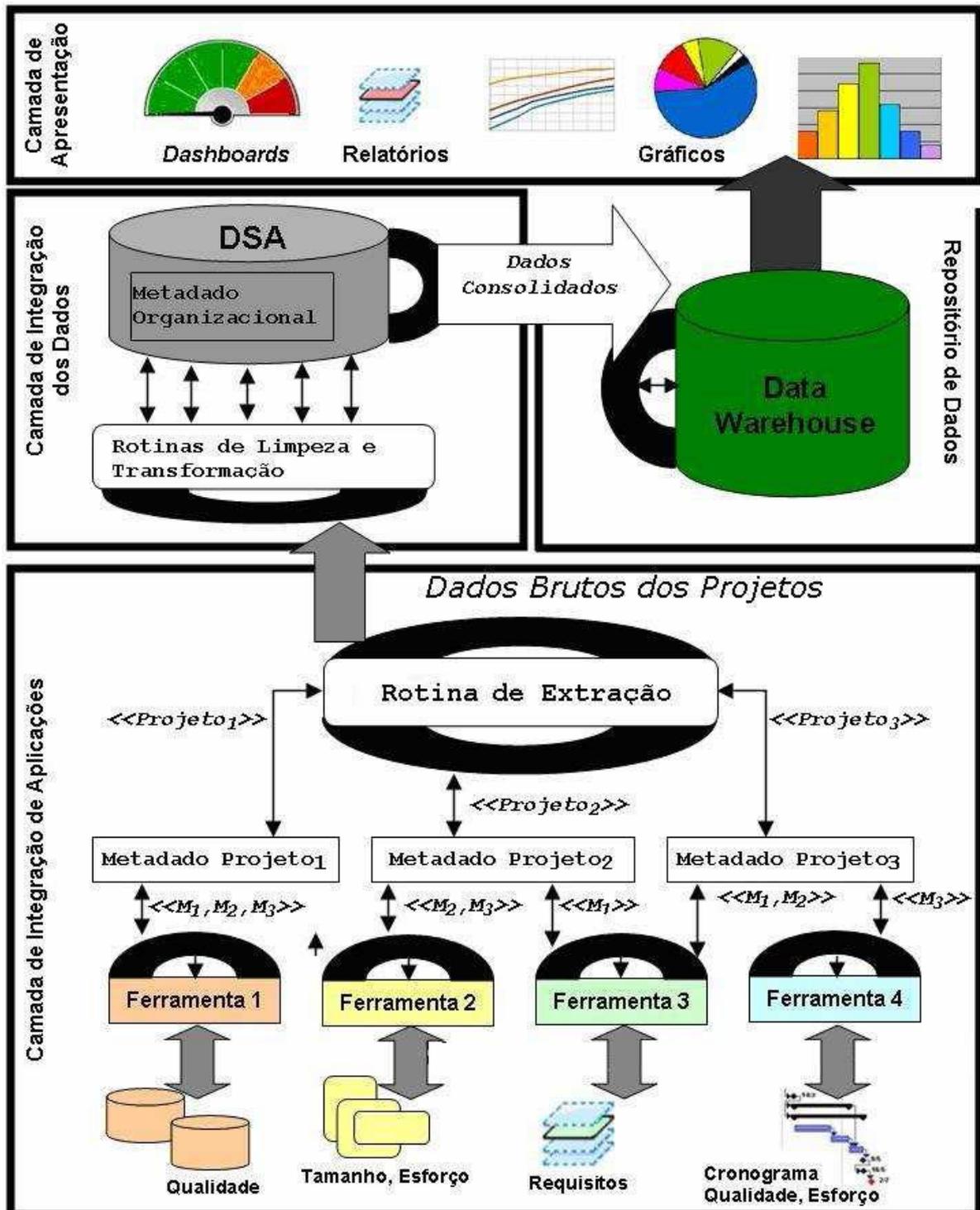


Fonte: Extraído de Silveira (2007).

O *SPDW+* também define um processo automatizado de ETC das métricas que trata a carga incremental, a alta frequência e a baixa latência e intrusão da

coleta. Para isso, a solução é consistente com a arquitetura orientada a serviços, apresentada pela Figura 4.4. Foram especificados os seus principais componentes: (i) Rotina de Extração, (ii) *wrappers*, (iii) Rotinas de Limpeza e Transformação, (iv) Metadados de Projeto, (v) DSA, (vi) serviço de carga incremental e (vii) Repositório de Dados (DW).

Figura 4.4 - Arquitetura do SPDW+



Fonte: Extraído de Silveira (2007).

4.4 NOVELLO [NOV06]

Este trabalho descreve a arquitetura de um ambiente de *Data Warehouse* que provê suporte a adoção de um PM através do armazenamento de dados resultantes de diferentes PDS em uma base de dados unificada e centralizada. O modelo multidimensional do DW, juntamente com os recursos de análise especificados e disponibilizados através da camada de apresentação, possui como adicional o fato de considerar as especificidades do ambiente de PDS. Ferramentas que implementam recursos OLAP e/ou BI voltadas a análises de processos mostram-se insuficientes diante desta realidade. O trabalho desenvolvido considera as características específicas de métricas voltadas à análise de PDS, as necessidades de informações de cada perfil de usuário encontrado em uma organização de software, como também suas diferentes expectativas de análise.

Grande parte das métricas que visam a análise do PDS é representada por razões. Esse fato dificulta a utilização da tecnologia OLAP que somente trabalha com agregações. Assim, o modelo analítico somente armazena as métricas bases e quando necessário, o ambiente calcula as métricas derivadas através de um redirecionador de consultas.

Os recursos analíticos e gráficos disponibilizados através da camada de apresentação consideram as necessidades de análise dos diferentes perfis de usuários organizacionais, provendo diferentes níveis de informação (organização, projeto, etc). A semântica aos dados disponibilizados é fornecida através da definição de taxonomias de categorias, baseada nos indicadores de métricas.

Quanto às limitações do modelo analítico, a principal delas refere-se à indisponibilidade de estimativas de esforço em nível de atividade. A existência dessa se faz necessária devido a não existência desta informação em fontes de dados do ambiente transacional. Quanto aos recursos analíticos e a camada de apresentação tem-se como limitação a indisponibilidade de informação segundo um perfil de usuário e a indisponibilidade de dados através da *web*.

4.5 CONSIDERAÇÕES SOBRE OS TRABALHOS RELACIONADOS

Diferentes alternativas vêm sendo desenvolvidas para analisar e monitorar seu PDS. Os trabalhos de Tenório [TEN10] e Rezende [REZ09] apresentam análises baseadas em estimativas de métricas de *software* utilizando *data mining*.

Novello apresenta um ambiente de DW para o armazenamento de métricas, entretanto não oferece suporte à monitoração. Silveira oferece suporte à monitoração dos projetos utilizando a técnica EVA, por isto o presente trabalho está sendo desenvolvido com base em [SIL07].

Todos os trabalhos apresentado baseiam suas soluções em ambientes de *Data Warehouse* o que reforça a escolha de utilizar a mesma metodologia para desenvolver o presente trabalho.

5. SOLUÇÃO PROPOSTA

Este capítulo descreve a solução computacional implementada para automatizar a mensuração da qualidade de *software* com suporte a monitoração de projetos. As próximas seções contemplam: (i) a definição do programa de métricas suportado pela solução; (ii) a determinação do modelo analítico e (iii) a especificação do processo de Extração, Transformação e Carga das métricas.

5.1 PROGRAMA DE MÉTRICAS

Para suportar as atividades de supervisão do cumprimento de prazos e custos dos projetos de *software* é necessário estabelecer um conjunto de métricas adequadas. A partir dessas métricas é possível realizar análises, monitorações e previsões. Os itens subitens desta seção relatam a inclusão do EVA no programa de métricas atual da organização e uma demonstração do acompanhamento de prazo e custo de um projeto utilizando esta técnica.

5.1.1 Programa de métricas com EVA

Segundo a técnica de EVA, descrita por Fleming em [FLE00], só é possível monitorar ao longo do tempo algo que apresente uma linha base, e que permita uma comparação entre o inicialmente planejado e o efetivamente realizado, verificando se o desempenho apresenta variações dentro de uma faixa de valores aceitável. Assim com base nestas exigências e na análise realizada por Silveira [SIL07] sobre as áreas de qualidade das métricas, foram identificadas as métricas mais adequadas a serem monitoradas.

Baseado nestas considerações, optou-se por incluir métricas para monitorar de maneira mais efetiva a previsibilidade de prazo e custo. Considerou-se esforço um caso particular do custo, quando os recursos alocados para realizar determinada tarefa apresentam custos equivalentes e, ao considerar custo entende-se que esforço também está sendo coberto.

Logo, para oferecer suporte à monitoração de prazo e custo, utilizando a técnica de EVA foram definidas as seguintes métricas:

- Diretas:
 - Data Inicial Real (DIR)
 - Data Final Real (DFR)
 - Data Inicial do *Baseline* Original (DIBO)
 - Data Final do *Baseline* Original (DFBO)
 - Data Inicial do *Baseline* Revisado (DIBR)
 - Data Final do *Baseline* Revisado (DFBR)
 - Custo Real (CR)
 - Custo do *Baseline* Original (CBO)
 - Custo do *Baseline* Revisado (CBR)
 - Percentual de Trabalho Completado (%TC)
 - Data de Status (DS)
- Derivadas:
 - Variação de Prazo do *Baseline* Original (VPBO)
 - Variação de Prazo do *Baseline* Revisado (VPBR)
 - Variação de Custo do *Baseline* Original (VCBO)
 - Variação de Custo do *Baseline* Revisado (VCBR)
 - Variação de Prazo Agregada (VPA)
 - Variação de Custo Agregada (VCA)
 - Índice de Desempenho de Prazo (IDP)
 - Índice de Desempenho de Custo (IDC)

As métricas referentes ao *baseline* revisado são corrigidas quando há replanejamento no projeto. No caso de um projeto possuir mais de um replanejamento, serão consideradas nas métricas de *baseline* revisado as métricas do último planejamento, pois entende-se que estas estão mais perto do efetivamente realizado.

A Tabela 5.1 apresenta a definição das métricas derivadas, juntamente com o seu objetivo as métricas diretas utilizadas para seu cálculo, a sua equação e unidade de medida. A elaboração do plano de métricas é baseada na dissertação de Silveira [SIL07].

Tabela 5.1 - Programa de Métricas Proposto

Métricas Derivadas	Objetivos	Métricas Diretas	Equação	Unidade
Variação de Prazo do <i>Baseline</i> Original (VPBO)	Dar visibilidade da aderência do projeto em relação aos compromissos do projeto, possibilitando que este seja entregue no prazo.	Data Final Real (DFR)	$VPBO = \frac{DFR - DIR}{DFBO - DIBO} \times 100$	%
		Data Final do <i>Baseline</i> Original (DFBO)		
		Data Inicial do <i>Baseline</i> Original (DIBO)		
Variação de Prazo do <i>Baseline</i> Revisado (VPBR)		Data Final Real (DFR)	$VPBR = \frac{DFR - DIR}{DFBR - DIBR} \times 100$	%
		Data Final do <i>Baseline</i> Revisado (DFBR)		
		Data Inicial do <i>Baseline</i> Revisado (DIBR)		
Variação de Custo do <i>Baseline</i> Original (VCBO)	Dar visibilidade da diferença entre o custo estimado para o projeto e o custo realizado.	Custo Real (CR)	$VCBO = \frac{CR - CBO}{CBO} \times 100$	%
		Custo do <i>Baseline</i> Original (CBO)		
Variação de Custo do <i>Baseline</i> Revisado (VCBR)		Custo Real (CR)	$VCBR = \frac{CR - CBR}{CBR} \times 100$	%
		Custo do <i>Baseline</i> Original (CBO)		
Variação de Prazo Agregada (VPA)	Fornecer a relação entre o <i>baseline</i> (original ou revisado), o <i>baseline</i> real e o percentual de trabalho completado até a data de <i>status</i> .	Data Final <i>Baseline</i> Original (DFBO)	$VPA = CBO \times \left(\frac{TC}{100} - \frac{DS - DIBO}{DFBO - DIBO} \right)$	%
		Data Inicial <i>Baseline</i> Original (DIBO)		
		Data de <i>Status</i> (DS)		
		Custo <i>Baseline</i> Original (CBO)		
		% Trabalho Completado (TC)		
Variação de Custo Agregada (VCA)	Fornecer a relação entre o custo (original ou revisado), o custo real e o percentual de trabalho completado.	Custo do <i>Baseline</i> Original (CBO)	$VCA = \frac{CBO \times TC}{100} - CR$	%
		% Trabalho Completado (TC)		
		Custo Real (CR)		

Índice de Desempenho de Prazo (IDP)	Fornecer a taxa de conversão do valor estimado em valor agregado.	% Trabalho Completado (TC)	$IDP = \frac{TC \times (DFBO - DIBO)}{100 \times (DS - DIBO)}$	%
		Data Final <i>Baseline</i> Original (DFBO)		
		Data Inicial <i>Baseline</i> Original (DIBO)		
		Data de <i>Status</i> (DS)		
Índice de Desempenho de Custo (IDC)	Fornecer a relação entre os custos consumidos pelo projeto e valor agregado.	Custo <i>Baseline</i> Original (CBO)	$IDC = \frac{CBO \times TC}{100 \times CR}$	%
		% Trabalho Completado (TC)		
		Custo Real (CR)		

Fonte: Mazuco (2011).

É importante ressaltar que a métrica %TC foi calculada com base em CR e CBO. Calculando-se qual a porcentagem de CBO equivale CR até a Data de *Status*.

5.1.2 Monitoração de Custo e Prazo segundo EVA

Nesta seção é apresentado um exemplo da monitoração de custo e prazo, segundo a técnica de EVA. Inicialmente, é apresentado o cronograma de um projeto fictício (Tabela 5.2), após são apresentadas métricas de monitoração (Tabela 5.3) e sua interpretação.

A Tabela 5.2 apresenta o planejamento inicial do projeto fictício, com duração de 3 semanas e possuindo 2 recursos. O Recurso 1 possui carga horária de 4 horas por dia e o Recurso 2, 6 horas por dia.

Tabela 5.2 - Cronograma de um projeto fictício.

Atividade	Recurso	CBO	DIBO	DFBO
Atividade 1	Recurso 1	16,00	05/out	08/out
Atividade 2	Recurso 2	8,00	05/out	06/out
Atividade 3	Recurso 1	40,00	09/out	22/out
Atividade 4	Recurso 2	30,00	07/out	13/out
Atividade 5	Recurso 2	40,00	14/out	22/out

Fonte: Mazuco (2011).

A Tabela 5.3 apresenta o cronograma monitorado com EVA. As colunas da tabela apresentam os valores estimados (CBO e VE DS – Valor estimado na data de

status), o valor agregado (VA), as métricas diretas (%TC e CR) e derivadas (VPA, VCA, IDP e IDC). Em relação ao prazo, supondo-se uma distribuição linear do orçamento, é esperado que ao término da primeira semana, as atividades estejam 33% concluídas; na segunda 66%; e, na terceira 100%. A tabela apresenta valores correspondentes à segunda semana de execução do projeto, com data de *status* igual a 15/out.

Tabela 5.3 - Exemplo de Cronograma Monitorado com EVA

	VE	VEDS	CR	%TC	VA	VPA	VCA	IDP	IDC
Atividade 1	16,00	16,00	16,00	100	16	0,00	0	1	1
Atividade 2	8,00	8,00	17,00	100	8	0,00	-9	1	0,47
Atividade 3	40,00	28	17,00	42	16,8	-1,66	-0,2	0,91	0,99
Atividade 4	30,000	30,00	21,00	70	21	169,8	0	0,53	1
Atividade 5	40,00	12,00	12,0	30	12	7,00	0	2,4	1

Fonte: Mazuco (2011).

As células em laranja da Tabela 5.3 correspondem às tarefas rigorosamente no prazo e no custo inicialmente planejado. As em verde identificam as atividades que encontram-se adiantadas ou abaixo do orçamento previsto. As células em vermelho representam atividades em atraso ou acima do custo planejado.

5.2 MODELO ANALITICO

O modelo analítico descrito nesta seção é conseqüente do Programa de Métricas apresentado na Seção 5.1.1. Também são apresentadas as questões de negócio que devem ser abordadas para um melhor acompanhamento e monitoramento dos projetos de software.

5.2.1 Questões de Negócio abordadas

Inicialmente para o desenvolvimento do modelo analítico foram considerados os perfis dos usuários da organização que irão utilizar o DW desenvolvido de acordo com o organograma apresentado na Figura 3.1. Através da análise do organograma (Figura 3.1), é possível verificar que os gestores interessados em realizar a análise dos dados sobre os projetos são o Diretor e o Supervisor de Desenvolvimento. O

Diretor, por ser o responsável pela empresa, preocupa-se com o desempenho dos projetos, que impactam diretamente no faturamento da empresa. O Supervisor de Desenvolvimento por ser responsável por reportar a situação dos projetos ao Diretor necessita ter um resumo da situação dos projetos por categoria e por tipo de projeto desenvolvimento ou manutenção.

O Analista de Sistemas necessita acompanhar os projetos que estão sob sua responsabilidade, porém precisa saber se cada atividade está dentro do prazo e do custo definido. O Desenvolvedor também pode estar interessado em saber o desempenho das tarefas designadas para ele em determinado projeto.

O desenvolvimento do modelo analítico foi dirigido para a realidade do ambiente organizacional em questão, onde as necessidades de análise organizacional sobre o PDS são representadas por diferentes perspectivas de análise. Baseado nas questões de negócio expostas foi elaborado o modelo dimensional que será apresentado a seguir.

5.2.2 Modelagem Dimensional

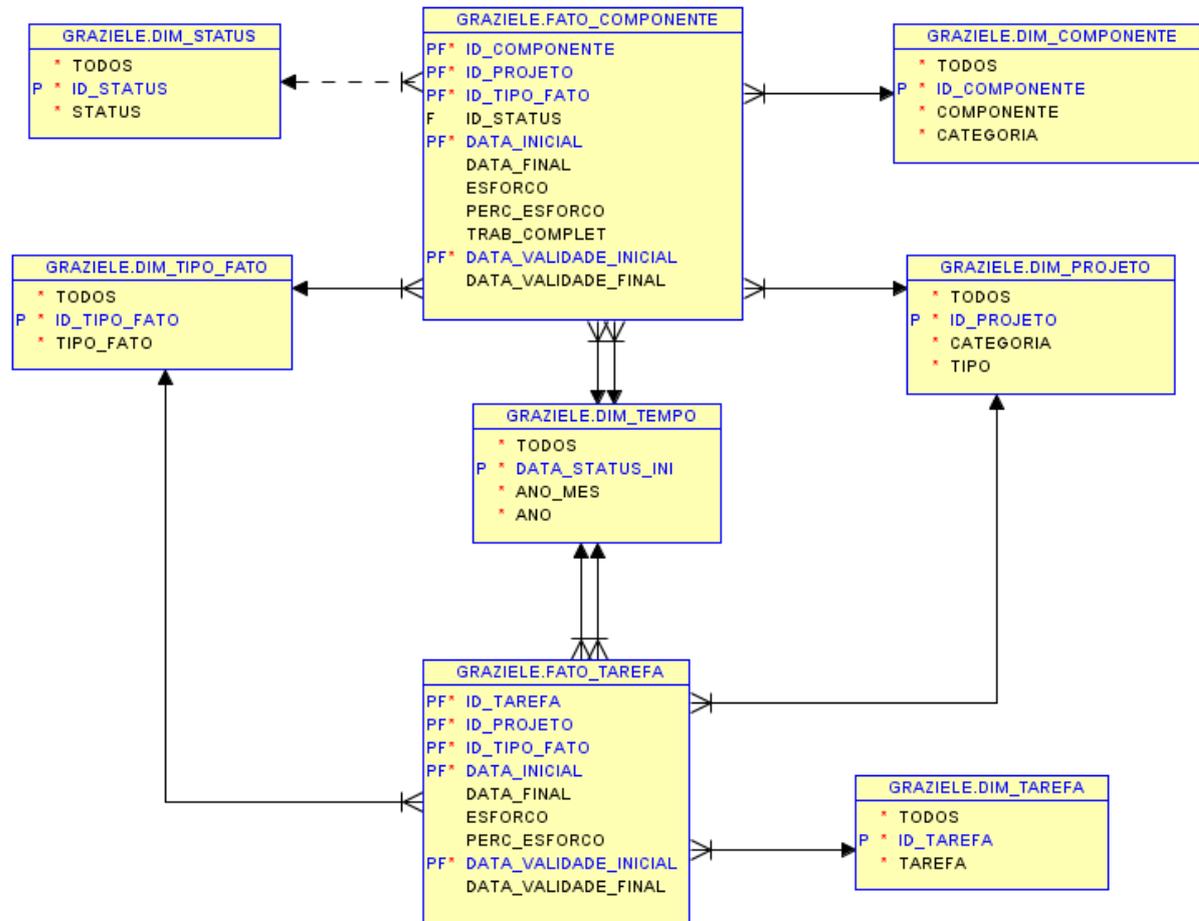
O modelo analítico desenvolvido, apresentado na Figura 5.1, possui a estrutura dimensional do tipo constelação de fatos e oferece suporte à análise e monitoração. As tabelas fato são compostas por chaves estrangeiras, responsáveis por manter a integridade referencial das dimensões e possibilitar diferentes perspectivas de análise.

Segundo Kimball [KIM02] as tabelas fato devem conter apenas métricas diretas, as métricas derivadas devem ser calculadas no momento da criação dos cubos.

Os fatos do modelo são descritos na Tabela 8, as dimensões são descritas na Tabela 9.

A tabela *Fato_Tarefa* armazena apenas métricas de análise sobre as tarefas (VPBO,VPBR,VCBO, VCBR), não incluindo as métricas para monitoração. A realização das tarefas ocorre de maneira diferente à realização dos componentes, não sendo necessário para a organização monitorar estas tarefas.

Figura 5.1- Modelo Analítico



Fonte: 1 Mazuco (2011).

Tabela 5.4 - Fatos do Modelo Analítico

Tabela Fato	Descrição
Fato_Componente	Armazena métricas (originais, revisadas e reais) de esforço e cronograma dos componentes.
Fato_Tarefa	Armazena métricas (originais, revisadas e reais) de esforço e cronograma das tarefas de um projeto.

Tabela 5.5 - Dimensões do Modelo Analítico

Tabela Dimensão	Descrição
Dim_Tempo_Inicial	Dados que caracterizam o tempo inicial de uma atividade.
Dim_Data_Validade_Inicial	Dados que caracterizam o tempo da validade inicial de um fato.
Dim_Projeto	Dados que caracterizam o projeto (categoria, e tipo – desenvolvimento ou manutenção).
Dim_Categoria	Dados que caracterizam a categoria do projeto (MPP, MGP, DPP, DGP).

Dim_Status	Dados que caracterizam o status de um componente (em andamento ou finalizado).
Dim_Tipo_Fato	Define se o fato é uma estimativa (<i>baseline</i> original, <i>baseline</i> revisado), registro de uma realização (real) ou registro das métricas derivadas do fato.
Dim_Componente	Dados que caracterizam um componente do projeto, bem como a sua categoria.
Dim_Tarefa	Dados que caracterizam uma tarefa do projeto.

Com relação às tabelas dimensão, é importante ressaltar a Dimensão *Dim_Tipo_Fato* que é composta por uma chave primária e um atributo categórico, contendo o tipo do fato: original (planejado), revisado (replanejado), real (efetivamente realizado) e métrica que contém os valores das métricas derivadas definidas no plano de métricas (Tabela 5.1). Assim, a partir dessa dimensão, é possível oferecer as quatro perspectivas de análise e, além disso, sumarizá-las em diferentes níveis.

É importante ressaltar a dimensão *Dim_Data_Validade_Inicial* pois esta contém a data da validade inicial de um fato, este conceito será detalhado na Seção 5.2.3.

5.2.3 Validade dos Fatos

Para oferecer suporte adequado à monitoração, a coleta de métricas deve ser realizada com frequência e latência na ordem de horas. Além disso, é necessário também disponibilizar estas métricas de maneira consistente no repositório de métricas. O repositório deve conter também o histórico destas métricas. Kimball sugere que sejam armazenados apenas dados novos ou atualizados no DW [KIM02]. Desta maneira, para atender a esta exigência, foi adotada a metodologia de validade dos fatos utilizada por Silveira [SIL07].

Foram definidos dois atributos: *data de validade inicial* e *data de validade final*, como limites inferior e superior de um intervalo. Um fato é válido no período compreendido entre as datas de validade inicial e final. Cada fato pode ter valor para data de validade inicial e valor nulo para a data de validade final no momento da sua criação.

Quando os dados são inseridos no DW a *data de validade inicial* recebe a data da carga e a *data de validade final* é nula, até que o fato deixe de ser válido e receba data de fim. Desta maneira o repositório não armazena dados repetidos e permite uma carga de maneira incremental de acordo com as regras para controlar inclusões, atualizações e término da validade dos fatos enquanto o projeto ainda está em andamento.

5.3 PROCESSO DE EXTRAÇÃO TRANSFORMAÇÃO E CARGA (ETC)

O processo de Extração, Transformação e Carga (ETC) é considerado em projetos de Data Warehouse como a etapa que exige maior esforço. Este processo é responsável por capturar os dados na fonte de origem, transformar e carregar os dados no repositório.

A arquitetura do sistema proposto, apresentada na Figura 5.2, é organizada em camadas: Aplicação, Integração dos dados e Apresentação. Esta seção descreve a solução a partir das camadas da arquitetura do sistema, estacando seus principais componentes: (i) a Camada de Aplicação juntamente com a rotina de extração; (ii) a Camada de Integração dos Dados, que contém a DSA e as rotinas de limpeza e transformação; (iii) a carga incremental dos dados consolidados de acordo com as regras estabelecidas; e (iv) o Repositório de Dados, implementado na forma de um DW e visualizado por intermédio de um cubo OLAP.

5.3.1 Rotina de Extração

O processo de extração é iniciado através de tarefas agendadas no banco de dados. No momento da execução, são selecionados os dados dos projetos que estejam na fase de execução ou que tenham sido encerrados no dia anterior, de acordo com o processo de desenvolvimento da organização apresentado na Seção 3.1.

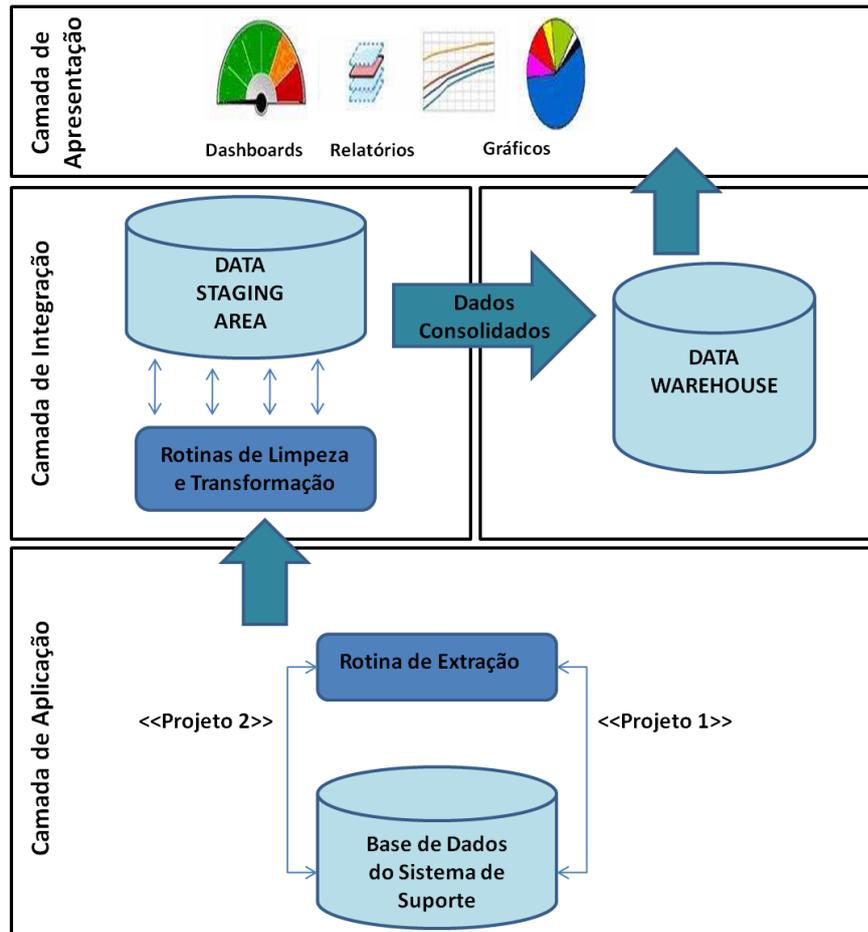


Figura 5.2 – Arquitetura do Sistema

O ideal é que a rotina de extração seja programada para executar após o expediente, garantindo que os dados do repositório estarão consistentes com as atividades realizadas no dia anterior. Desta forma, a coleta das métricas é executada de modo automatizado, com o mínimo de intrusão. Esta automatização do processo de extração é fundamental para diminuir a latência e possibilitar uma maior frequência de captura, essenciais a monitoração.

A rotina de extração também pode ser disparada pela ação de um determinado recurso, em qualquer momento do dia. Entretanto, o modelo analítico desenvolvido suporta apenas uma carga por dia. Por isso, quando a tarefa agendada for executada ao final do dia, os dados que estiverem no repositório serão sobrescritos.

5.3.2 Data Staging Area (DSA)

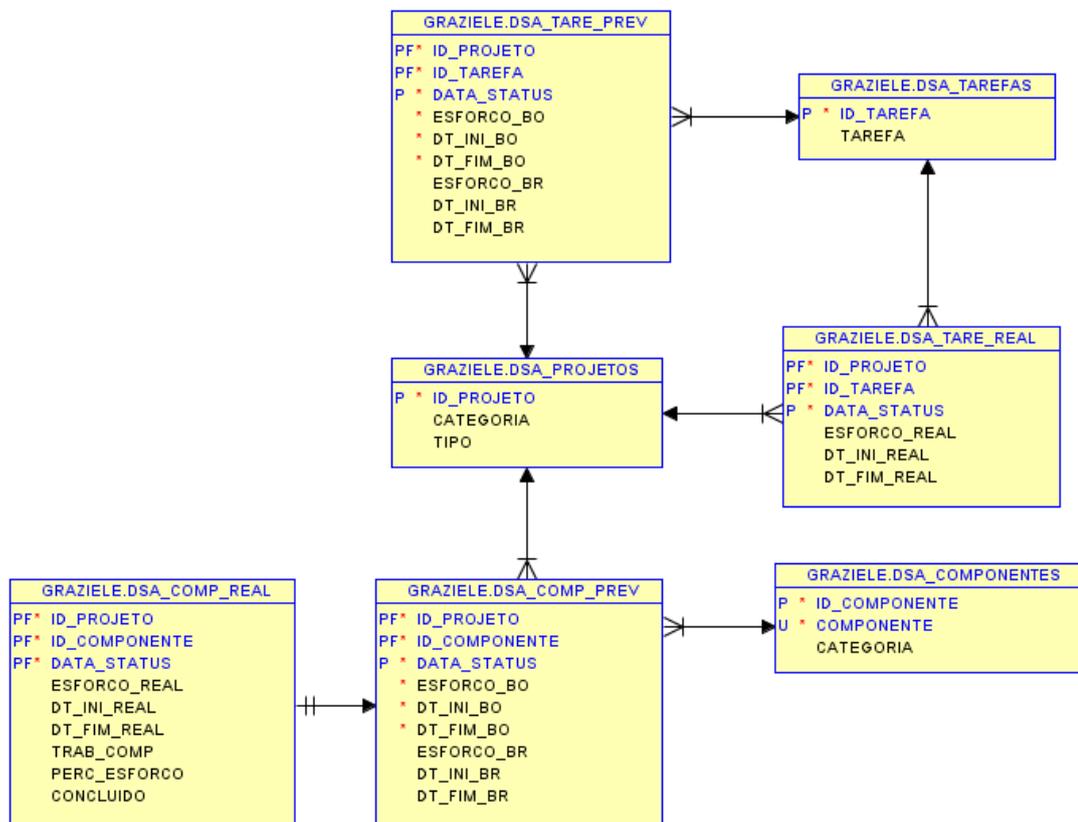
A camada de integração dos dados é composta pela DSA e o repositório de dados. Esta camada é responsável pela transformação e limpeza dos dados armazenados temporariamente na DSA.

O objetivo da DSA é minimizar a intrusão do processo de ETC das métricas. A DSA é composta por um conjunto de tabelas onde os dados brutos extraídos das fontes de origem devem ser armazenados. Depois os dados devem ser pré-processados pelas rotinas de limpeza e transformação (Seção 5.3.2.2), considerando o modelo analítico alvo descrito na Seção 5.2.

5.3.2.1 Estrutura da DSA

Com base nas características do ambiente de desenvolvimento, que apresenta uma base de dados homogênea e centralizada, foi desenvolvida uma DSA dedicada. Para isto, foi definida a estrutura das tabelas temporárias, incluindo seus atributos considerando o modelo analítico alvo. A Figura 5.3 apresenta a estrutura das tabelas que compõe a DSA.

Figura 5.3 - Estrutura da DSA



Fonte: Mazuco (2011).

Os dados que se encontram na DSA foram pré-processados durante a extração, transformações como a soma do número de horas realizadas (*esforço_real*) nos componentes e tarefas.

5.3.2.2 Rotinas de Limpeza e Transformação

Para realizar a limpeza e a transformação dos dados, foi desenvolvida a rotina *Popula_DSA*. Esta rotina é responsável também pela extração dos dados.

O processo de limpeza dos dados não é trabalhoso, pois o modelo de dados da organização é bem elegante e não houve necessidade de completar dados faltantes e nem verificar a integridade entre as chaves primária e estrangeira. A base de origem já trata destas validações.

O processo de transformação incluiu a formatação dos campos de data, o cálculo do percentual de esforço e do trabalho completado.

5.3.2.3 Carga do Repositório de dados

Visando oferecer suporte adequado a monitoração e manter o repositório de dados atualizado e consistente, comportando cargas freqüentes, esta solução propõe um procedimento de carga incremental. Com isso, é possível manter o histórico e o acompanhamento diário dos projetos. Para tanto, a carga incremental considera a validade dos fatos, de acordo com a definição apresentada na Seção 5.2.3.

Com base nos conceitos relacionados à validade dos fatos, as regras definidas por Silveira em [SIL07] foram adaptadas para o contexto deste trabalho.

- Regra 1: projetos encerrados, recebem como *data de validade inicial* (D_I) e *data de validade final* (D_F) a data da carga. Projetos em andamento, na primeira carga recebem a D_I como a data da carga e a D_F recebe o valor nulo.
- Regra 2: quando um fato atualizado, caracterizado por apresentar chave primária igual à previamente armazenada no DW, com exceção do valor que corresponde à data da carga e pelo menos uma das suas métricas com valor diferente, este deve ser inserido no DW recebendo como D_I a data da carga e como D_F o valor nulo. Neste momento o fato anterior a este (com a data da carga anterior) deve ser atualizado recebendo em D_F a data da carga.

- Regra 3: nas tabelas fato, apenas fatos que efetivamente começaram devem ser carregados, ou seja, apenas fatos com data de início diferente de nulo.

Para atender a estas três regras, foi desenvolvida a rotina *Popula_Fatos*, escrita em linguagem SQL. Foi desenvolvida também a rotina *Popula_Dimensoes* que seleciona os dados das tabelas da DSA destinados a popular as dimensões *Dim_Projeto*, *Dim_Componente* e *Dim_Tarefa*. Esta rotina também é responsável por popular a *Dim_Tempo* selecionando as datas distintas das tabelas da DSA. Esta rotina ainda popula as dimensões *Dim_Tipo_Fato* e *Dim_Status* a primeira com os valores: original, revisado e real e a segunda com os valores: em andamento e finalizado.

Estas rotinas podem ser agendadas como tarefas do banco de dados, a serem executadas logo após a rotina *Popula_DSA*.

5.4 CONSIDERAÇÕES SOBRE A SOLUÇÃO PROPOSTA

A solução descrita neste capítulo propõe um processo automatizado de ETC de métricas de *software* a serem armazenadas em um repositório de dados na forma de um DW.

O modelo analítico proposto nesta seção suporta tanto a análise de projetos já encerrados como a monitoração de projetos em andamento utilizando um programa de métricas que a abrange a técnica de *Earned Value Analysis* (EVA).

O suporte a monitoração é realizado através do tratamento da validade dos fatos ao longo do tempo, permitindo assim a carga incremental de dados para o repositório para que o projeto possa ser acompanhado dia a dia.

O processo de ETC pode ser automatizado através de tarefas agendadas no banco de dados.

6. TESTE DA SOLUÇÃO

Este capítulo descreve o conjunto de testes realizados sobre a solução proposta no Capítulo 5. Para tanto apresenta: (i) os objetivos e procedimentos de teste; (ii) os recursos computacionais, configurados para realizar o teste; (iii) a descrição dos dados utilizados para o teste; (iv) a estrutura do cubo OLAP construído, utilizado como interface de teste; e (v) análise dos resultados obtidos.

6.1 OBJETIVOS

Os testes realizados possuem como objetivo validar a solução descrita no Capítulo 5, ou seja, verificar se a solução permite realizar a análise dos projetos encerrados e monitorar os projetos em andamento com a técnica de EVA.

A organização disponibilizou sua base de dados para a realização dos testes, e os dados utilizados são de projetos reais. Os testes não foram realizados em ambiente organizacional, mas pela autora deste trabalho.

Dois tipos de teste foram efetuados: (i) com um projeto encerrado (análise) e (ii) com um projeto em andamento (monitoração). Os testes consistem basicamente, na execução do processo de ETC das métricas, a atualização do modelo analítico, a geração do cubo de dados e a verificação dos valores das métricas, com o auxílio da visualização das dimensões do cubo.

6.2 DESCRIÇÃO DOS RECURSOS COMPUTACIONAIS UTILIZADOS

Para a realização dos testes foi construído um ambiente com os seguintes recursos computacionais:

1. Banco de dados Oracle 11g;
2. O software *SQL Developer*, utilizado para a criação da base de dados, como para a geração dos diagramas da DSA e do modelo analítico;
3. A ferramenta da Oracle, *Analytic Workspace Manager (AWM)* utilizada para a geração dos cubos OLAP.

6.2.1 Oracle Analytic Workspace Manager (AWM)

O AWM é uma ferramenta utilizada para criar, desenvolver e gerenciar dados dimensionais em todo o ciclo de vida do desenvolvimento de um projeto no banco de dados Oracle. Com o AWM é possível criar um espaço de trabalho analítico e então adicionar dimensões e cubos OLAP.

A ferramenta possui algumas limitações quanto à criação do modelo analítico, por exemplo, não suporta o cálculo das métricas derivadas definidas no plano de métricas apresentado na Seção 5.1, além disso, o AWM também não permite ter como medida no cubo um atributo do tipo data. Para que a ferramenta pudesse ser utilizada, foram necessários alguns ajustes na solução proposta. Estes ajustes estão documentados na Seção 6.3.

6.3 DESCRIÇÃO DOS DADOS UTILIZADOS NO TESTES

Para que os testes pudessem ser realizados utilizando a ferramenta Oracle AWM [AWM11], os dados utilizados no teste, assim como o modelo analítico, precisaram de alguns ajustes.

As tabelas fato armazenam as métricas derivadas já calculadas pelo processo de ETC. A tabela *Fato_Componente* armazena as métricas referente a análise e acompanhamento dos projetos: VPBO, VPBR, VCBO, VCBR, VPA, VCA, IDC e IDP. Destas, métricas IDC e IDP são referentes à monitoração. A tabela *Fato_Tarefa* armazena apenas métricas referentes a análise, como descrito na Seção 5.2 as métricas armazenadas são: VPBO, VPBR, VCBO, VCBR e VPA.

Para o armazenamento destas métricas nas tabelas fato, foi criada mais uma entrada na tabela *Dim_Tipo_Fato*: Métricas. Esta entrada nas tabelas fato indica que naquele registro estão armazenadas as métricas do projeto.

Ainda sobre as métricas definidas no plano de métricas, é importante ressaltar que devido a uma particularidade dos dados da organização, foi adicionado o valor 1 no cálculo das métricas que calculam diferença de datas VPBO, VPBR, VPA e IDP. Isto foi necessário, pois nos dados extraídos dos projetos, muitos componentes são planejados para iniciar e terminar no mesmo dia, gerando divisões por zero caso

fosse utilizado o cálculo original. A Tabela 6.1 apresenta o cálculo das métricas referentes a prazo corrigidas.

Tabela 6.1 - Ajuste Métricas Sobre Prazo

Métricas Derivadas	Equação
Varição de Prazo do <i>Baseline</i> Original (VPBO)	$VPBO = \frac{(DFR - DFBO) + 1}{(DFBO - DIBO) + 1} \times 100$
Varição de Prazo do <i>Baseline</i> Revisado (VPBR)	$VPBR = \frac{(DFR - DFBR) + 1}{(DFBR - DIBR) + 1} \times 100$
Varição de Prazo Agregada (VPA)	$VPA = CBO \times \left(\frac{TC}{100} - \frac{(DS - DIBO) + 1}{(DFBO - DIBO) + 1} \right)$
Índice de Desempenho de Prazo (IDP)	$IDP = \frac{TC \times (DFBO - DIBO) + 1}{100 \times (DS - DIBO) + 1}$

Fonte: Mazuco (2011).

Com relação aos dados armazenados nas tabelas fato, as informações de *data final* e *data de validade final* foram armazenadas em formato numérico, ou seja, a data final é o cálculo do número de dias entre a data inicial e a data final. Esta adaptação foi necessária, pois como as duas informações podem ser nulas, não é possível conectá-las em dimensões com a ferramenta AWM. Então, caso a data final seja nula, o campo recebe 999 representando que a data é nula.

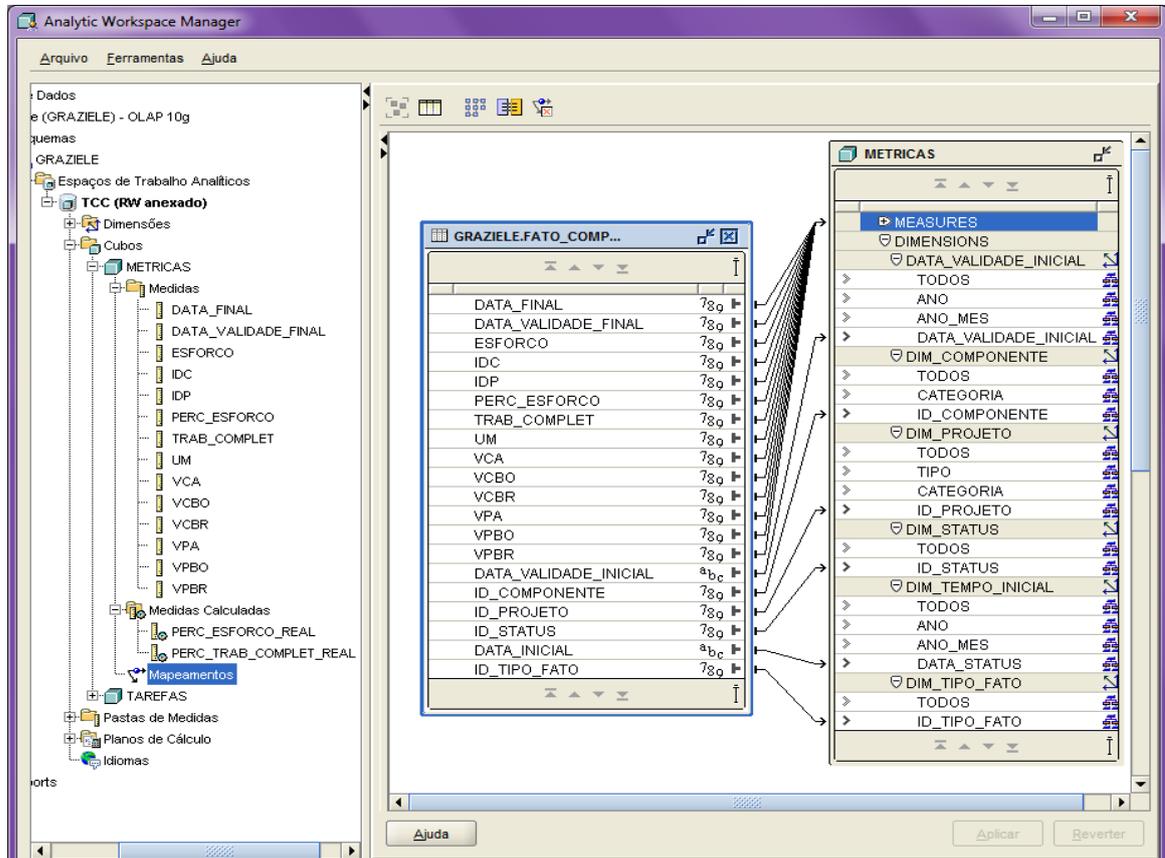
Quanto ao cálculo do percentual de trabalho completado, no trabalho de Silveira [SIL07] foi calculado de acordo com o peso de cada atividade. Não foi possível reproduzir este comportamento, por limitações da ferramenta AWM no cálculo das métricas. O percentual do trabalho completado é calculado sobre o percentual de esforço. Caso o percentual de esforço seja maior do que 100, e o componente está concluído, este possui 100 em %TC, caso contrário possui o valor do percentual de esforço.

6.4 CUBO OLAP

Para realização dos testes, foi criado o cubo OLAP, composto pelas tabelas fato e as dimensões propostas no modelo analítico (Seção 5.1), juntamente com a inclusão da *Dim_Validade_Inicial*, que determina se os valores fato são válidos no

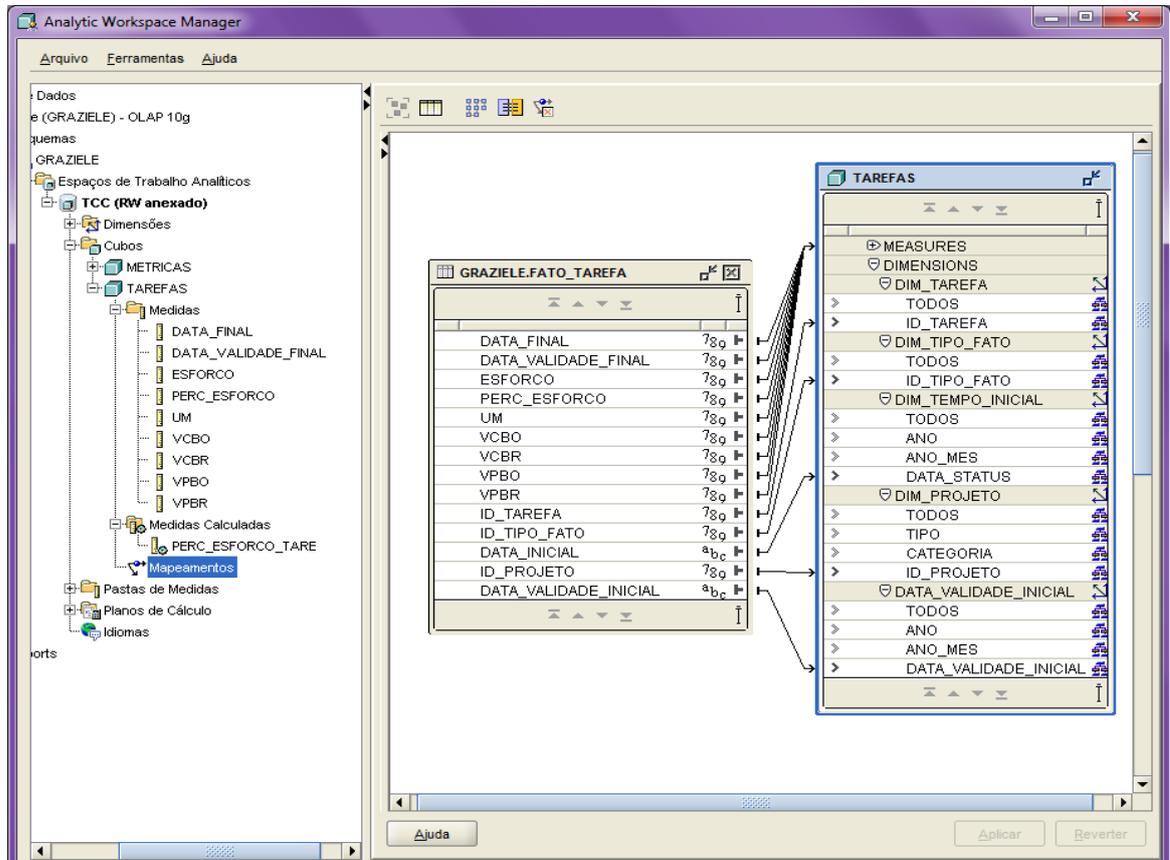
presente. A Figura 6.1 ilustra a tabela *Fato_Componente*, a Figura 6.2 ilustra a tabela *Fato_Tarefa* mapeada no AWM, bem como suas dimensões e medidas.

Figura 6.1 - Fato Componente no AWM



Fonte: Mazuco (2011).

Figura 6.2 - Fato Tarefa no AWM



Fonte: Mazuco (2011).

Para que o %TC e o percentual de esforço não seja acumulado na geração do cubo, foram criadas duas medidas calculadas dividindo este valor por 1 para estas medidas apresentem o valor mais próximo da realidade.

6.5 TESTE DE MONITORAMENTO E ANÁLISE

Para testar a capacidade de monitoramento e análise de projetos, foram realizadas consultas nas tabelas *Fato_Componente* e *Fato_tarefa*.

Primeiramente, o processo de ETC foi executado com os dados de um projeto, do tipo DGP (Seção 3.1). Foram realizadas duas cargas de dados, uma referente ao dia 04/10/2011 e outra referente ao dia 11/10/2011. Estas datas foram escolhidas a fim de que se possa visualizar melhor o andamento do projeto como um todo. É importante ressaltar que a carga de dados no cubo se dá de maneira incremental, ou seja, o mesmo cubo é capaz de suportar várias cargas de dados para o mesmo projeto.

A seguir são apresentados os resultados nas consultas das tabelas fato.

A Figura 6.3 apresenta os dados referente à carga realizada em 04/10/2011.

Figura 6.3 – Fato Componente em 04/10/2011.

		2011/10/04															
		Num Dias	Num Dias Valid	Esforc	Idc	Idp	Vca	Vcbo	Vcbr	Vpa	Vpbo	Vpbr	PERC_ESFORCO	TRAB_COMPLET			
230607	PROC8010	*	1.035,00	28,00	91,22	1,00	0,68	0,00	-71,95	-71,95	-6,56			28,04	28,04		
		Original	18,00	7,00	40,00												
		Revisado	18,00	7,00	40,00												
		Real	999,00	7,00	11,22										28,04	28,04	
		Métricas		7,00		1,00	0,68	0,00	-71,95	-71,95	-6,56						
		PROC8010	*	38,00	14,00	80,00											
		HLP	*	2,00	14,00	0,34											
		Relatorio	*														
		Transacao	*	38,00	210,00	48,94	5,62	1,06	-6,86	979,63	715,37	-22,51	775,00	775,00	201,77	100,00	
			*	3,00	28,00	3,22	1,80	0,00	0,56	-44,44	-44,44	-5,04	100,00	100,00	55,55	100,00	
		TRS0040	Original	1,00	7,00	1,26											
			Revisado	1,00	7,00	1,26											
			Real	1,00	7,00	0,70										55,55	100,00
			Métricas		7,00		1,80	0,00	0,56	-44,44	-44,44	-5,04	100,00	100,00			
		TRS0030	*	3,00	28,00	1,98	0,15	0,00	-1,29	560,87	560,87	-0,92	100,00	100,00	659,42	100,00	
			Original	1,00	7,00	0,23											
			Revisado	1,00	7,00	0,23											
			Real	1,00	7,00	1,52										659,42	100,00
			Métricas		7,00		0,15	0,00	-1,29	560,87	560,87	-0,92	100,00	100,00			
		TRS0050	*	3,00	28,00	1,08	0,37	0,00	-0,39	169,57	169,57	-0,92	100,00	100,00	268,11	100,00	
	Original		1,00	7,00	0,23												
	Revisado		1,00	7,00	0,23												
	Real		1,00	7,00	0,62										268,11	100,00	
		Métricas		7,00		0,37	0,00	-0,39	169,57	169,57	-0,92	100,00	100,00				

Fonte: Mazuco (2011).

Neste teste é possível observar os dados referentes às métricas de EVA. O resultado das métricas pode ser interpretado da seguinte forma para o componente PROC8010:

Analisando as métricas IDP, IDC, VCA e VPA, pode-se concluir que o custo do componente está dentro do orçamento planejado, porém o componente está atrasado quanto ao prazo. Isto indica que o custo diário planejado para este componente não está sendo atendido.

A Figura 6.4 ilustra a carga do projeto referente ao dia 11/10/2011, onde é possível observar o andamento do projeto. O componente PROC8010 que antes estava com 28,04% do trabalho completado, no dia 11/10/2011 está com 73,54% do trabalho completado. É possível notar também que a situação do componente referente ao custo e ao prazo é a mesma analisada na carga anterior.

Figura 6.4 - Fato Componente em 11/10/2011

		2011/10/11														
		Num Dias	Num Dias Valid	Esforço	Idc	ldp	Vca	Vcbo	Vcbr	Vpa	Vpbo	Vpbr	PERC_ESFORÇO	TRAB_COMPLET	Nu	
230607	PROC8010	*	51,00	3.996,00	109,42	1,00	0,89	0,00	-26,45	-26,45	-3,92	83,33	83,33	73,54	73,54	
		Original	18,00	999,00	40,00											
		Revisado	18,00	999,00	40,00											
		Real	15,00	999,00	29,42									73,54	73,54	
	Metricas		999,00		1,00	0,89	0,00	-26,45	-26,45	-3,92	83,33	83,33				
	PROC8010	*	38,00	1.998,00	80,00											
	HLP	*	2,00	1.998,00	0,34											
	Relatorio	*														
	Transacao	*	38,00	29.970,00	48,94	5,62	0,42	-6,86	979,63	715,37	-58,97	775,00	775,00	201,77	100,00	
	TRS0040	*	3,00	3.996,00	3,22	1,80	0,00	0,56	-44,44	-44,44	-13,86	100,00	100,00	55,55	100,00	
		Original	1,00	999,00	1,26											
		Revisado	1,00	999,00	1,26											
		Real	1,00	999,00	0,70									55,55	100,00	
	Metricas		999,00		1,80	0,00	0,56	-44,44	-44,44	-13,86	100,00	100,00				
	TRS0030	*	3,00	3.996,00	1,98	0,15	0,00	-1,29	560,87	560,87	-2,53	100,00	100,00	659,42	100,00	
		Original	1,00	999,00	0,23											
		Revisado	1,00	999,00	0,23											
		Real	1,00	999,00	1,52									659,42	100,00	
	Metricas		999,00		0,15	0,00	-1,29	560,87	560,87	-2,53	100,00	100,00				
	TRS0050	*	3,00	3.996,00	1,08	0,37	0,00	-0,39	169,57	169,57	-2,53	100,00	100,00	268,11	100,00	
Original		1,00	999,00	0,23												
Revisado		1,00	999,00	0,23												
Real		1,00	999,00	0,62									268,11	100,00		
Metricas		999,00		0,37	0,00	-0,39	169,57	169,57	-2,53	100,00	100,00					

Fonte: Mazuco (2011).

6.6 CONSIDERAÇÕES SOBRE OS TESTES

A partir dos resultados obtidos com exploração do cubo OLAP gerado para testar a solução, foi possível concluir que: i) modelo analítico proposto oferece suporte adequado à análise e à monitoração de projetos de *software*; ii) a solução apresenta de forma consistente as métricas propostas no programa de métricas, abrangendo adequadamente a técnica de EVA; iii) o processo de ETC é automatizado e possibilita cargas freqüentes, essencial à monitoração.

É importante mencionar que a construção do cubo OLAP, descrito na seção 6.4 foi fundamental para a geração de resultados e exploração das métricas. A utilização deste permitiu concluir que a solução proposta atende ao objetivo inicial, que é monitorar projetos de *software*.

7. CONSIDERAÇÕES FINAIS

Este trabalho propõe um ambiente para facilitar a gestão de métricas de um ambiente de desenvolvimento de *software*, visando atender a algumas exigências do CMMI quanto a medição e análise assim como monitoração e controle de projetos de *software*.

Para realizar este trabalho, foi proposto o armazenando destas métricas em um repositório estruturado na forma de um *Data Warehouse*. Para tanto, foi definido um modelo analítico e um programa de métricas que oferece suporte à monitoração de projetos de *software* segundo a técnica de *Earned Value Analysis* (EVA).

Para avaliar esta solução foi realizado um processo de testes que iniciou com a extração dos dados da sua fonte de origem, passou pela limpeza e transformação destes dados. Por fim, os dados foram carregados em um *Data Warehouse*, de onde foi possível gerar um cubo OLAP para a consulta dos dados armazenados.

Para a realização do teste da solução foi necessário alguns ajustes na solução proposta, descritos na Seção 6.2, pois a ferramenta disponível para realizar os testes é limitada e não realiza todos os cálculos que o cubo deveria fazer.

Com a realização dos testes foi possível verificar a adequação do programa de métricas para monitoração de projetos e a eficácia do processo de Extração, Transformação e Carga dos dados, permitindo a carga incremental dos dados no *Data Warehouse*.

Considera-se que o resultado obtido com este trabalho é satisfatório. Mesmo com a necessidade de efetuar ajustes para gerar o cubo OLAP na ferramenta AWM, foi possível mostrar que a solução proposta é eficaz para a monitoração de projetos de *software* atendendo assim a parte das exigências do CMMI para garantir qualidade do processo de desenvolvimento de *software*.

7.1 TRABALHOS FUTUROS

Os trabalhos futuros centram-se na implementação da solução proposta em uma ferramenta como o Oracle BIEE, que possui mais recursos para realizar a análise dos dados, como *dashboards*, e gráficos, facilitando a visualização dos dados.

A solução proposta neste trabalho poderia ainda com pequenos ajustes, atender a outros modelos de desenvolvimento de *software* que não o modelo cascata.

Finalmente outro trabalho futuro poderia ser desenvolvido na área de mineração de dados a fim de realizar a predição das métricas de *software*.

REFERÊNCIAS

- [AWM11] ORACLE. **Analytic Workspace Manager**. Disponível em: <http://www.oracle.com/technetwork/database/options/olap/index.html> Acesso em 20 de novembro de 2011.
- [BER09] BERARDI, R. **Avaliação de Qualidade de Dados de Métricas de Esforço Baseado em Data Provenance e Fuzzy Logic**. 137 f. Dissertação (Mestrado em Ciência da Computação) – Faculdade de Informática, PUCRS, Porto Alegre, 2009.
- [CKS04] CHRISSIS, M.; KONRAD, M.; SHRUM, S; **Understanding Model Representations and Levels: What Do They Mean?**, 2004. 35 slides. Presentation sponsored by the U.S. Department of Defense by Carnegie Mellon University.
- [FLE00] FLEMING, Q.; KOPPELMAN, J. **Earned Value Project Management**. 2nd edition, Pennsylvania, USA: Project Management Institute, 2000. 212 p.
- [HAN01] HAN, J.; KAMBER, M. **Data Mining: Concepts and Techniques**. San Francisco: Morgan Kaufmann, 2001. 550 p.
- [IEEE98] ANSI/IEEE Std 1061-1998. **IEEE Standard for a Software Quality Metrics Methodology**, Piscataway, NJ: IEEE Standards Dept., 1998. 26 p.
- [INM05] INMON, W. **Building the Data Warehouse**. Indianapolis, IN: John Wiley & Sons, Inc, 2005. 543 p.
- [KAJ01] 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):161176. Apud [BER09].
- [KAN03] KAN, S. H. **Metrics and Models in Software Quality Engineering**. Boston, Massachusetts, Addison Wesley, Second Edition, 528 p. 2003.
- [KIM02] KIMBALL, R. **The Data Warehouse Toolkit: the complete guide to dimensional modeling Second Edition**. New York, NY: John Willey & Sons, Inc, 2002. 436 p.
- [MCT11] **Portal do Ministério da Ciência e Tecnologia**. Disponível em: <http://www.mct.gov.br/> Acesso em: 08 de abril de 2011.
- [NOV06] NOVELLO, C. T. **Uma abordagem de Data Warehouse para Análise de Processos de Desenvolvimento de Software**. 153 f. Dissertação (Mestrado em Ciência da Computação) – Faculdade de Informática, PUCRS, Porto Alegre, 2006.

- [PMI08] Project Management Institute. **A Guide to the Project Management Body of Knowledge (PMBOK Guide)**. 4rd Edition. Newton Square: Project Management Intitute, 2008. 337 p.
- [SOF09] SOFTEX – **Melhoria de Processos do Software Brasileiro**. Disponível em: http://www.softex.br/mpsbr/_guias/guias/MPS.BR_Guia_Geral_2009.pdf f Acesso em: 02 de abril de 2011.
- [PRE01] PRESSMAN, R. **Software Engineering: A practitioner's Approach**. New York, New York, McGraw-Hill series in computer science; 5th edition, 860 p., 2001.
- [REZ09] REZENDE, B. **Uso de KDD na Predição de Métricas em um Processo de Desenvolvimento de Software**. 98 f. Monografia (Bacharelado em Ciência da Computação) – Faculdade de Informática, PUCRS, Porto Alegre, 2009.
- [SIL07] SILVEIRA, P. **Processo de ETC orientado a serviços para um ambiente de gestão de PDS baseado em métricas**. 168 f. Dissertação (Mestrado em Ciência da Computação) – Faculdade de Informática, PUCRS, Porto Alegre, 2007.
- [SEI10] SEI - Software Engineering Institute. **CMMI for Development**, Version 1.3. Pittsburgh: Carnegie Mellon University and Software Engineering Institute, 2010. Disponível em: <http://www.sei.cmu.edu/reports/10tr033.pdf> Acesso em: 27 março 2011.
- [SOM04] SOMMERVILLE, I. **Software Engineering**. 5th edition. Boston: Addison-Wesley, 2004. 592 p.
- [TEN10] TENORIO, N. **Modelo–E10: Um Modelo para Estimativas de Esforço em Manutenção de Software**. 134 f. Tese (Doutorado em Ciência da Computação) – Faculdade de Informática, PUCRS, Porto Alegre, 2010.
- [TUR09] TURBAN, E. [et al] **Business Intelligence: um enfoque gerencial para a inteligência do negócio**. Porto Alegre: Bookman 2009, 256 p.