



Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Engenharia de Software

Uma proposta de coleta e visualização de métricas de custo, tamanho e esforço, em projetos de software ágeis, com apoio de ferramenta Data Warehousing

Autor: Fernando de Souza Santos
Orientador: Mestre Hilmer Rodrigues Neri

Brasília, DF
2013



Fernando de Souza Santos

Uma proposta de coleta e visualização de métricas de custo, tamanho e esforço, em projetos de software ágeis, com apoio de ferramenta Data Warehousing

Monografia submetida ao curso de graduação em (Engenharia de Software) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia de Software).

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Mestre Hilmer Rodrigues Neri

Brasília, DF

2013

Fernando de Souza Santos

Uma proposta de coleta e visualização de métricas de custo, tamanho e esforço, em projetos de software ágeis, com apoio de ferramenta Data Warehousing/
Fernando de Souza Santos. – Brasília, DF, 2013-

92 p. : il. (algumas color.) ; 30 cm.

Orientador: Mestre Hilmer Rodrigues Neri

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB
Faculdade UnB Gama - FGA , 2013.

1. AgileEVM. 2. Data Warehouse. I. Mestre Hilmer Rodrigues Neri. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Uma proposta de coleta e visualização de métricas de custo, tamanho e esforço, em projetos de software ágeis, com apoio de ferramenta Data Warehousing

CDU

Fernando de Souza Santos

Uma proposta de coleta e visualização de métricas de custo, tamanho e esforço, em projetos de software ágeis, com apoio de ferramenta Data Warehousing

Monografia submetida ao curso de graduação em (Engenharia de Software) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia de Software).

Trabalho aprovado. Brasília, DF, 18 de dezembro de 2013:

Mestre Hilmer Rodrigues Neri
Orientador

Doutor Paulo Roberto M. Meirelles,
UnB/ FGA
Convidado 1

Doutor Luiz Augusto Fontes
Laranjeira, UnB/ FGA
Convidado 2

Brasília, DF
2013

*Esse trabalho é dedicado às mulheres da minha vida: Alice, Ana, Beatriz, Camila,
Leticia, Lucimar; e ao meu pai, Valdivino.*

Agradecimentos

Agradeço primeiramente a Deus. Sou grato a Deus pelo dom da vida, pelo seu amor e misericórdia infinitos, e por tudo que sou. É ele que me dá forças para continuar perante as adversidades da vida e que me ajuda a perseverar a ser uma pessoa melhor sempre.

À minha filha, fonte de inspiração, motivação, e felicidade plena.

Aos meus pais, responsáveis por todos os valores morais e cristãos que tenho vivido ao longo de minha vida. Foram eles que me motivaram a fazer as minhas escolhas de acordo com minha vontade, sempre me apoiando e demonstrando seu carinho e admiração por mim. À minha mãe um agradecimento especial, pois ela me ensinou a enfrentar a vida com alegria e fé, e a se dedicar a toda e qualquer coisa que se faça na vida. E ainda ensina todos os dias.

À mãe da minha filha e noiva, Camila, que sempre está ao meu lado me incentivando a sempre fazer melhor as coisas as quais me comprometo. Que com todo seu carinho e admiração que tem por mim me faz crer a cada dia que fiz a escolha certa, e tenho certeza que não poderia casar com outra mulher a não ser ela.

A minhas avós, tios, tias e primos, que mesmo à distância acreditam em mim e sempre estão por perto.

Aos meus amigos de curso que estão comigo na caminhada e que juntos vamos concluir mais uma etapa na vida, em especial ao Eusyar, Herbert, Greg, João, Ricardo, Luiz, entre outros.

Aos meus amigos de infância que me atuaram a tanto tempo e que com sua simplicidade não me deixam esquecer que amigos de verdade existem e vale a pena ter muitos. Em especial a Rayane, ao Klaus, Attílio, Flávio, Bruno, Guilherme.

Agradeço aos professores da UnB – Campus Gama que de alguma forma contribuíram para a minha formação acadêmica, profissional e pessoal durante esses longos anos na universidade e principalmente lutando diariamente por condições melhores de estudo, de infraestrutura e qualidade na educação.

E ao professor Hilmer, que com muita atenção, paciência e zelo dedicou seu valioso tempo a me orientar nesse trabalho.

A suprema felicidade da vida é a convicção de ser amado por aquilo que você é, ou melhor, apesar daquilo que você é. [Os Miseráveis – Victor Hugo, 1862]

Resumo

O gerenciamento de projeto é de fundamental importância, pois permite ao gerente monitorar o andamento do projeto nos pontos de interesse da organização, do cliente e das pessoas envolvidas. Entre as métricas de software de produto, processo e de projeto, as de projeto são exploradas para que seja possível manter o projeto sob controle em termos de custo, tempo e escopo. A comparação entre os métodos de gerenciamento tradicional e ágil, PMBOK e SCRUM, respectivamente, fez-se necessária para o estudo das técnicas de Valor Agregado em projetos tradicionais e de Valor Agregado em projetos Ágeis. A aplicação da técnica de Valor Agregado Tradicional em projetos ágeis esbarra em pressupostos que vão na contramão dos valores defendidos pelo manifesto ágil. Logo, utilizando-se da técnica de Valor Agregado Ágil para projetos ágeis de desenvolvimento de software é possível acompanhar o desempenho e progresso do projeto através do monitoramento dos custos. Em posse desses dados, é possível utilizar um ambiente de Data Warehouse para automatizar o processo de extração, transformação, carga (ETL) e visualização de custos oriundas da produção do software.

Palavras-chaves: AgileEVM. Data Warehouse. Gerenciamento de Projeto. SCRUM. Métodos Ágeis.

Abstract

The project management is crucial because it allows the manager to monitor the progress of the project in points of interest of the organization, the customer and the people involved. Between software metrics' that are categorized in product, process and project, the projects one are explored so that you can keep the project on track in terms of cost, time and scope. The comparison between the traditional and Agile management methods, SCRUM and PMBOK, respectively, it is necessary to study the techniques of Earned Value Management in traditional projects and Agile Earned Value Management in Agile projects. The application of the Tradicional Earned Value technique in Agile projects faces on assumptions that go against the values espoused by the Agile Manifesto. Thus, using the AgileEVM technique for Agile software development projects, it is possible to monitor the performance and progress of the project by monitoring costs. With these data, it is possible to use a Data Warehouse environment to automate the process of extraction, transformation, loading (ETL) and visualization of costs derived from the production of software.

Key-words: AgileEVM. Data Warehouse. Project Management. SCRUM. Agile Methods.

Lista de ilustrações

Figura 1 – Nível típico de custos e pessoal ao longo do seu ciclo de vida. Fonte (PMI, 2008)	32
Figura 2 – Mapeamento de grupos de processos de gerenciamento de projetos e áreas de conhecimento. Fonte (VARGAS, 2009)	32
Figura 3 – Grupo de Processos de Gerenciamento de Projetos Fonte (VARGAS, 2009)	33
Figura 4 – Esqueleto SCRUM. Fonte (SCHWABER, 2004)	36
Figura 5 – Relacionamento entre as métricas de qualidade. Fonte (ISO/IEC, 2011)	40
Figura 6 – Gerenciamento do Valor Agregado. Fonte (PMI, 2008)	42
Figura 7 – Componentes de um DWing. Fonte (KIMBALL; ROSS, 2002)	50
Figura 8 – Arquitetura de um DWing. Fonte (KIMBALL; ROSS, 2002)	53
Figura 9 – (a)Esquema Estrela. (b) Esquema Floco de Neve. (c) Esquema Constelação de Fatos. Fonte (BALLAR et al., 2006)	56
Figura 10 – Cubo 3-D. Fonte (HAN; KAMBER,)	57
Figura 11 – Arquitetura do Pentaho BA.	67
Figura 12 – Quadro Kanban do Projeto COMIGRAR	68
Figura 13 – Visão Arquitetural do ambiente <i>DW</i>	69
Figura 14 – Tabelas Dimensão do <i>DW</i>	71
Figura 15 – Tabelas Fato do <i>DW</i>	72
Figura 16 – AgileEVM <i>DW</i>	72
Figura 17 – Funcionalidades do projeto COMIGRAR.	74
Figura 18 – Histórias de Usuário do projeto COMIGRAR.	74
Figura 19 – Release 1 com uma Sprint	75
Figura 20 – Release 2 com uma Sprint	75
Figura 21 – Release 3 com duas Sprints.	76
Figura 22 – Lista de profissionais do time Scrum	76
Figura 23 – Transformação da Dimensão Projeto, História e Tarefa	77
Figura 24 – Transformação da Fato AgileEVM	77
Figura 25 – Job do Projeto COMIGRAR.	78
Figura 26 – Valor Planejado, Valor Agregado, Custo Atual e BAC por Release gráfico de linhas	78
Figura 27 – Valor Planejado, Valor Agregado e Custo Atual por Release, gráfico de barras	79
Figura 28 – Indicadores de Desempenho - CPI e SPI	79
Figura 29 – ETL Dimensão Usuário	89
Figura 30 – ETL Fato Esforço	89

Figura 31 – ETL Fato Tamanho	89
Figura 32 – Porcentagem da representatividade do BAC por Release	91
Figura 33 – BAC acumuludado do projeto COMIGRAR	91
Figura 34 – Variações de prazo e escopo	91
Figura 35 – Fato Tamanho: Pontos Planejados, Concluídos e Adicionados	92
Figura 36 – Fato Esforço: quantidade de horas trabalhadas por Sprint	92
Figura 37 – Estimativas por Release	92

Lista de tabelas

Tabela 1 – Elementos Básicos EVM Tradicional	43
Tabela 2 – Elementos Derivados do EVM Tradicional	44
Tabela 3 – Comparação dos termos do EVM	45
Tabela 4 – Parâmetros iniciais da release	46
Tabela 5 – Métricas para calculo do AgileEVM	46
Tabela 6 – Parâmetros Complementares	46
Tabela 7 – Definições e Equações AgileEVM	47
Tabela 8 – Comparação Redmine x Icescrum	64
Tabela 9 – Análise da comparação entre Redmine x Icescrum	65
Tabela 10 – Comparação Pentaho Business Analytics x SpaboBI	66
Tabela 11 – Análise da comparação entre Pentaho x SpagoBI	67
Tabela 12 – Processos de Negócio	70
Tabela 13 – Tipos de Tabela Fato	73

Sumário

1	Introdução	23
1.1	Contextualização	23
1.2	Justificativa	25
1.3	Problema	26
1.4	Objetivos	27
1.4.1	Objetivo Geral	27
1.4.2	Objetivos Específicos	27
2	GERENCIAMENTO DE PROJETO DE SOFTWARE	29
2.1	Contextualização	29
2.2	Gerenciamento Tradicional	30
2.2.1	O guia PMBoK	31
2.2.1.1	Ciclo de vida e a organização do projeto	31
2.2.1.2	Grupos de Processos e Áreas de Conhecimento	33
2.3	Gerenciamento Ágil	34
2.3.1	SCRUM	36
2.3.1.1	Papéis SCRUM	37
2.3.1.2	O Fluxo de Trabalho	37
3	MÉTRICAS: UMA VISÃO SOBRE AS MEDIDAS DE CUSTO, ES- COPO E TEMPO	39
3.1	CONTEXTUALIZAÇÃO	39
3.2	EVM TRADICIONAL	41
3.3	AgileEVM	45
4	DATA WAREHOUSE COMO APOIO À ATIVIDADES DE MEDIÇÃO	49
4.1	Componentes de um ambiente de DWing	50
4.1.1	Sistemas de Fonte de Dados Operacionais - (OSS, sigla em inglês)	51
4.1.2	Área de armazenagem de dados (DSA, sigla em inglês)	51
4.1.2.1	ETL	51
4.1.3	Área de Apresentação dos dados	52
4.1.4	Ferramentas de Acesso aos Dados - (DAT, sigla em inglês)	52
4.2	Modelagem Dimensional	52
4.2.1	Arquitetura	52
4.2.2	Tabela Fato	54
4.2.3	Tabelas Dimensão	55
4.3	Processo de Modelagem Dimensional	55

4.3.0.1	Esquema estrela	56
4.3.0.2	Esquema Floco de Neve	56
4.3.0.3	Esquema Constelação de Fatos	56
4.4	OLAP	57
5	COMIGRAR – Um estudo de Caso	61
5.1	Definição do Estudo de Caso COMIGRAR	61
5.1.1	Objetivo	61
5.1.2	O caso	61
5.1.3	Teoria	62
5.1.4	Questão de pesquisa	62
5.1.5	Métodos	62
5.1.6	Estratégia selecionada	63
5.2	Procedimentos de coleta de dados	63
5.3	Executar o estudo de caso com coleta de dados	63
5.3.1	Ferramenta de Gerenciamento de Projeto	63
5.3.2	Icescrum	65
5.3.3	Ferramenta de suporte a ambiente <i>DWing</i>	65
5.3.4	Pentaho Business Analytics	67
5.3.5	Fontes de Coleta	67
5.3.6	Modelagem do Esquema Constelação de Fatos do <i>DW</i>	68
5.3.6.1	Arquitetura DW	68
5.3.6.2	Análise do Processo de Negócio	68
5.3.6.3	Granularidade	70
5.3.6.4	Dimensões	71
5.3.6.5	Fatos	71
5.3.6.6	AgileEVM <i>Data Warehouse</i>	72
5.3.7	Coleta de Dados	73
5.3.7.1	ETL	76
5.4	Analisar dados coletados	78
5.5	Reportar	78
5.5.1	Resultados do Estudo de Caso - COMIGRAR	80
6	Conclusão	81
6.1	Trabalhos Futuros	82
	Referências	83

Apêndices	87
APÊNDICE A – ETL	89
APÊNDICE B – Gráficos do Negócio	91

1 INTRODUÇÃO

1.1 Contextualização

O gerenciamento de projeto é uma atividade de fundamental importância em qualquer organização, pois esta atividade proporciona ao gerente ¹, e também a equipe, uma visão ampla do andamento do projeto nos pontos estratégicos de interesse da organização, do cliente e das pessoas envolvidas. É importante observar que o gerenciamento de projetos de desenvolvimento de software se sobrepõe com sua própria gestão da produção. (PMI, 2008)

Os projetos e seu gerenciamento são executados em um ambiente mais amplo que o do projeto propriamente dito. A compreensão desse contexto mais amplo ajuda a garantir que o trabalho seja conduzido em alinhamento aos objetivos da empresa e gerenciado de acordo com as metodologias e práticas estabelecidas pela organização. (PMI, 2008)

A norma ISO/IEC (2007) define quatro atividades de um processo de medição: 1) Estabelecer e sustentar o compromisso de medição; 2) Planejar o processo de medição; 3) Realizar o processo de medição; e 4) Avaliar a medição. Processo esse que permite o monitoramento da execução do projeto, que fornece informações importantes para a tomada de decisão, envolvendo a identificação e realização de ações corretivas e preventivas. (BARCELLOS, 2010)

As métricas de software podem ser classificadas em três categorias: 1) de processo, que tem por objetivo controlar o processo de desenvolvimento; 2) de produto, relacionam com as características inerentes do produto desenvolvido; e, 3) de projeto, que monitora o projeto considerando atributos como o tamanho do time, produtividade, tempo e esforço. (SILVEIRA, BECKER e RUIZ (2012 apud KAN, 2002)

A técnica de gerenciamento do valor agregado (EVM, sigla em inglês) é amplamente difundida e utilizada pelas organizações para medir desempenho de projetos, tradicionalmente, em função de seus custos. (VARGAS, 2002)

Medidas de custo, tempo e escopo, provêm ao gestor, a visão consolidada sobre o andamento da execução projeto. Essas três dimensões-chaves (medidas) utilizadas para monitorar, também são conhecidas como a tríplice restrição em projetos e permeando essas três dimensões, há a dimensão da qualidade. (VARGAS, 2011)

A aplicação do EVM em projetos ágeis, ao menos em sua forma clássica, esbarra

¹ Um gerente de projeto ágil, ScrumMaster, tem perfil de líder e incentiva o time a alcançar seu maior nível de desempenho, participando do processo. (MOUNTAINGOAT, 2013)

em pressupostos que vão na contramão dos valores e princípios adotados nos métodos ágeis (BECK et al., 2001) e com isso, surgiu a necessidade de adaptar sua utilização neste contexto, dando origem ao AgileEVM (sigla em inglês). (SULAIMAN; BARTON; BLACKBURN, 2006)

O AgileEVM utiliza-se dos mesmos conceitos do EVM tradicional, e, aplicado ao contexto de práticas ágeis no desenvolvimento de projetos de software, é capaz de colaborar para a acurácia de métricas relacionadas ao gerenciamento, em particular nas categorias de métricas de processo e projeto.

Os gerentes de projetos têm necessidade de tempestividade e eficiência na interpretação de dados, para que seja possível tomar decisões mais consistentes ao longo da execução do projeto. Essas decisões podem ser melhor sistematizadas ao se utilizar um ambiente de *Datawarehousing-DWing*, apoiado por uma ferramenta que forneça e permita a manipulação das informações, facilitando as etapas de coleta, armazenamento, acesso aos dados e visualização. (SILVEIRA; BECKER; RUIZ, 2012)

Um banco de dados tradicional armazena dados num determinado período onde são registrados um dado por vez, operações pré-definidas são executadas, possuem restrições quanto a demanda do banco de dados, e geralmente não armazenam dados históricos, e sim, atualizam os registros. (KIMBALL; ROSS, 2002) Segundo OLIVEIRA (2007), hoje em dia, os diversos sistemas de operação, cada vez mais sofisticados, registram grandes volumes de dados sobre diversas áreas da organização. Acontece que, no momento da tomada de decisão, essas informações estão dispersas em diferentes fontes, o que acaba dificultando, ou até inviabilizando a análise gerencial. Essa dificuldade de acesso e visualização de medidas oriundas da produção do software, pode gerar prejuízos devido às oportunidades perdidas e decisões errôneas, ou equivocadas.

Já um banco de dados de *Data Warehouse*(*DW*, sigla em inglês), componente de um ambiente de *Data Warehousing* (*DWing*, sigla em inglês), armazena dados históricos e analíticos voltados à tomada de decisões. *DWing* é um ambiente composto pelo *DW* mais ferramentas para manipulação de seu back-end e front-end. (NERI, 2002) Isso determina o processamento e armazenamento grande volume de dados, podendo envolver consultas complexas que acessem um grande número de dados, exigindo assim um eficaz sistema de acesso a dados. (OLIVEIRA, 2007) Segundo OLIVEIRA (2007) O *Data Warehouse* é uma solução que procura de maneira flexível e eficiente tratar grandes volumes de dados e obter informações que auxiliem o processo para tomada de decisão.

O uso de um ambiente de *DWing*, aplicado no contexto de medidas da produção de software, tem se mostrado viável, conforme é possível observar em (CHULANI et al., 2003) (CASTELLANOS et al., 2005) (PALZA; FUHRMAN; ABRAN, 2003) (FOLLECO et al., 2007) (SILVEIRA, 2007)

Diante disso, este trabalho descreve uma solução que viabiliza a coleta e visualização de medidas de custo, escopo e tempo, em projetos ágeis de software, com suporte de ferramenta de *DW*.

Este trabalho de conclusão de curso está organizado da seguinte maneira: no Capítulo 2, são apresentados os conceitos relativos ao Gerenciamento de Projetos, onde ocorre uma comparação entre as características de projetos que utilizam uma abordagem tradicional de gerenciamento e aqueles que utilizam um abordagem ágil. Para esta são explicados os conceitos do Scrum, e para aquela, os conceitos do Guia PMBoK.

No capítulo 3 são discutidos os conceitos relacionados à medição e métricas de projetos de software, momento em que são apresentadas, e também comparadas, as características da aplicação da técnica de EVM em sua forma tradicional e sua adaptação para contexto de projetos ágeis, o AgileEVM.

No capítulo 4 são apresentados os conceitos de uma solução para o problema elucidado na seção anterior. Esses conceitos são de um *Data Warehouse*, onde são apresentados os componentes, arquitetura, ambiente e modelagem de um *DW*.

No capítulo 5 é apresentada a implementação de uma solução, utilizando dados de um projeto real, evidenciando a viabilidade técnica de seu uso.

E por fim no capítulo 6 é apresentada a conclusão onde é feita a discussão dos resultados, além da proposição de trabalhos futuros.

1.2 Justificativa

A busca por informações confiáveis, de forma tempestiva, e que retratem a realidade do ambiente de produção de software, além da necessidade de se correlacionar e analisar métricas de diferentes medidas, com vistas à tomada de decisão gerencial, é beneficiada pelo uso de um ambiente de *DWing*.

A definição mais difundida entre os pesquisadores apresenta o *DW* com um conjunto de dados integrados, consolidados, históricos, segmentados por assunto, não-voláteis, variáveis em relação ao tempo, e de apoio às decisões gerenciais.(INMON, 2005)

Do ponto de vista do gerente (líder), dado as características deste papel, que se espera é haja um painel situacional, onde as métricas oriundas da produção do software, ao serem interpretadas, sejam capazes de fornecer melhores insumos para as tomadas de decisões técnicas e gerenciais, por todo time.

Nesse contexto, métricas de escopo, tempo e custo são métricas tipicamente de gerenciamento, se forem corretamente coletadas, visualizadas e conseqüentemente analisadas, são capazes de fornecer importantes dados gerenciais sobre a situação da produção de um determinado projeto. Embora as métricas gerenciais devam ser combinadas com

as métricas de produto, de satisfação do cliente, para que uma análise gerencial seja mais eficaz, este trabalho se propõe a endereçar as questões relacionadas a métricas gerenciais de projeto.

O desenvolvimento de software baseado em métodos ágeis, neste trabalho em particular o SCRUM, possui o foco na qualidade do produto e no valor de negócio deste para o cliente. Nesse contexto deve-se utilizar métricas que representem melhor esse domínio, sem contudo, deixar de observar e analisar as métricas de gerenciamento.

Porém, tão importante quanto a escolha das métricas a serem coletadas, é fundamental que haja um correto alinhamento entre os interesses do cliente, do time, do líder de desenvolvimento e do gerente de projetos, ou seja, esses interesses devem coexistir de forma balanceada.

Outro fator relevante a se destacar está no ponto de minimização da interferência humana na coleta das métricas gerenciais, tornando essa atividade menos intrusiva possível, de forma a não onerar ou desviar o foco do time de desenvolvimento com a atividade de coleta das métricas.

1.3 Problema

Existe uma necessidade em minimizar a interferência humana na coleta de métricas de gerenciamento de projeto e também uma dificuldade em visualizar dados relacionados a produção do software. Este problema pode estar associado à não utilização de ferramentas que favoreçam a automatização do processo de coleta de métricas. Outro sim, ao fato das métricas estarem em diferentes ferramentas (repositório de código, testes, custo, tempo, escopo, e demais). Isso dificulta o armazenamento em um repositório centralizado e a visualização destas de forma a apoiarem as decisões gerenciais e técnicas na tempestividade necessária aos tomadores de decisão.

Considerando os pontos apresentados, adotou-se como questão de pesquisa a seguinte pergunta: um ambiente de *DWing* facilita a coleta, armazenamento e visualização de métricas de gerenciais de custo, tempo e escopo? Afirmações a serem alcançadas:

- Qual uma possível solução para automatizar a coleta das métricas gerenciais custo, tempo e escopo?
- A ferramenta iceSCRUM satisfaz a utilização da técnica AgileEVM?
- É possível integrar as ferramentas IceSCRUM, planilha eletrônica e Pentaho para dar suporte à interpretação de métricas de gerenciamento de projeto de software?

1.4 Objetivos

1.4.1 Objetivo Geral

Implementar uma proposta de solução que permita a coleta automática e semiautomática de métricas de gerenciamento; o armazenamento em um repositório centralizado; definição de um cubo de dados; uma forma de visualização destas; com o suporte de um ambiente de *DWing*.

1.4.2 Objetivos Específicos

1. Descrever as características do gerenciamento de projetos tradicionais e projetos ágeis;
2. Utilizar a adaptação da técnica tradicional de monitoramento de projetos, EVM, para o domínio de projetos ágeis
3. Descrever as principais características da aplicação da técnica de análise de valor agregado nos contextos de projetos tradicionais e ágeis;
4. Utilizar ferramentas de software livre;
5. Coletar dados de um projeto real de desenvolvimento de software.

2 GERENCIAMENTO DE PROJETOS DE SOFTWARE

2.1 Contextualização

Gerenciar um projeto é aplicar conhecimentos, habilidades, ferramentas e técnicas às atividades do projeto a fim de atender aos requisitos de negócio. O gerenciamento inclui: identificar requisitos; adaptação às diferentes necessidades à medida que o projeto é planejado e realizado; balancear as restrições conflitantes do projeto que incluem, mas não se limitam a: escopo, qualidade, tempo, custo. (IEEE, 1998)

O empreendimento de um esforço temporário para criar um produto, serviço ou resultado exclusivo é denominado de projeto e possui um fluxo de atividades com início, meio e fim, cujo resultado é único. A crescente aceitação do gerenciamento de projetos indica que a aplicação desses elementos adequadamente pode ter um impacto significativo no sucesso de um projeto. (PMI, 2008)

A área de gerenciamento de projetos é uma das áreas que mais cresce em utilização no mundo, sendo, hoje, objetivo de investimento em capacitação e metodologia pela maioria das empresas. Em todo projeto, é senso comum que uma das principais dificuldades está na medição e na avaliação dos resultados obtidos, sejam eles resultados finais ou durante sua execução, relacionados a prazos, custos, qualidade, escopo, risco e outros. (VARGAS, 2011)

VARGAS (2009), elucida que a proposta do gerenciamento de projetos é estabelecer um processo estruturado e lógico para lidar com eventos que se caracterizam pela novidade, complexidade e dinâmica ambiental. Um dos fatores que impulsionam o gerenciamento de projetos é o crescimento da competitividade: quem for mais rápido e competente certamente conseguirá os melhores resultados. Essa competitividade incita as empresas a conseguirem resultados com menos recursos, em um menor tempo e com mais qualidade.

Perguntas tais como: 1) Quanto tempo levará para desenvolver o software?; 2) Quanto custará o desenvolvimento do software?; dentre outras, são algumas das perguntas cujas respostas todo patrocinador quer saber, pois antes de comprometer os recursos destinados a um projeto, o patrocinador deseja ter uma estimativa de prazo e custo. (PFLEEGER, 2004)

Para se aplicar os conceitos de controle de processo faz-se necessário diferenciar duas modalidades de metodologia, não concomitantes, aplicáveis ao gerenciamento de

projetos: metodologia com processo definidos (ou prescritivos) e a metodologia empírica.

De acordo com [MARTINS \(2007\)](#), a primeira define o contexto do projeto e determina o escopo, custo, tempo e qualidade que se consolidam em um detalhado plano, onde as entradas são estabelecidas logo no início do projeto.

Já a segunda, define as entregas de forma abrangente e superficial, começando por um contexto inicial, que evolui e se adapta ao longo da execução. Seus pilares são: transparência – os aspectos do processo que afetam o resultado final devem ser conhecidos e estarem visíveis para aqueles que controlam o processo; inspeção – vários aspectos do processo devem ser inspecionados para a detecção de variações; adaptação – surge com frequência mudanças no processo e nos recursos do processo que devem ser ajustados para minimizar maiores desvios. ([SCHWABER, 2010](#))

2.2 Gerenciamento Tradicional

A metodologia com processos definidos e prescritivos, também chamada de abordagem tradicional, é a mais adequada em situações onde os passos a serem executados, em geral, são conhecidos, como por exemplo, na implantação de uma infraestrutura de TI. Em projetos tradicionais, certo conjunto de entradas produzirá um conjunto específico de saídas. ([MARTINS, 2007](#))

Frederick W. Taylor, considerado o pai da Teoria Clássica da Administração, desenvolveu quatro princípios básicos: ([SILVA, 2001](#))

1. Desenvolvimento de um método científico para o trabalho do operário. Esse princípio reforça o planejamento inicial necessário em um processo prescritivo. Esse planejamento envolveu a busca do melhor processo de executar uma atividade no menor tempo possível.
2. Estabelecimento de processo científico de seleção e treinamento do operário. O princípio insere a especialização e treinamento de recursos humanos.
3. Cooperação entre as gerências e os operários. Com o intuito de garantir que o trabalho fosse realizado de acordo com o que foi planejado e da maneira certa, esse princípio implanta o conceito de controle de atividades.
4. Divisão do trabalho dos operários em função da sua especialização. Esse princípio refere-se à padronização de atividades e tarefas que um processo prescritivo possui. Cada atividade passa a ter um responsável especialista por desenvolvê-la.

Esses princípios transferem a responsabilidade de decisão sobre as tarefas, que antes eram exercidas pelos operários, para as mãos de um gerente. Pressupondo que existe

uma melhor e padronizada forma de trabalho, cabe então ao gerente encontrá-lo. Essa forma de organizar o trabalho estimula a produção em série, de massa, a baixos custos, e a extrema especialização (automatização do ser humano). (SILVA, 2001) Esse arcabouço teórico é base de sustentação de diferentes métodos de desenvolvimento de produtos e gestão da produção, como por exemplo o guia PMBoK, no contexto do gerenciamento de projetos de software.

No desenvolvimento de software é o mesmo que dizer que há uma processo padrão de desenvolvimento, onde há separação de funções e papéis, e que as atividades são determinísticas, previamente definidas, cabendo ao gerente identificar e sequenciar logicamente essas atividades, de forma a seguir um plano preditivo, definido nos meses iniciais de um projeto. Um Software possui diversas características que o transformam em um produto com propriedades intrínsecas, com vários estados possíveis, fazendo do software um produto altamente complexo. (5CQUALIBR, 2010)

2.2.1 O guia PMBoK

O Guia PMBOK® fornece diretrizes para o gerenciamento de projetos individuais, define o gerenciamento e os conceitos relacionados, e descreve o ciclo de vida do gerenciamento de projetos e os processos relacionados. Tem por objetivo ser um guia com um conjunto de conhecimentos e boas práticas de aplicação. Não é uma metodologia, é uma referência básica, logo, a norma não é abrangente nem completa, possibilita o uso de ferramentas e metodologias distintas para implementar sua estrutura, e fornece um vocabulário comum para se discutir, escrever e aplicar o gerenciamento de projetos entre os membros envolvidos.

O guia é baseado em várias áreas e processos que organizam o trabalho a ser realizado durante o projeto. Os processos se relacionam e interagem segundo uma lógica definida para a condução do trabalho, realizada através de entradas, ferramentas e técnicas, e saídas. Existe o pressuposto de que o detalhamento do trabalho segue de forma logicamente encadeada, e por meio desta visão, existe a premissa, cujo, sucesso do projeto é alcançado pela cumprimento dos passos descritos no plano.

2.2.1.1 Ciclo de vida e a organização do projeto

Consiste nas fases que oferecem uma estrutura básica para o gerenciamento do projeto, independente do trabalho específico envolvido, todos os projetos podem ser mapeados para a estrutura do ciclo de vida: 1) Início do projeto; 2) Organização e preparação; 3) Execução do trabalho do projeto; e 4) Encerramento do projeto. Esta estrutura pode ser visualizada na Figura 1. (PMI, 2008)

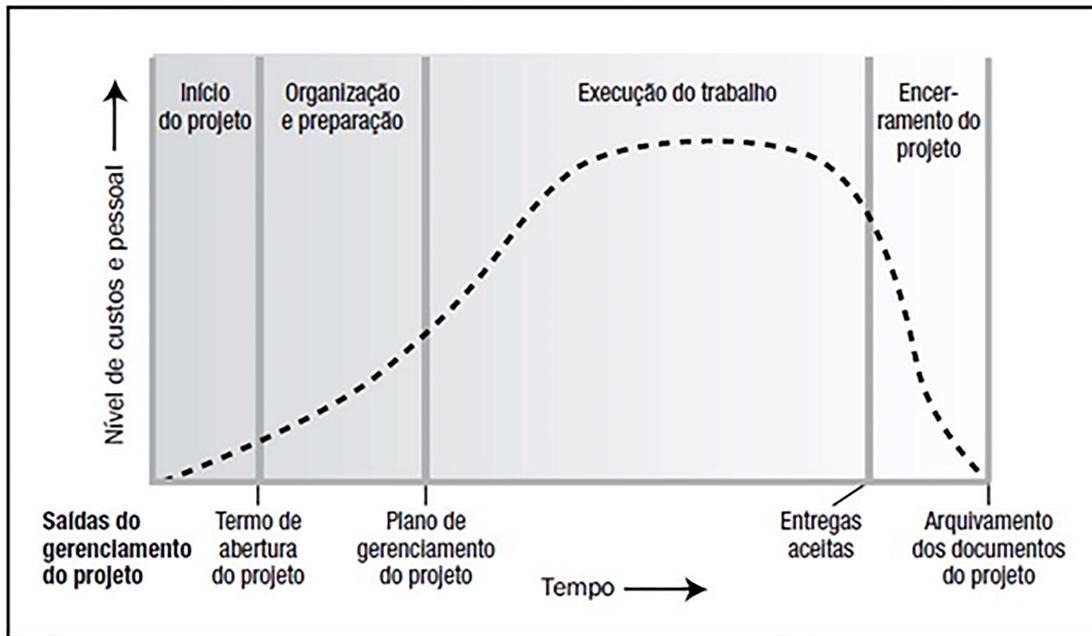


Figura 1 – Nível típico de custos e pessoal ao longo do seu ciclo de vida. Fonte (PMI, 2008)

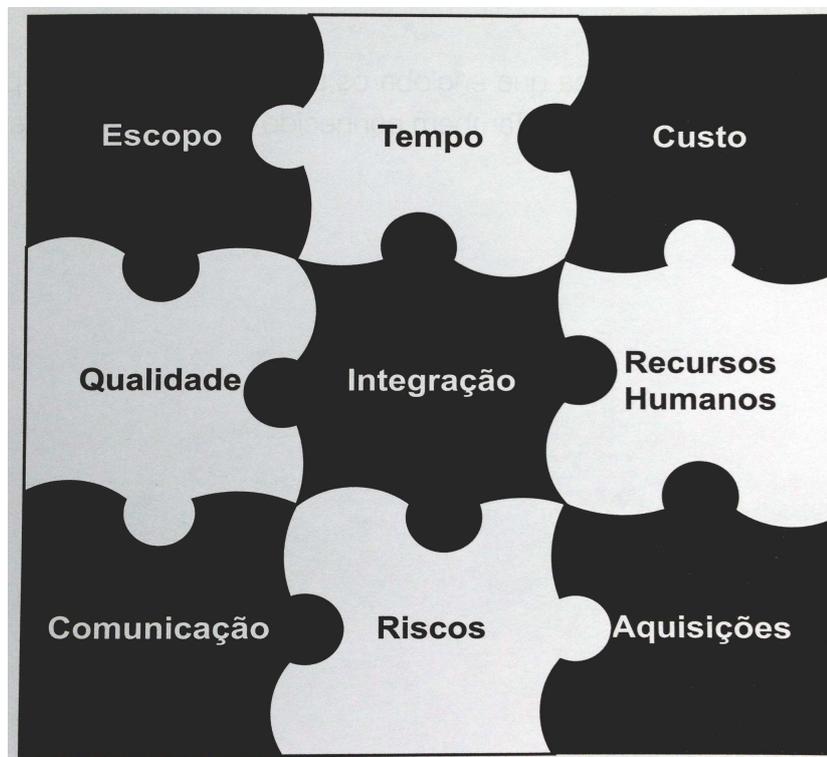


Figura 2 – Mapeamento de grupos de processos de gerenciamento de projetos e áreas de conhecimento. Fonte (VARGAS, 2009)

2.2.1.2 Grupos de Processos e Áreas de Conhecimento

Um processo é um conjunto de ações e atividades inter-relacionadas, que são executadas para alcançar um produto, resultado ou serviço predefinido. Os processos de gerenciamento de projetos são agrupados em cinco categorias: Iniciação; Planejamento; Execução; Monitoramento e Controle e encerramento. São representados por nove áreas, conforme pode ser visualizado na Figura 2

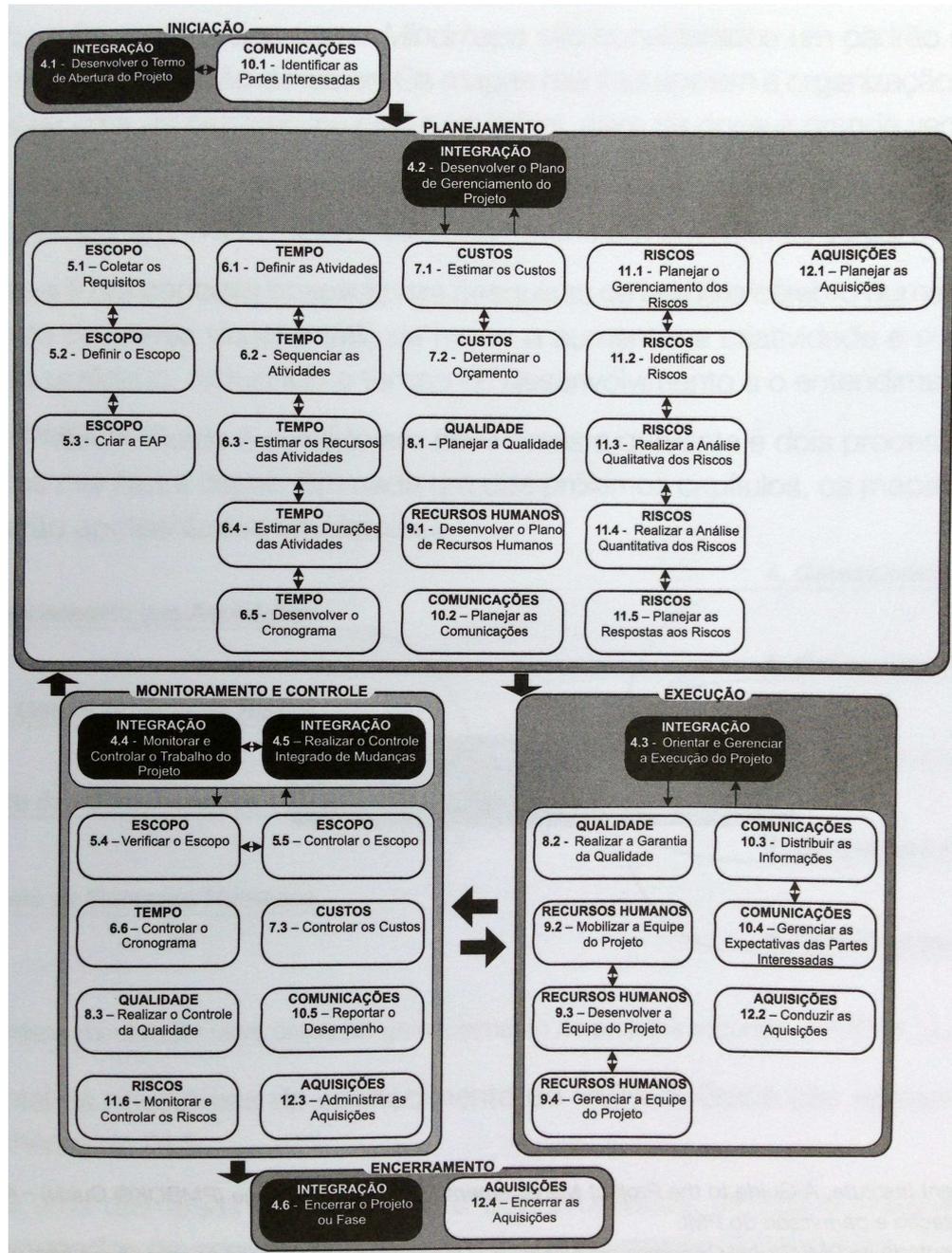


Figura 3 – Grupo de Processos de Gerenciamento de Projetos Fonte (VARGAS, 2009)

Esses processos são apresentados como elementos distintos com interfaces bem definidas. Porém, na prática eles se sobrepõem e sua aplicação é iterativa e muitos deles

são repetidos durante o projeto. A natureza integrativa do gerenciamento de projetos requer que o grupo de processos de monitoramento e controle interaja com os outros grupos de processos, o de iniciação começa o projeto e o de encerramento termina, e o de planejamento fornece ao grupo de processos de execução o plano de gerenciamento e os documentos do projeto à medida que o projeto avança.

As áreas de conhecimento na prática são interativas e podem se sobrepor e interagir. Descrevem quais são as tarefas, atividades e produtos de trabalho que devem ser produzidos ao longo dos grupos de processos. E a Figura 3 apresentada a relação entre os grupos de processos e áreas de conhecimento do PMBoK.

2.3 Gerenciamento Ágil

Muitas empresas de desenvolvimento de software estão se esforçando para se tornar mais ágeis. Equipes ágeis bem sucedidas estão produzindo software de alta qualidade para melhor atender, mais rapidamente, as necessidades dos usuários e com um custo menor e maior qualidade do que as equipes tradicionais. (COHN, 2012)

As empresas que tem tido sucesso na adoção de métodos ágeis, adotando um método como o SCRUM, combinado com outros métodos ágeis, estão evidenciando ganhos significativos de produtividade com diminuição de custos correspondentes. Elas são capazes de levar produtos ao mercado mais rapidamente e com um maior grau de satisfação do cliente, provendo maior visibilidade para o processo de desenvolvimento, levando a uma maior previsibilidade, como consequência, e não como causa. E para estas empresas os termos “fora de controle”, “projetos que jamais serão concluídos” tornaram-se coisa do passado. (COHN, 2012)

Um método empírico requer frequentemente transparência, inspeções e adaptações durante o projeto, que é definido de forma inexata e pode gerar resultados imprevisíveis. É mais adequado e indicado para projetos de inovação e criação de novos produtos, como o desenvolvimento de software, por exemplo. A metodologia empírica, um dos pilares do SCRUM, é indicada nas situações onde as entradas do processo variam e o processo é muito complexo para produzir resultados semelhantes. (MARTINS, 2007)

O método empírico foi introduzido pela escola das relações humanas visava se contrapor à escola clássica. Seus métodos são fundamentalmente a pesquisa experimental e tinham por objetivo demonstrar, mediante investigação empírica, que o resultado de determinada atividade não é obtido unicamente através de estudo científico de suas operações, além de observar que a interação entre os diferentes indivíduos, participantes do processo, funcionava como mecanismo informal na gerência de processos. (SILVA, 2001)

A concepção de sistema aberto reconhece que o sistema mantém um relaciona-

mento dinâmico com seu ambiente, ou seja, o sistema recebe várias entradas, atua sobre elas, e exporta determinadas saídas, e o modelo dinâmico da visão sistêmica considera ainda que um sistema é composto de subsistemas inter-relacionados afirmando que o todo não é simplesmente como a soma das partes individuais. Recentemente essa teoria também tem sido utilizada para explicar o comportamento da produção de um software. (SILVA, 2001)

Ao encontro da teoria de sistemas abertos, aproxima-se a teoria das restrições a qual busca aumentar a capacidade produtiva da organização através de melhorias no fluxo do processo, ao enxergar a organização como um sistema em que todas as partes são inter-relacionadas.(KATAYAMA, 2011)

Já os sistemas complexos adaptativos são redes com grande número de agentes que interagem entre si e com muitas características em comum. São complexos, pois, estão além da capacidade descritiva e são adaptativos, pois, são capazes de se adaptar a novas condições impostas a eles. Organizações são sistemas adaptativos complexos, e criativos e inovadores quando dão lugar para a inovação na proximidade do caos.(LORENZ, 1995) Trata-se de um estado paradoxal, pois é estável e instável ao mesmo tempo, orientado pela contraditória dinâmica da competição e cooperação, da amplificação e restrição, da exposição a tensão criativa e proteção dela. (IAMAMOTO; GAILLAND, apud STACEY, 1996) Dessa maneira, há diversos eventos aleatórios acontecendo, o que torna qualquer tentativa de previsão mera adivinhação. Nesses sistemas o todo é maior do que a soma das partes, logo são totalmente imunes a métodos científicos. São também livres de qualquer coordenação global, absoluta, efetiva e duradoura.(IAMAMOTO; GAILLAND, apud STACEY, 1996)

SATO, DERGINT e HATAKEYAMA (2005) elucidam que mudanças estão acontecendo de forma mais freqüente e profunda, e que é cada vez mais difícil prever e controlar tais mudanças. Isto exige uma nova postura: aceita-las e não controla-las. Baseado nisso, a teoria da complexidade e os sistemas adaptativos complexos oferecem uma forma para as organizações conviverem com elas. Nesse contexto de evolução das empresas em direção a sistemas evolutivos complexos, o papel do gerente de projetos como uma autoridade central controladora desaparece, dando lugar a agentes de mudança ou agentes reguladores, trabalhando no sentido de influenciar as pessoas e viabilizar a auto-organização dos projetos da empresa.

Todas essas teorias juntas: 1) A Escola das Relações Humanas (foco no indivíduo); 2) Sistemas Abertos (relacionamento dinâmico entre as partes); 3) Restrições (foco em mudanças e progresso contínuo); e, 4) Sistemas Complexos e Adaptativos, reforçam, justificam e validam, o processo empírico e a metodologia SCRUM.

2.3.1 SCRUM

O Scrum é um método de gerenciamento que segue os princípios e valores do manifesto ágil (BECK et al., 2001) e possui práticas que auxiliam equipes a entregar produtos em ciclos curtos de produção, favorecendo um feedback rápido, melhoria contínua e rápida adaptação a mudanças. Seu uso tem sido predominantemente na indústria de software, mas também está provando ser eficaz em áreas diferentes e tem funcionando muito bem em qualquer escopo complexo e inovador. (SCRUMALLIANCE, 2013)

O Scrum se baseia na idéia de controle empírico de processo, isto é, utiliza o conceito de progresso por meio da capacidade produtiva do time e do controle das restrições no processo, e não procura a predição, como forma de medir variações e desempenho. Ao invés de fornecer completas e detalhadas descrições de como tudo deverá ser feito no projeto, permite a equipe ¹ decidir por si só como deverá ser feito o trabalho, possibilitando que a equipe descubra qual a melhor forma de como resolver o problema apresentado. (SCRUMMETHODOLOGY, 2013)

Equipes ágeis possuem características de auto-organização e multifuncionalidade: auto-organização, ou seja, a equipe é auto organizável, onde não há um líder geral que decide qual tarefa cada pessoa irá fazer, pois essas questões são decididas em equipe; multifuncionalidade, ou seja, cada membro da equipe precisa ter habilidade de se responsabilizar pela produção de uma dada funcionalidade desde a idéia original até sua implantação. (MOUNTAINGOAT, 2013) O modelo de equipe no Scrum é projetado para aperfeiçoar a flexibilidade, criatividade e a produtividade das pessoas. (SCHWABER; SYTHERLAND, 2011)

A estrutura do framework ainda prevê papéis, eventos, artefatos e algumas restrições, que serão detalhados nas próximas subseções. Vale ressaltar que esta estrutura segue o modelo de ciclo de vida iterativo e incremental, como é mostrado na Figura 4. (SCHWABER; SYTHERLAND, 2011)



Figura 4 – Esqueleto SCRUM. Fonte (SCHWABER, 2004)

¹ Adotaremos neste trabalho equipe e time como sinônimos

2.3.1.1 Papéis SCRUM

Product Owner (PO)

O Product Owner é responsável por representar os interesses de todo mundo que possua uma participação, ou interesse, tanto no projeto como no sistema resultante, ou seja, representa o negócio, clientes ou usuários, e orienta a equipe a construir o produto certo. (MOUNTAINGOAT, 2013) Ele fornece a base inicial e contínua do projeto apresentando seus requisitos gerais e planos de entrega. O PO é o único responsável por gerenciar o Backlog do produto e deve garantir que a funcionalidade mais importante seja produzida primeiro. Isso é alcançado pela frequentemente priorização do Backlog do produto. (SCHWABER, 2004)

Time de Desenvolvimento

O time consiste de membros que realizam o trabalho, são auto gerenciáveis, auto organizáveis, indivisíveis em sub-equipes, multifuncionais e possuem a responsabilidade de tornar o Backlog do produto em um incremento funcional de software em uma Sprint, além de gerenciar seus próprios trabalhos a serem feitos. O membros do time são responsáveis pelo sucesso de cada sprint e do projeto como um todo. (SCHWABER, 2004)

ScrumMaster (SM)

O ScrumMaster é responsável por garantir o uso dos princípios e práticas do Scrum. É muitas vezes considerado um treinador para a equipe, ajudando-a a fazer o melhor trabalho que puder. Isso envolve a remoção de quaisquer impedimentos ao progresso, facilitando reuniões e interagindo diretamente com o Product Owner para garantir que o Backlog do produto esteja em “boa forma” e pronto para a próxima Sprint. (SCHWABER, 2004)

2.3.1.2 O Fluxo de Trabalho

A partir do documento de visão, o PO elabora, em alto nível, um plano de execução que inclua o Backlog do Produto (BP), que é uma lista de requisitos funcionais e não funcionais e prioriza o BP e o itens que estão no topo, que mais agregam valor, de modo que sejam primeiramente implementados pelo time. O BP é então dividido em releases, a partir das quais as Sprints são planejadas. (SCHWABER, 2004)

Cada Sprint inicia com uma reunião de planejamento da Sprint, onde o PO e o time selecionam o que será feito na Sprint. O time esclarece as dúvidas sobre os requisitos diretamente com o PO, de forma que a meta da Sprint seja definida.

Os itens do BP selecionados para a sprint formam o Backlog da Sprint. Todos os dias o time se reúne por 15 minutos em uma reunião diária SCRUM, onde cada membro responde três perguntas:

- O que eu fiz desde a reunião diária anterior? (KEITH, 2010)
- O que eu farei até a próxima reunião diária? (KEITH, 2010)
- O que me impede de desempenhar meu trabalho eficientemente? (SCHWABER, 2004)

No fim da Sprint, realiza-se a reunião de revisão da Sprint, onde o time apresenta para o PO o que foi desenvolvido durante a Sprint. O PO e os stakeholders inspecionam o incremento e fazem adaptações no projeto para otimizar as chances de alcançar os objetivos. Antes de iniciar uma nova Sprint, o ScrumMaster realiza uma reunião de retrospectiva, em que todos os membros do time participam e, baseado no Scrum, revisam todo o processo de desenvolvimento para torna-lo mais efetivo na próxima Sprint. (SCHWABER, 2004)

3 MÉTRICAS: UMA VISÃO SOBRE CUSTO, ESCOPO E TEMPO

3.1 CONTEXTUALIZAÇÃO

De acordo com a norma [ISO/IEC \(2007\)](#) medição é um conjunto de operações que tem o objetivo de determinar o valor de uma medida¹. Já [PRESSMAN \(2011 apud FENTON; PFLEEGER, 1997\)](#), dissertam que medição é o processo pelo qual números ou símbolos são anexados aos atributos de entidades no mundo real para defini-los de acordo com regras claramente estabelecidas.

De acordo com a norma [IEEE \(1998\)](#), uma métrica de software 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.

Ainda o Institute of Electrical and Electronics Engineers (IEEE) afirma que as métricas de software são classificadas como diretas ou derivadas. Onde as métricas diretas não dependem de nenhum atributo e não precisam 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. ([IEEE, 1998](#))

A norma [ISO/IEC \(2007\)](#) ainda define quatro atividades de um processo de medição: 1) Estabelecer e sustentar o compromisso de medição; 2) Planejar o processo de medição; 3) Realizar o processo de medição; e 4) Avaliar a medição. Dentro do processo de medição existem várias tarefas que devem ser alcançadas para garantir a execução correta do processo e para sustentar um conjunto de princípios básicos que orientem a definição de métricas de software.

Já a norma [ISO/IEC \(2011\)](#) define o modelo de qualidade de software, onde as métricas de qualidade interna, externa e as de qualidade em uso se relacionam e seguem um fluxo de dependência e de influência, levando em consideração entradas de métricas de produto, o efeito do produto de software e contextos de uso. A [Figura 5](#) ilustra o modelo de qualidade.

¹ A definição formal da [ISO/IEC \(2007\)](#) não utiliza o termo métrica, sendo que este é utilizado em outras referências como [Basili, Caldiera e Rombach \(1996\)](#) [IEEE \(1998\)](#). Contudo, no contexto do deste trabalho, compreende-se o termo medida como uma dimensão da produção observada externamente ao produto software e métrica o valor quantitativo que alimenta a medida

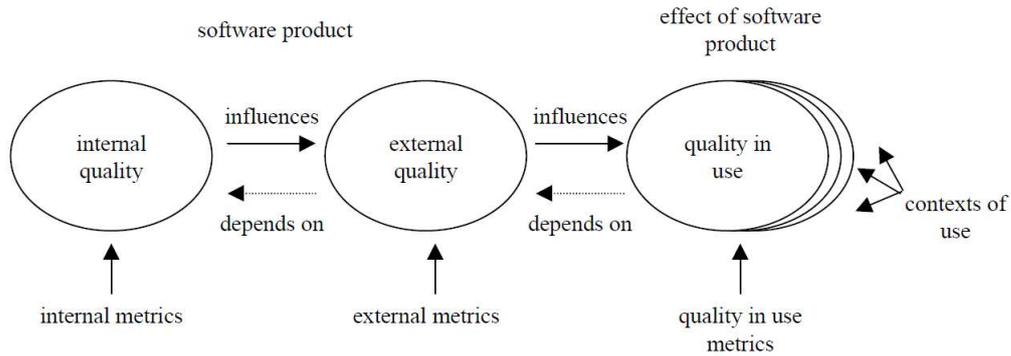


Figura 5 – Relacionamento entre as métricas de qualidade. Fonte (ISO/IEC, 2011)

As métricas internas de qualidade fornecem aos usuários a capacidade de medir a qualidade dos resultados intermediários e, assim, prever a qualidade do produto final. Isso permite que o usuário identifique precocemente problemas de qualidade e possa iniciar ações corretivas o mais cedo possível. (ISO/IEC, 2011)

As métricas de qualidade exterior podem ser usadas para medir a qualidade do software medindo o comportamento do sistema do qual faz parte. As métricas de qualidade em uso medem se um produto atende às necessidades de usuários específicos para atingir metas especificadas com eficácia, produtividade, segurança e satisfação em um contexto de uso especificado. (ISO/IEC, 2011) SILVEIRA, BECKER e RUIZ (2012) afirmam que a qualidade do software está diretamente relacionada com a qualidade do processo através do qual o software é desenvolvido.

Observando as categorias apresentadas por SPIES e RUIZ (2012) podemos observar certa diferença no foco da medição :

- Processo: têm por objetivo obter controle sobre o processo de desenvolvimento e monitorar como o trabalho vem sendo feito como. Exemplos: Abordagem tradicional: métrica de aderência ao processo. (CMMI,) Abordagem Ágil: capacidade produtiva do time e velocidade de produção (POPPENDIECK; POPPENDIECK, 2011)
- Produto: têm relação com as características inerentes do produto desenvolvido. Exemplos: Abordagem tradicional: métrica de densidade de defeitos. (CMMI,) Abordagem ágil: métrica de percentual de cobertura de código. (BECK; GAMMA, 1998)
- Projeto: visam monitorar o projeto considerando atributos como o tamanho do time, produtividade, cronograma e esforço, de modo a manter o projeto controlado e dentro das expectativas. Exemplos: Abordagem tradicional: EVM (VARGAS, 2002) (SULAIMAN; BARTON; BLACKBURN, 2006)

Diante do arcabouço teórico apresentado neste e no capítulo 2, podemos observar que existe uma diferença no foco da medição entre as metodologias listadas no capítulo 2:

- Foco da medição tradicional: utiliza método científico tradicional, onde os métodos estatísticos e matemáticos procuram medir, por exemplo: esforço de cada indivíduo; tempo de execução de tarefas e atividades; estabelecimento de padrões; busca de previsibilidade, predição; custo, prazo e escopo através de técnicas como pontos de função, EVM tradicional. O foco está individualmente nos processos padronizados e imutáveis dentro de um processo de produção.(PMI, 2008)
- Foco da medição ágil: utiliza método científico empírico, onde os métodos estatísticos e matemáticos procuram medir, por exemplo: o resultado; o esforço coletivo; o fluxo de trabalho; a qualidade do produto; a capacidade de auto-organização; entre outros, pois, como citado no capítulo 2, software tem características de um sistema complexo e adaptativo.(IAMAMOTO; GAILLAND,)

A seguir é apresentada duas variações de aplicação de uma técnica para coleta e interpretação de métricas de gerenciamento de projetos, as quais no contexto desse trabalho engloba as medidas de custo, escopo e tempo.

3.2 EVM TRADICIONAL

O (PMI, 2008) define o EVM como um método comumente usado para medir desempenho e tem a característica de integrar as medidas de escopo, custos e tempo, calculando o valor agregado pelo percentual de esforço realizado (horas) multiplicado pelo orçamento planejado (R\$), até a data da avaliação. Com isso auxilia a equipe de gerenciamento a avaliar e medir o desempenho e progresso do projeto. (HELDMAN, 2009)

Já (VARGAS, 2011) define o EVM como a avaliação entre o que foi obtido em relação ao que foi realmente gasto e ao que se planejava gastar, onde se propõe que o valor a ser agregado inicialmente por uma atividade é o custo orçado para ela. Na medida em que cada atividade ou tarefa de um projeto é realizada, aquele valor inicialmente orçado para a atividade passa, agora, a constituir o Valor Agregado do projeto.

Segundo FLEMING e KOPPELMAN (1998) muitas empresas privadas estão tendo dificuldades para empregar esses rígidos critérios em seus projetos. Suas percepções são de que existem muitas exigências que não agregam valor ao projeto.

O EVM monitora três elementos básicos: O VP (PV, em inglês), CA (AC, em inglês) e o VA (EV, em inglês) podem ser apresentados em um gráfico capaz de mostrar as variações existentes entre eles. Se não houver variações, todas as linhas do gráfico ficarão

sobrepostas, ou seja, serão iguais, o que indica que o projeto está evoluindo conforme planejado. Na Figura 6.

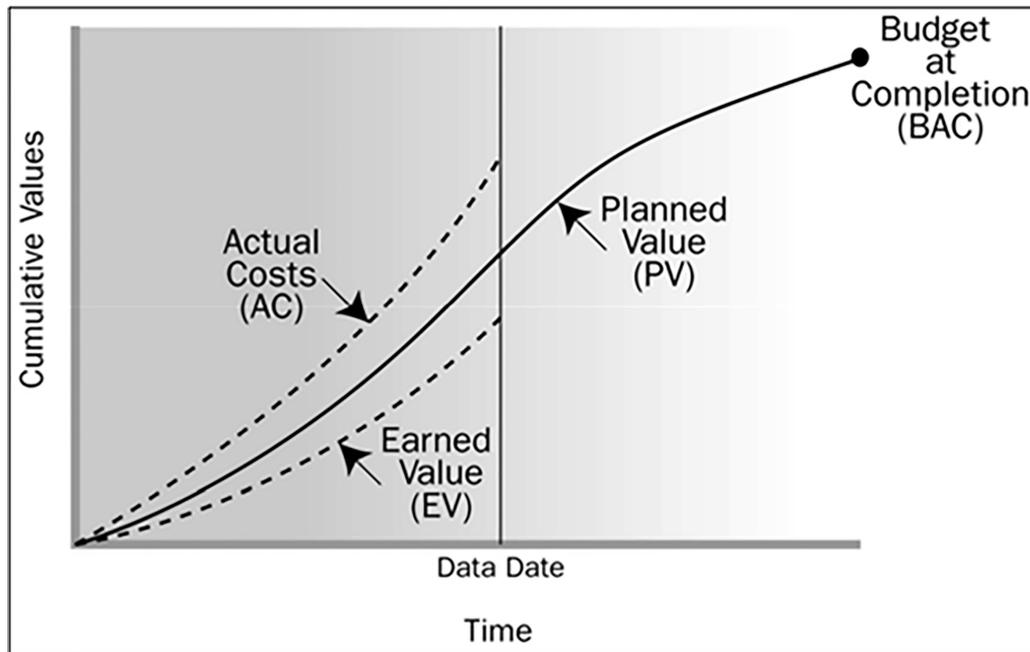


Figura 6 – Gerenciamento do Valor Agregado. Fonte (PMI, 2008)

A variação dos valores planejado e agregado e do custo atual resultam em medidas de desempenho, a partir da linha de base aprovada, que também serão monitoradas e demonstra interpretações ao gerenciamento do valor agregado. Na tabela 1 são descritos VP, CA e VA:

A Tabela 2 apresenta o elementos derivados do EVM Tradicional

Os valores da VP e VC podem ser convertidos em indicadores de desempenho para refletir o desempenho dos custos e dos prazos de qualquer projeto. (PMI, 2008) As variações e os índices são úteis para determinar o andamento do projeto e também para fornecer uma base para a estimativa de custos e prazos.

FLEMING e KOPPELMAN (1998) destacam que o conceito de valor agregado surgiu no Departamento de Defesa Americano, o qual desenvolveu uma diretiva que impôs 35 critérios de controle de sistemas baseado em custo e cronograma. Além disso, afirmam que embora algumas pessoas considerem esses 35 critérios uma idéia utópica a ser implementada por todas as empresas, muitas pessoas dentro das da industria privada tem enfrentado dificuldade em empregar esses critérios rigidios em todos seus projetos – principalmente em projetos comerciais.

(FLEMING; KOPPELMAN, 1998) ainda citam que a percepção das pessoas que tentaram implementar o EVM é de que há muitos critérios que não adicionam valor para que a técnica seja empregada universalmente em todos os projetos. Quando bem

Tabela 1 – Elementos Básicos EVM Tradicional

Métrica	Definição
Valor Planejado	O valor planejado é o custo autorizado e orçado designado para o trabalho a ser executado em uma atividade, atribuição, recurso ou em um componente da estrutura analítica do projeto. (PMI, 2008) É calculado como os custos de linha de base divididos em fases e acumulador até a data de referência, ou a data atual. (VARGAS, 2011)
Custo Atual	O custo atual é o custo da conclusão do componente de trabalho em um dado período de tempo. (HELDMAN, 2009) O CA tem que corresponder em definição ao que foi orçado para o VP e medido no VA. (PMI, 2008)
Valor agregado	O valor agregado é o valor do trabalho concluído até a presente data quando comparado ao custo estimado (VP) atribuído ao componente de trabalho. É a soma dos custos orçados cumulativos do trabalho concluído para todas as atividades que foram executadas desde a data de medição. (HELDMAN, 2009) O VA sendo medido deve ser relacionado à linha de base do VP, e o VA medido não pode ser maior que o orçamento VP de um componente. O termo VA é frequentemente usado para descrever a porcentagem completa de um projeto. (PMI, 2008)

aplicada, pode dar ao gerente de projeto um sinal alerta prévio de que o projeto está se encaminhando para uma extrapolação do custo a menos que ações imediatas sejam tomadas. E afirmam que o mundo do software precisa de algo menos formal do que esses 35 critérios, algo que possa ser adaptado para as características de cada projeto.

VARGAS (2009) disserta que, no que diz respeito ao valor da técnica de valor agregado, os resultados encontrados são enquadrados em uma faixa de pouco valor, ficando abaixo de praticamente todas as técnicas analisadas, o que sugere que a popularidade da técnica de valor agregado não retrata sua aplicabilidade ou valor. E conclui que na indústria encontra-se notada dificuldade tanto na coleta dos dados quanto na baixa velocidade da geração da informação.

Para justificar essa afirmação FLEMING e KOPPELMAN (1998 apud VARGAS, 2009) propõem, também, que outro fato de dificuldade encontrada consiste no detalhamento adequado da EAP que pode ser extensa, gerando um custo de controle alto, ou pequena, que pode representar uma diminuição na precisão dos dados de custo atual e prazo.

Para comprovar PETERSON e OLIVER (2001 apud VARGAS, 2009), afirma que há uma baixa aplicação do EVM na área de tecnologia e marketing, onde aspectos relacionados ao trabalho criativo atuam como variantes do escopo previamente definido, tornando sua aplicabilidade limitada e diretamente relacionada à estabilidade do escopo

Tabela 2 – Elementos Derivados do EVM Tradicional

Métrica	Fórmula	Definição
Varição de custos (VC)	$VC = EV - CA$	A variação de custos mede o desempenho real até a presente data ou durante o período contra aquilo que foi gasto. Se a VC for positiva, indica que os custos permanecem inferiores às projeções, ou seja, gastou-se menos do que o planejado; e se a VC for negativa, indica que os custos superar o custo planejado. (HELDMAN, 2009)
Varição de prazos (VP)	$VP = EV - VP$	A variação de prazos informa se o cronograma está adiantado ou atrasado em relação ao planejado para um período de tempo. A variação de prazos do EVM finalmente se igualará a zero quando o projeto terminar, pois todos os valores planejados terão sido agregados. (PMI, 2008)
Índice de desempenho de custos (IDC ou CPI)	$CPI = EV / CA$	O índice de desempenho de custos é uma medida do valor do trabalho executado comparado ao custo atual ou progresso feito no projeto. É considerado a métrica mais crítica do EVM e mede a eficiência de custos do trabalho executado. (PMI, 2008) <ul style="list-style-type: none"> • CPI menor que 1.0 indica um excesso de gasto financeiro para o trabalho executado. (PMI, 2008) • CPI maior que 1.0 indica um desempenho de gasto financeiro abaixo do limite. (PMI, 2008)
Índice de desempenho de prazos (IDP ou SPI)	$SPI = EV / VP$	O índice de desempenho de prazos é uma medida do progresso alcançado comparado ao progresso planejado num projeto. Às vezes é usado em conjunto com o índice de desempenho de custos (IDC) para prever as estimativas finais do término do projeto. (PMI, 2008) <ul style="list-style-type: none"> • SPI menor que 1.0 indica que menos trabalho foi executado do que o planejado. (PMI, 2008) • SPI maior que 1.0 indica que mais trabalho foi executado do que o planejado. (PMI, 2008)

definido.

PETERSON e OLIVER (2001 apud VARGAS, 2009) afirmam que com o crescimento de projetos de curto prazo, com equipes reduzidas e escopo genericamente definido, onde a definição do trabalho restante é definida à medida que os trabalhos atuais ocorrem, a análise de Valor Agregado, conforme proposto pela Instrução 5000.2R (DOD, 1997) e pela ANSI/EIA 748, torna-se inviável, devido às projeções imprecisas decorrentes do escopo mal definido, bem como aos custos percebidos pelos empreendedores como elevados.

Para dar relevância para esse Trabalho de Conclusão de Curso é preciso fundamentar os conceitos de Gerenciamento de Valor Agregado Ágil, comumente utilizado como AgileEVM.

3.3 AgileEVM

Baseado nas metodologias ágeis que enfatizam um planejamento multi-nível e incremental, e desencorajam um planejamento completo e detalhado, [SULAIMAN, BARTON e BLACKBURN \(2006\)](#) afirmam que a técnica EVM Tradicional, que assume um completo planejamento dos Pacotes de Trabalho de um projeto e depois atribuem custo e duração a estes pacotes, tem a sua utilização em projetos ágeis sido questionada.

Para resolver esse questionamento, [SULAIMAN, BARTON e BLACKBURN \(2006\)](#) definem o Agile Earned Value Management – AgileEVM (em português Gerenciamento Ágil do Valor Agregado): um conjunto simplificado de cálculos de valor agregado adaptado do EVM Tradicional usando métricas de times ágeis, que evidenciam a validade da adaptação do EVM Tradicional para ser usado em projetos SCRUM.

A implementação do AgileEVM concentra-se em medir progresso do escopo, no nível de release, pois é ao final de cada sprint que tem-se a velocidade do time naquela sprint e seus decorrentes (CA). No tabela 3 é possível observar três itens de comparação

Tabela 3 – Comparação dos termos do EVM

Métrica	EVM	AgileEVM
Orçamento ao final	Custo total orçado para a release ou projeto.	O custo orçado para a release.
Porcentagem planejada completada	Quantidade de tarefas planejadas até esse ponto dividido pelo desempenho da linha de base.	O número da Sprint atual dividido pelo total de Sprints planejadas.
Porcentagem atual completada	O custo atual dos pacotes de trabalho já completados dividido pelo custo total do orçamento ao final.	O número total de pontos de estórias completadas dividido pelo número total de pontos de estória planejadas.

entre o EVM Tradicional e o AgileEVM.

O importante a se observar ao se comparar o EMV Tradicional com o AgileEVM, é que esta mede a variação do escopo, enquanto que aquela mede a variação do esforço, em função da variação do custo. Isso pode ser explicado pela forma como as duas abordagens tratam a produção do software.

Na Tabela 4 é apresentado um conjunto de métricas que auxiliam a criar linha de base inicial para a medição de progresso do projeto, no contexto do AgileEVM.

Tabela 4 – Parâmetros iniciais da release

Nome	Definição
PC	Pontos completados – os pontos de trabalho completados do Backlog da release durante a Sprint.
PA	Pontos adicionados – os pontos adicionados ou subtraídos do Backlog da release durante a Sprint.
SC	Custo da Sprint – o que foi gasto.

A Tabela 5 exibe os parâmetros coletados ao final de cada uma das sprints.

Tabela 5 – Métricas para calculo do AgileEVM

Nome	Definição
PRP	Pontos planejados para a release.
RPC	Pontos completados da release.
APC	Porcentagem atual completada da release. Pontos Completados / Pontos Planejados.
PPC	Porcentagem completada planejada.

Para calcular o valor planejado e o valor agregado, [SULAIMAN, BARTON e BLACKBURN \(2006\)](#) afirmam ser necessário ter uma medição precisa da porcentagem atual completada e a porcentagem planejada completada.

A Tabela 6 apresenta os parâmetros complementares para o cálculo do AgileEVM.

Tabela 6 – Parâmetros Complementares

Nome	Definição
BAC	O custo orçado a ser gasto na release.
PS	Número total de sprints planejadas para a release.
PRP	Número total de pontos planejados para a release.

E a Tabela 7 apresenta as definições e fórmulas utilizadas para o cálculo do AgileEVM.

Desta forma o foco da solução deste trabalho é construir um arcabouço que auxilie o time a tomar decisões a respeito do projeto e do produto de forma mais consciente e tempestiva, uma vez que essas decisões serão apoiadas pelo uso de um ambiente de DWing, apresentado no capítulo seguinte.

Tabela 7 – Definições e Equações AgileEVM

Nome	Definição
BAC	Custo orçado a ser gasto na release. É o gasto inicial para a release.
AC	Custo atual: esse corresponde ao custo orçado para liberação.
$PV = \frac{PPC}{BAC} *$	Valor planejado
$EV = \frac{APC}{BAC} *$	Valor agregado
$CV = EV - AC$	Varição do custo
$SV = EV - PV$	Varição do prazo
$CPI = EV/AC$	Índice de desempenho de custos
$SPI = EV/PV$	Índice de desempenho de prazo
$ETC = \frac{1}{CPI} * (BAC - EV)$	Estimativa para o término.
$EAC = AC + ETC$	Estimativa no término.

4 DATA WAREHOUSE-DW COMO APOIO À ATIVIDADES DE MEDIÇÃO DE PROJETOS DE DESENVOLVIMENTO SOFTWARE

Quanto mais as informações de negócio são compartilhadas entre os projetos da organização, mais abrangente e eficaz é a capacidade da organização para avaliar os seus projetos. [SILVEIRA, BECKER e RUIZ \(2012\)](#) [KIMBALL e ROSS \(2002\)](#) afirmam que a solução para o maior grau de integração e de normalização destas informações é representá-las com o uso de um Data Warehousing, onde as métricas são estruturadas de acordo com um modelo multidimensional. Um DW é uma coleção de dados históricos integrados, não-voláteis, orientado a assunto, destinado a apoiar os processos de tomada de decisão durante um período específico de tempo. ([KIMBALL; ROSS, 2002](#))

- Integrado: Múltiplas fontes de dados que devem ser limpas, tratadas, convertidas, formatadas, e resumidas antes de serem armazenadas.
- Orientado a assunto: organiza os dados pelos principais elementos do negócio para que os dados sejam analisados filtrando dados relevantes e excluindo os não úteis para a tomada de decisão.
- Não-volátil: 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.

Já na perspectiva do usuário, vale ressaltar que([BOUMAN, 2009](#)):

- Todas as informações estão em um mesmo lugar - não há a necessidade de buscar informações em diversas fontes ou em estruturas de pastas confusas e infinitas;
- Informações atualizadas – as informações no DW são automaticamente carregadas e atualizadas em uma base regular;
- Acesso rápido – O DW é otimizado para a recuperação rápida de informação;
- Não há limites de tamanho - Um DW pode armazenar uma quantidade quase ilimitada de dados;

- Todo o histórico disponível – um DW mantém dados atuais e dados históricos de toda a informação. Isto significa que qualquer análise de tendências ou comparação ao longo do tempo é apoiado pelo ferramenta. O histórico disponível não é apenas um conjunto de “dados antigos”, e sim um conjunto que agrega valor quando mudanças são controladas;
- Fácil de entender - O DW é modelado em termos de negócios e reflete a maneira como o stakeholder olha para a sua organização;
- Definições claras e uniformes – todo mundo na organização utiliza as mesmas definições, o que simplifica muito a comunicação;
- Dados padronizados – todos os dados estão em conformidade com as normas, o que significa que existe apenas uma definição e apenas um conjunto de valores para cada pedaço de informação;

Um DW tem como objetivos: 1) fornecer um fácil acesso às informações da organização; 2) apresentar consistentemente as informações da organização; 3) ser adaptativo e resistência a mudanças; 4) ser um ambiente seguro para proteger as informações da organização; 5) servir de base para a melhoria na tomada de decisões; e, 6) ser aceito pela comunidade de negócio. (KIMBALL; ROSS, 2002)

4.1 Componentes de um ambiente de DWing

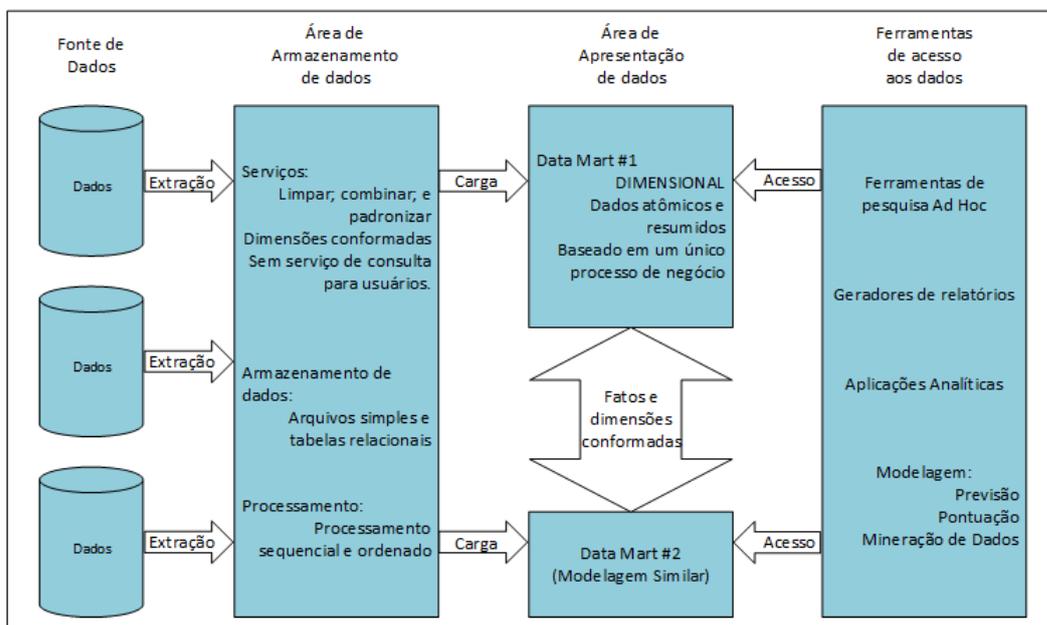


Figura 7 – Componentes de um DWing. Fonte (KIMBALL; ROSS, 2002)

DWing é o ambiente composto pelo DW mais as ferramentas para manipulação de seu back-end e front-end. As etapas de extração, transformação, carga e atualização do DW, formam o seu back-end e caracterizam o processo chamado Extraction-Transform-Load (ETL, sigla em inglês). As ferramentas de apoio às consultas On-Line Analytical Processing (OLAP, sigla em inglês) ao DW representam o seu front-end. O DW é um repositório de dados central e consolidado.

Nas próximas subseções são descritos os principais componentes e tecnologias de um ambiente de DWing.

4.1.1 Sistemas de Fonte de Dados Operacionais - (OSS, sigla em inglês)

Sistemas OSS são aqueles que capturam as transações do negócio, ou seja, estes representam as fontes de coleta dos dados, das informações do negócio. Portanto, são fontes que estão fora do escopo do DW, pois se tem pouco ou nenhum controle sobre o conteúdo e o formato dos dados. Por serem conhecidos como sistemas orientados a processo (OLTP, sigla em inglês), esses sistemas devem ter alto desempenho e disponibilidade. Os dados provenientes do OSS são parte da infra-estrutura da organização e são, geralmente, detalhados, atualizáveis e não-redundantes. (KIMBALL; ROSS, 2002) Cabe ressaltar que os dados podem estar dispersos em outras fontes, como: arquivos textos e planilhas eletrônicas, entre outras.

4.1.2 Área de armazenagem de dados (DSA, sigla em inglês)

Área de armazenagem de dados é tanto uma área de armazenamento como um conjunto de processos comumente identificados como o ETL. Essa área armazenagem é tudo que existe entre as difentes fontes de dados e o DW. Em um DW os dados operacionais brutos são transformados e ajustados para consulta do usuário e consumo dos mesmos. (KIMBALL; ROSS, 2002)

4.1.2.1 ETL

Extração é o primeiro passo no processo de obtenção de dados para o ambiente do DW. Uma vez que os dados são extraídos para a área de armazenamento, existem inúmeras potenciais transformações que podem ser realizadas, como por exemplo: limpeza dos dados (correção ortografia, resolver conflitos de domínio, lidar com elementos em falta, ou converter para formatos padrão); combinação de dados de várias fontes; duplicação de dados; atribuição de chaves do DW. Transformações são precursoras para o carregamento dos dados, passo final do processo de ETL. KIMBALL e ROSS (2002) afirma que ETL é um processo crítico e pode consumir até 85% de todo o esforço global na criação e manutenção de um ambiente DW.

4.1.3 Área de Apresentação dos dados

A área de apresentação de dados é onde os dados são organizados, armazenados e disponibilizados para consulta direta pelos usuários, geradores de relatórios e outras aplicações analíticas. Essa é a área onde os usuários têm acesso. De modo geral, referem-se a área de apresentação de dados como uma série data marts integrados. (KIMBALL; ROSS, 2002)

Um data mart é uma fatia da área total da apresentação, ou seja, minimalistamente falando, um data mart apresenta os dados de um único processo de negócio. Os data marts da área de apresentação devem contar dados atômicos e detalhados. Dados atômicos são necessários para evitar consultas imprevisíveis por conta dos usuários. (KIMBALL; ROSS, 2002)

4.1.4 Ferramentas de Acesso aos Dados - (DAT, sigla em inglês)

Ferramentas de acesso a dados nada mais são que as mais diversas formas de acesso aos dados da área de apresentação, e uma forma bem comum é através de ferramentas. Uma ferramenta de visualização de dados pode ser tão simples como uma ferramenta de consulta ad hoc ou tão complexa como uma mineração de dados sofisticado ou aplicação de modelagem. (KIMBALL; ROSS, 2002)

4.2 Modelagem Dimensional

Modelagem dimensional é uma técnica para modelar banco de dados e deixá-los simples e compreensíveis. Contém as mesmas informações que um modelo normalizado, porém organiza os dados em um formato cujo os objetivos são a compreensão do usuário, o desempenho de queries, e a resistência à mudança. (KIMBALL; ROSS, 2002)

4.2.1 Arquitetura

A arquitetura de um ambiente *DWing* pode variar conforme o contexto do assunto abordado, isso ocorre devido às necessidades de cada empresa. (KIMBALL; ROSS, 2002) Porém, um ambiente *DWing* consiste basicamente de três camadas, são elas:

- Camada de Acesso a Dados Operacionais.
- Camada de Extração e transformação de dados, e Carga destes dados no *DW*.
- Camada de Acesso à Informação. (KIMBALL; ROSS, 2002)

A Figura 8 descreve a arquitetura geral de um ambiente *DWing*, e é possível visualizar as três camadas citadas anteriormente.

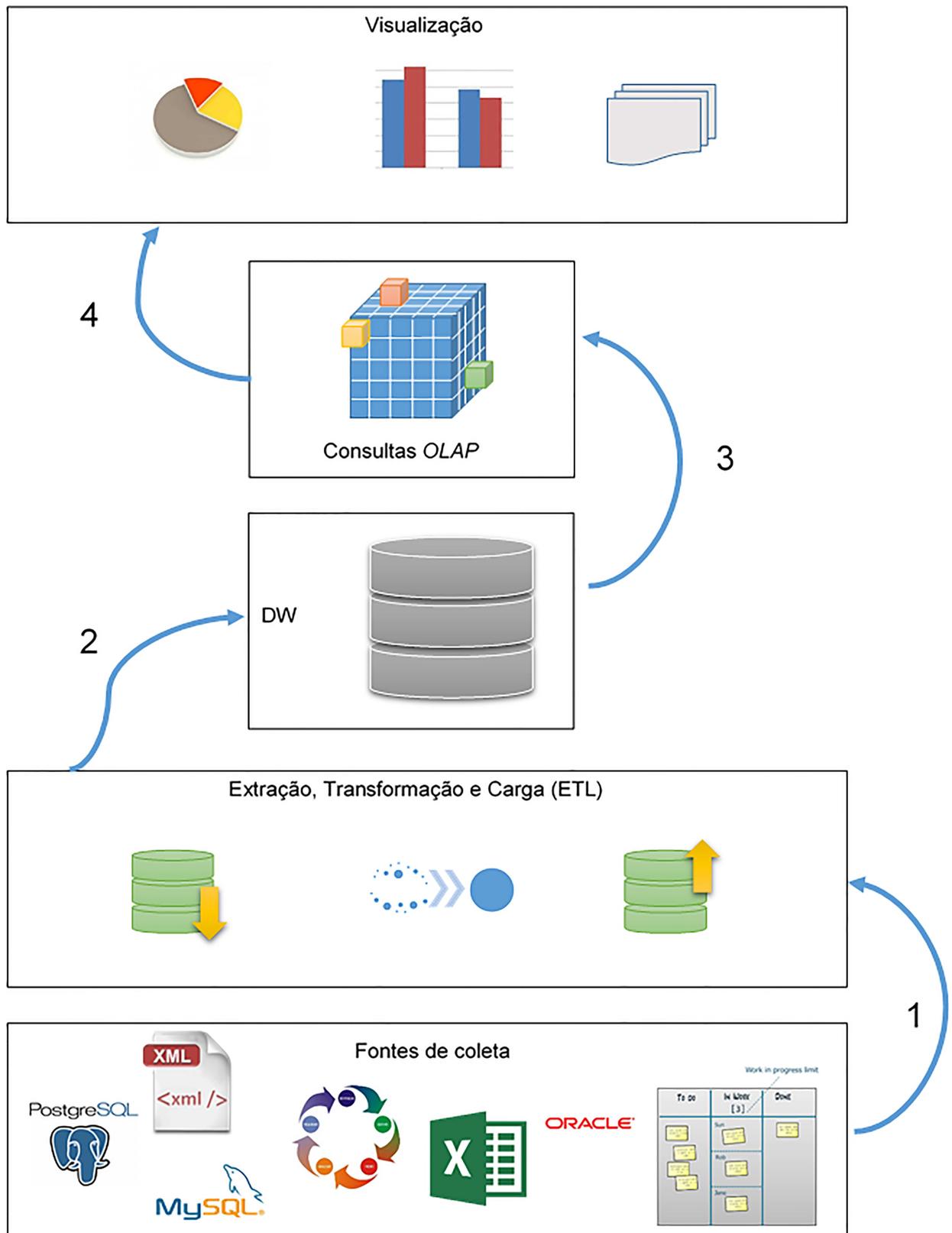


Figura 8 – Arquitetura de um DWing. Fonte (KIMBALL; ROSS, 2002)

- O bloco Fontes de coleta representa a camada de acesso a Dados Operacionais.
- Os blocos ETL e DW representam a camada de Extração e Transformação, e Carga,

respectivamente.

- Os blocos Consultas OLAP e Visualização correspondem à camada de acesso à informação.

As setas 1, 2, 3 e 4 representam o fluxo de dados e informações de um ambiente *DWing*.

4.2.2 Tabela Fato

Uma tabela fato é uma estrutura que contém muitas ocorrências de dados e representa os dados que ocorrem muitas vezes. É a primeira tabela em um modelo dimensional em que as medições numéricas de desempenho são armazenadas e o termo fato é utilizado para representar uma medida de negócio, onde uma medição é feita na intersecção de todas as dimensões do negócio. A lista de dimensões define o grão da tabela de fatos e diz qual o escopo da medição. (KIMBALL; ROSS, 2002)

Fatos aditivos são fatos acumulados. Acumular medições é crucial porque as aplicações de DW quase nunca recuperam uma única linha da tabela fato. Em vez disso, eles trazem de volta centenas, milhares, ou até milhões de linhas fato de cada vez, e o melhor a se fazer com tantas linhas é acumula-las. Como por exemplo, o total (soma) de produtos vendidos.

Fatos semi-aditivo, são fatos que apresentam um resumo de alguma medição, e esse resumo não pode ser efetuado através de uma soma, e sim através de alguma medida de posição, como por exemplo a média aritmética.

Fatos não aditivos são fatos que não podem ser logicamente resumidos entre linhas, ou seja, não pode ser agrupados, como por exemplo porcentagens e proporções. Os fatos mais úteis em uma tabela de fatos são numéricos e aditivos.

Tabelas fato tem duas ou mais chaves estrangeiras que se conectam a chaves primárias das tabelas de dimensão. Quando todas as chaves na tabela fato correspondem à respectiva chave primária corretamente na tabela dimensão correspondente, se diz que as tabelas satisfazem a propriedade de integridade referencial.

Cada tabela fato tem sua própria chave primária composta por um subconjunto de chaves estrangeiras, que é conhecida como chave composta ou concatenada. Toda tabela fato em um modelo dimensional tem uma chave composta e, inversamente, toda tabela que tem uma chave composta é uma tabela fato. Outra maneira de identificar é que toda tabela que expressa um relacionamento de muitos-para-muitos é uma tabela fato.

4.2.3 Tabelas Dimensão

As tabelas dimensão contém as descrições textuais do negócio. Em um modelo dimensional bem projetado, tabelas de dimensão tem muitas colunas ou atributos. Estes atributos descrevem as linhas na tabela de dimensão. Não é incomum uma tabela dimensão ter de 50 a 100 atributos. As tabelas de dimensões tendem a ser relativamente pequenas em termos de número de linhas, mas são enormes em largura com várias colunas. Cada dimensão é definida pela sua chave primária, que serve como base para a integridade referencial com qualquer tabela fato que ela esteja associada.

Atributos de dimensão servem como a fonte principal de restrições de consulta, agrupamentos e relatórios. Em uma consulta(*query*) ou solicitação de relatório, os atributos são identificados pela palavra POR. Por exemplo, total vendido POR semana. O poder do DW é diretamente proporcional à qualidade e profundidade dos atributos de dimensão.

4.3 Processo de Modelagem Dimensional

1. Processo de Negócio

Selecione o processo de negócio a modelar.

Um processo é uma atividade natural realizada na organização que normalmente é apoiado por uma fonte de coleta de dados. É preciso garantir a consistência dos dados efetuando uma única operação de carga, por exemplo.

2. Grão

Declare o grão do processo de negócio.

Declarar o grão significa especificar exatamente o que uma linha da tabela de fato representa. O grão transmite o nível de detalhe associado com as medidas da tabela fatos. Ele fornece a resposta para a pergunta: "Como você descreveria uma única linha na tabela fato?"

3. Dimensões

Escolha as dimensões que se aplicam a cada linha da tabela fato.

Com a escolha de cada dimensão, lista-se atributos que irão detalhar cada tabela de dimensão.

4. Fatos

Identifique os fatos numéricos que irão preencher cada linha da tabela fato. Os fatos são determinado pela resposta à pergunta: "O que estamos medindo?"

Os stakeholders estão muito interessados em analisar a medida de desempenho de cada um dos processos de negócio.

Tipos de Schema

Segundo [KIMBALL e ROSS \(2002\)](#) existem três esquemas para modelagem multidimensional:

4.3.0.1 Esquema estrela

É o esquema mais popular em DW, por que oferece um desempenho melhor das consultas (*queries*) a serem executadas, especialmente em consultas grandes se comparado com o modelo relacional e além disso tem a grande vantagem de ser mais fácil de entender. Consiste de uma tabela fato circundada (ligada por relações) por um conjunto de tabelas dimensão. Quando desenhando, assemelha-se a forma de uma estrela, daí o nome. As tabelas de dimensões não são desnormalizada. ([BALLAR et al., 2006](#))

4.3.0.2 Esquema Floco de Neve

Normalizando e expandindo o tamanho das tabelas dimensão no esquema estrela resultará na implementação do esquema floco de neve. A dimensão é dita floco de neve quando as colunas de baixa cardinalidade da dimensão são separadas em tabelas normalizadas que então se relacionam de volta para a tabela dimensão de origem. ([BALLAR et al., 2006](#))

4.3.0.3 Esquema Constelação de Fatos

O Esquema Constelação de Fatos é constituído de duas ou mais tabelas fatos unidas por uma ou mais dimensões. Esse tipo de esquema pode ser visto como uma coleção de esquemas estrela. O esquema de Constelação de Fatos é mais comumente utilizado, visto que ele pode modelar assuntos múltiplos e inter-relacionados. ([BALLAR et al., 2006](#))

A figura 9 ilustra os três esquemas acima citados.

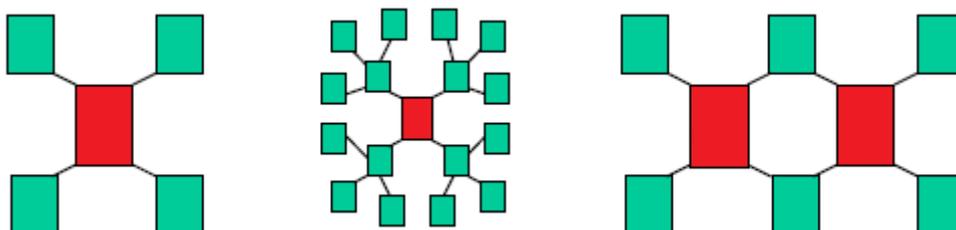


Figura 9 – (a)Esquema Estrela. (b) Esquema Floco de Neve. (c) Esquema Constelação de Fatos. Fonte ([BALLAR et al., 2006](#))

4.4 OLAP

O termo OLAP, inicialmente proposto por [Codd, Codd e Salley \(1993\)](#), vem sendo utilizado para caracterizar as aplicações voltadas ao suporte de atividades de análise, com o objetivo de prover a visualização dos dados sob diferentes perspectivas gerenciais e comportar todas as atividades de análise.

A definição e execução de operações OLAP se baseia no esquema multidimensional de dados que permite visualizar os dados ou seja, é possível modelar e visualizar os dados nas dimensões de um cubo¹, cujas células contêm valores medidos e os lados do cubo definem as dimensões dos dados. Pode-se ter mais de três dimensões no tecnicamente chamado de hiper-cubo, apesar de normalmente os termos cubo e cubo de dados serem usados como sinônimos de hiper-cubo.

Todo modelo multidimensional é organizado em torno de um tema central, que no contexto de DW, é representado por uma tabela fato identificadas na modelagem, e as tabelas dimensão são representadas pelas dimensões do cubo, que podem ter três dimensões, quatro dimensões, n-dimensões. ([HAN; KAMBER,](#)) A figura abaixo ilustra um cubo.

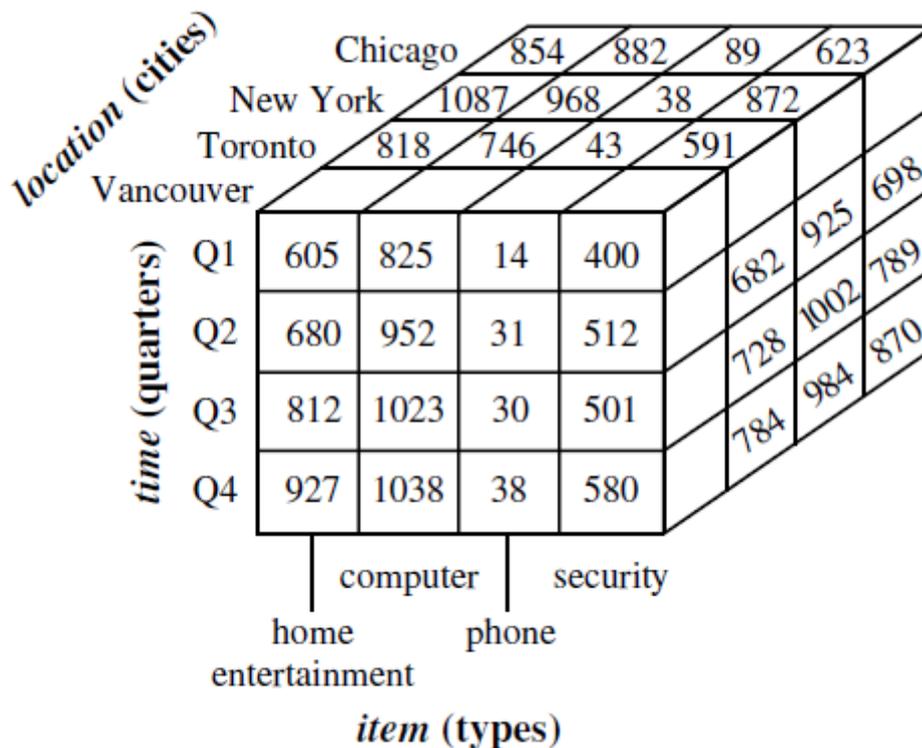


Figura 10 – Cubo 3-D. Fonte ([HAN; KAMBER,](#))

¹ Nomenclatura metaforicamente adotada, posto que matematicamente um cubo possui três, e apenas três, dimensões

A análise multidimensional tem se tornado um caminho popular para estender a capacidade de *queries* e de geração de relatórios, isto é, a análise multidimensional, ao invés de realizar várias consultas com muitas junções, em um banco de dados OLTP. Ela estrutura os dados permitindo um acesso rápido e fácil às respostas das perguntas dos usuários. Esse acesso facilitado e rápido é permitido através de pré cálculos realizados com antecedência e tornando as respostas prontamente disponíveis, ou seja, como as consultas já estão realizadas, já se sabe o resultado de cada uma, então o retorno da consulta é praticamente instantâneo. (BALLAR et al., 2006)

Em um modelo multidimensional, os dados são organizados em diversas dimensões, e cada dimensão contém múltiplos níveis de abstração definidos pelo conceito de hierarquias. Esta organização permite aos usuários flexibilidade para visualizar os dados a partir de perspectivas diferentes e em relações complexas (HAN; KAMBER,), oferecendo aos usuários a possibilidade de olhar para um número enorme de fatores interdependentes envolvidos em um problema de negócio. (BALLAR et al., 2006)

Existem algumas operações OLAP executáveis em cubo que materializam esses diferentes pontos de vista, permitindo a consulta e a análise dos dados de maneira interativa. Assim, as operações OLAP oferecem um ambiente amigável para análise interativa de dados. (HAN; KAMBER,)

As operações mais conhecidas são:

- *Drill-up*: executa agregação em um cubo de dados, podendo ser subindo uma hierarquia de conceitos para uma dimensão ou reduzindo a dimensão. Quando a operação *Roll-up* é realizada por redução de dimensão, uma ou mais dimensões são removidas do cubo. (HAN; KAMBER,)
- *Drill-down*: a operação inversa ao *Drill-up*. Ela navega a partir de dados menos detalhados para dados mais detalhados. Essa operação pode ser realizada descendo uma hierarquia de conceitos para uma dimensão ou introduzindo outras dimensões. (HAN; KAMBER,)
- *Slice and dice*: realiza uma seleção em uma dimensão do cubo, resultando em um subcubo. (HAN; KAMBER,) Define um membro ou um grupo de membros que estão separados, de todas as outras dimensões, e depois realiza o cruzamento com essas dimensões. Por exemplo, as dimensões produto, lojas e tempo, ao isolar dois atributos de produto (leite e biscoito), ao se desejar saber o total de leites e biscoitos vendidos por todas as loja, realiza-se então uma operação *Slice*. A operação *dice* seria então juntar membros de uma dimensão em um cubo com os outros diversos membros das outras dimensões, seguindo o exemplo anterior, o cubo seria montado por datas e por lojas específicas.

- *Pivot*: também chamado de rotação, é uma operação de visualização que rotaciona o eixo do cubo de modo a proporcionar uma apresentação alternativa dos dados, ou seja, realiza uma troca de dimensões, colunas. (HAN; KAMBER,)(BALLAR et al., 2006)

Segundo SILVEIRA, BECKER e RUIZ (2012), um ambiente de DWing contribui para a representação de métricas, sejam elas de produto, processo ou projeto, em um modelo multidimensional, e contribui também para diminuir a intrusão, além de apoiar as atividades de monitoramento. A intrusão pode ser interpretada como o nível de envolvimento das pessoas durante a execução do processo de captura de dados e SILVEIRA, BECKER e RUIZ (2012) e GOPAL, MUKHOPADHYAY e KRISHNAN (2005) afirmam que a intrusão é um dos fatores cruciais de aceitação de um programa de métricas.

Baseado nessas idéias, este Trabalho de Conclusão de Curso apresenta um DW para medidas de escopo, custo e tempo, voltado para projetos ágeis.

5 COMIGRAR – Um estudo de Caso

Existem quatro metodologias de pesquisas que podem ser aplicadas em um estudo empírico: a) pesquisa de opinião; b) experimental; c) pesquisa-ação; e, d) estudo de caso. [RUNESON e HOST \(2009\)](#) justificam a aplicação da metodologia de estudo de caso em Engenharia de software afirmando que ela é uma área multidisciplinar e que há muitos fatores que têm impacto sobre o resultado das atividades da engenharia de software, e portanto, a abordagem do estudo está na fronteira entre o objeto e o ambiente, além da interação entre os dois. Neste trabalho foi escolhida a metodologia de estudo de caso para conduzir um estudo de viabilidade técnica acerca da solução proposta.

5.1 Definição do Estudo de Caso COMIGRAR

5.1.1 Objetivo

O estudo de caso realizado tem caráter exploratório, pois tem o propósito de explorar uma situação e formular hipóteses sobre ela. Essa realidade vai ao encontro da tese formulada por [RUNESON e HOST \(2009\)](#) que afirmam que as pesquisas de Estudo de caso tem uma característica exploratória e foco nas métricas qualitativas.

5.1.2 O caso

O caso de estudo é o gerenciamento do projeto COMIGRAR.

O projeto COMIGRAR é desenvolvido pelo Laboratório Avançado de Produção, Pesquisa & Inovação em Software (LAPPIS), vinculado à Universidade de Brasília.

Consiste na implementação e manutenção da plataforma COMIGRAR (Conferência Nacional sobre Migrações e Refúgio) hospedada no portal <http://participa.gov.br/comigrar>. A plataforma COMIGRAR oferece ferramentas, conteúdos e espaços de diálogo para promoção da participação de atores, especialmente daqueles que tradicionalmente se encontram apartados dos debates sobre política migratória no país. É coordenada pelo Ministério da Justiça, por meio da Secretaria Nacional de Justiça/Departamento de Estrangeiros - DEEST, em parceria com o Ministério do Trabalho e Emprego e o Ministério das Relações Exteriores, com o apoio da Organização Internacional para as Migrações – OIM e do Programa das Nações Unidas para o Desenvolvimento- PNUD. ([COMIGRAR, 2013](#))

O estudo de caso foi executado durante dez semanas entre os meses de setembro e novembro no ano 2013. Período esse que se situa após o início do projeto e antes do

encerramento do mesmo.

O projeto se encontra em um contexto de onze profissionais que, trabalhando semanalmente, com cargas horárias diferentes, e possuem como cliente o Governo Federal. Utilizando o software Noosfero (uma plataforma web para redes sociais que possui as funcionalidades de Blog, e-Portfolios, RSS, discussão temática e agenda de eventos num mesmo sistema. (NOOSFERO, 2013); utilizam como metodologia de desenvolvimento de software o SCRUM, como citado no capítulo 2, baseia-se em histórias de usuário e sprints, contexto esse apropriado para a aplicação da teoria do AgileEVM.

5.1.3 Teoria

O arcabouço teórico de sustentação do estudo de caso foi apresentado nos capítulos 2 - GERENCIAMENTO DE PROJETO DE SOFTWARE, 3 - MÉTRICAS: UMA VISÃO SOBRE AS MEDIDAS DE CUSTO, ESCOPO E TEMPO e no 4 - DATA WAREHOUSE COMO APOIO À ATIVIDADES DE MEDIÇÃO que correspondem à metodologia SCRUM, ao AgileEVM e a *Data Warehouse*, respectivamente.

5.1.4 Questão de pesquisa

O estudo de caso possui as seguintes questões de pesquisa:

1. É possível coletar métricas do AgileEVM a partir do projeto COMIGRAR?
2. É possível aplicar as métricas coletadas do projeto COMIGRAR no *Data Warehouse* desenvolvido neste TCC?

5.1.5 Métodos

RUNESON e HOST (2009) citam três níveis sobre como a coleta de dados pode ser realizada.

- Primeiro Nível: métodos diretos. O pesquisador está em contato direto com o sujeito e coleta dados em tempo real.
- Segundo Nível: métodos indiretos. O pesquisador coleta dados sem interagir com o sujeito durante a coleta.
- Terceiro Nível: análise independente de artefatos. O pesquisador coleta dados que já estão disponíveis.

RUNESON e HOST (2009) ainda apresentam seis tipos métodos de coleta de dados: a) Entrevista; b) Observação; c) Dados Arquivados; d) Métricas; e, e) Checklist.

No contexto do projeto COMIGRAR e deste TCC, apenas os métodos Dados arquivados e Métricas serão utilizados.

5.1.6 Estratégia selecionada

As métricas de interesse são métricas de projeto, oriundas dos conceitos do capítulo 3 - MÉTRICAS: UMA VISÃO SOBRE AS MEDIDAS DE CUSTO, ESCOPO E TEMPO, e portanto são métricas extraídas do processo de gerenciamento do projeto e que estavam presentes em: planilha excel, quadro KANBAN, banco de dados, e registros em papel.

A estratégia de coleta se resume em diminuir ao máximo a intrusão no momento da coleta, ou seja, a intenção foi economizar tempo dos membros do time para extrair dados sobre a produção, mantendo-os focados no desenvolvimento do produto.

Por poucas vezes foi necessário o diálogo com o ScrumMaster, Arthur, para o esclarecimento de dúvidas.

5.2 Procedimentos de coleta de dados

A coleta foi realizada no início e fim de cada uma das sprints. Portanto, ao final de cada sprint, com as métricas que são coletadas, é possível comparar o trabalho planejado e o trabalho efetivamente realizado, em termos de custo, esforço e tamanho, por meio do AgileEVM.

5.3 Executar o estudo de caso com coleta de dados

Antes de iniciar o estudo de caso, foi realizado um estudo comparativo de ferramentas presentes no mercado necessárias para a execução deste trabalho. A seguir um comparativo e a seleção de uma das ferramentas analisadas.

Para selecionar a ferramenta mais adequada os seguintes critérios gerais para seleção foram identificados:

- C1 - A ferramenta seja *open source*;
- C2 - A ferramenta tenha um uso amplo no mercado em que a ferramenta atua;

5.3.1 Ferramenta de Gerenciamento de Projeto

Além dos critérios gerais, outros foram estabelecidos:

- C3 - A ferramenta suporte os conceitos de metodologias ágeis;

- C4 - A ferramenta possua release estável;
- C5 - A ferramenta armazenada as métricas necessárias para os cálculos do Agile-EVM.

As ferramentas comparadas foram o Icescrum e RedMine. A Tabela 8 apresenta um estudo comparativo

Tabela 8 – Comparação Redmine x Icescrum

Característica		
Open Source	X	X
Licença	GPL	GPL
Linguagem de Suporte	Ruby on Rails	JAVA e JSP
Idioma com Suporte	Português, Inglês e mais 32	Inglês e Francês
Idioma da Documentação	Francês, Japonês, Russo, Chinês, Espanhol e Alemão	Inglês e Francês
Atribuição de papéis	X	X
Controle de acesso por papéis	X	-
Suporte a múltiplos projetos	X	X
Wiki	X	-
Geração de gráficos	X	X
Suporte aos SGBDS	MySQL, PostgreSQL, SQLite	Oracle, MySQL, PostgreSQL, SQLServer
Calendário	x	-
Fórum	X	-
E-mail notifications	X	X
Orientado a Metodologia Ágil	-	X
Comunidade Ativa	Fórum, IRC	Site, fórum e blog
Implementa KANBAN	-	X
Última versão Estável em nov/2013	5	R6#10

Uma análise baseada nos critérios gerais e específicos foi realizada e pode ser visualizada na Tabela 9

Após o estudo comparativo, foi optado pela ferramenta Icescrum, pois no critério C3 ela se mostra mais vantajosa do que a ferramenta RedMine, pois o iceScrum é primariamente voltado para metodologias ágeis. A seguir descrição mais detalhada da ferramenta iceScrum.

Tabela 9 – Análise da comparação entre Redmine x Icescrum

Critério	RedMine	Icescrum
C1	X	X
C2	X	X
C3	-	X
C4	X	X
C5	-	-

5.3.2 Icescrum

O iceSCRUM é uma ferramenta de gerenciamento de projetos de desenvolvimento de software que utiliza o framework SCRUM como base conceitual. A ferramenta inclui práticas que vêm de Kanban, como o Work In Process (WIP) para as tarefas urgentes, bem como a criação de diagramas de fluxo cumulativos. (KAGILUM, 2013)

O IceSCRUM é uma aplicação web e se adapta às necessidades de equipes ágeis geograficamente distribuídas. A ferramenta ajuda a implementar as principais práticas ágeis: visão, características, histórias de usuários, a estimativa de esforço com o Planning Poker, quadro de tarefas, entre outras. O iceSCRUM é um software livre e de código aberto, disponível sob GNU Affero GPL V3, e parcialmente disponível sob LGPL v3. Usando post-it virtuais, o iceSCRUM facilita a agilidade. (KAGILUM, 2013). Encontra-se disponível para download em: <http://www.iceSCRUM.org/en/download-en/>. A comunidade está ativa e a última release (R6#10) para download foi disponibilizada em 24/10/2013. (KAGILUM, 2013)

Na ferramenta é possível criar funcionalidades, histórias de usuário associadas às funcionalidades e pode-se atribuir um valor estipulado no planning poker. Para cada história é possível criar tarefas as quais pode-se atribuir o valor em horas para a realização. A ferramenta realiza a coleta dos dados e exibi-los em diversos gráficos. (KAGILUM, 2013)

Contudo, o IceSCRUM, não dá suporte a utilização da técnica AgileEVM.

5.3.3 Ferramenta de suporte a ambiente *DWing*

Para selecionar a ferramenta que melhor suporta um ambiente *DWing*, foram identificadas duas ferramentas: Pentaho Business Analytics e SpagoBI.

O seguintes critério de comparação foram identificados:

- C6 - A ferramenta suporta ETL;
- C7 - A ferramenta permita Análise e Reporte de informações;
- C8 - Maior quantidade de downloads realizado.

A Tabela 10 apresenta um estudo comparativo entre as ferramentas e a Tabela 11 apresenta a análise das ferramentas baseada nos critérios estabelecidos.

Tabela 10 – Comparação Pentaho Business Analytics x SpagoBI

Característica	 pentaho	 SpagoBI
Open Source	Sim	Sim
Licença	Apache	Apache
Linguagem de Suporte	Java	Java
Idioma com Suporte	Inglês, Alemão, Francês	Inglês
Idioma da Documentação	Inglês, Espanhol e Russo	Inglês
Suporte a SGBD	Oracle, MySql, PostgreSQL, Ingres, Microsoft SQL Server	Oracle, MySql, PostgreSQL, Ingres, Microsoft SQL Server
Tecnologia	Java Swing application	Eclipse Java Tool
ETL	Pentaho Data Integration	Talend Open Studio
Data Minig	Weka	Weka
Interpretador	XML	-
Gerador de Código	-	Java
Análise e Reporte	SAIKU	JasperReport, BIRT
Suporte a Big Data	X	X
Análises Preditivas	X	-
Suporte a Clustering	X	-
WebServer	Tomcat	Tomcat
Integração com aplicativos terceiros	Webservices, ERP, CRM systems	ERP, CRM systems
Big Data Sources	Hadoop, cloudera, mongoDB, teradata, splunk, Cassandra, ApacheHBase	Cassandra, mongoDB, ApacheHBase
Teamwork e Colaboração	-	X
Quantidade de downloads no período: 01/12/2013 a 15/12/2013.	23815	5886
Última versão Estável em nov/2013	5.0	5.4.0

Após a realização do estudo comparativo, a ferramenta Pentaho Business Analytics foi selecionada, pois mostrou-se com maior penetração no mercado e maior estabilidade no ramo de atuação da ferramenta. Isso pode ser observado em atendimento ao critério C8, que se mostra quatro vezes maior em relação a outra ferramenta. Além disso o orientando do TCC é familiarizado com a ferramenta e tem interesse em aprofundar seus conhecimentos. A seguir é apresentada uma descrição detalhada da ferramenta.

Tabela 11 – Análise da comparação entre Pentaho x SpagoBI

Critério	Pentaho BA	SpagoBI
C1	X	X
C2	X	X
C6	X	X
C7	X	X
C8	X	X

5.3.4 Pentaho Business Analytics

O Pentaho Business Analytics é um plataforma que simplifica a manipulação de qualquer dado e inclui um arcabouço de ferramentas que facilmente ajudam a analisar, visualizar, explorar, reportar e prever. De código livre, integrada e extensível, o PBA é arquiteturado para garantir que cada membro da equipe - de desenvolvedores a usuário do negócio - possa facilmente traduzir um dado em valor de negócio. É 100

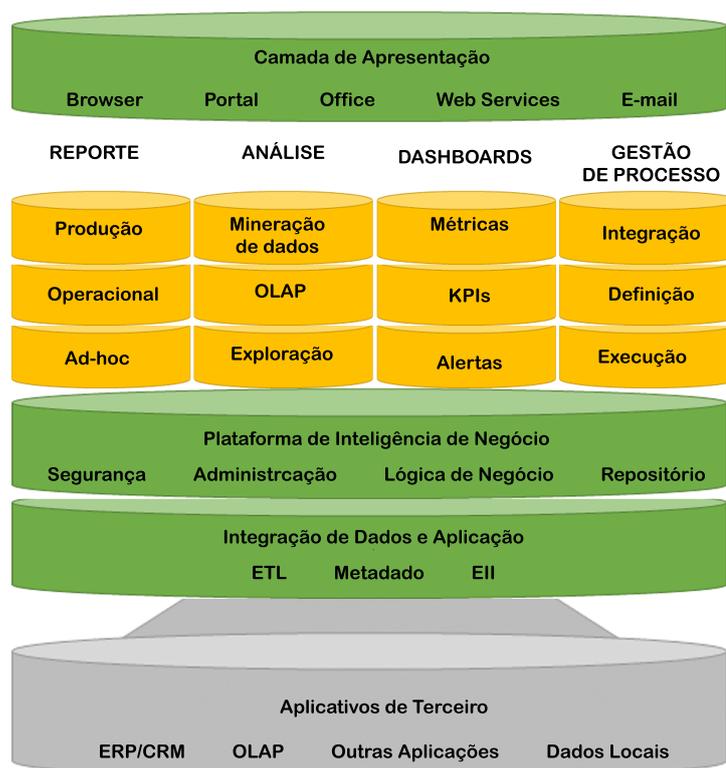


Figura 11 – Arquitetura do Pentaho BA.

As ferramentas Pentaho Data Integration (PDI) e Saiku, dão o suporte para realizar as operações ETL, e as operações OLAP e visualização, respectivamente.

5.3.5 Fontes de Coleta

As fontes de coletas são:

- Quadro KANBAN

– KANBAN é um sistema de controle de produção para a produção *just-in-time* que faz uso pleno das capacidades dos trabalhadores e foi desenvolvido pela TOYOTA. (SUGIMORI et al., 1977). É utilizado como um sinalizador o qual estabelece conexões diretas entre o fornecedor e o cliente, informando a quantidade e o tempo correto para que o produto passe para a próxima etapa. (KATAYAMA, 2011 apud SPEAR; BOWEN, 1999). O quadro KANBAN utilizado no projeto COMIGRAR pode ser visualizado na Figura 12.

- Planilha Eletrônica Open Office
- Ferramenta Icescrum



Figura 12 – Quadro Kanban do Projeto COMIGRAR

5.3.6 Modelagem do Esquema Constelação de Fatos do DW

5.3.6.1 Arquitetura DW

O ambiente *Data Warehouse*, no qual o estudo de caso deste TCC foi implementado, pode ser visualizado na Figura 13.

5.3.6.2 Análise do Processo de Negócio

O processo de negócio selecionado é o contexto do gerenciamento do projeto COMIGRAR, que utiliza a metodologia SCRUM, e por consequência, juntamente com informações de custo, viabilizam a aplicação da técnica do AgileEVM.

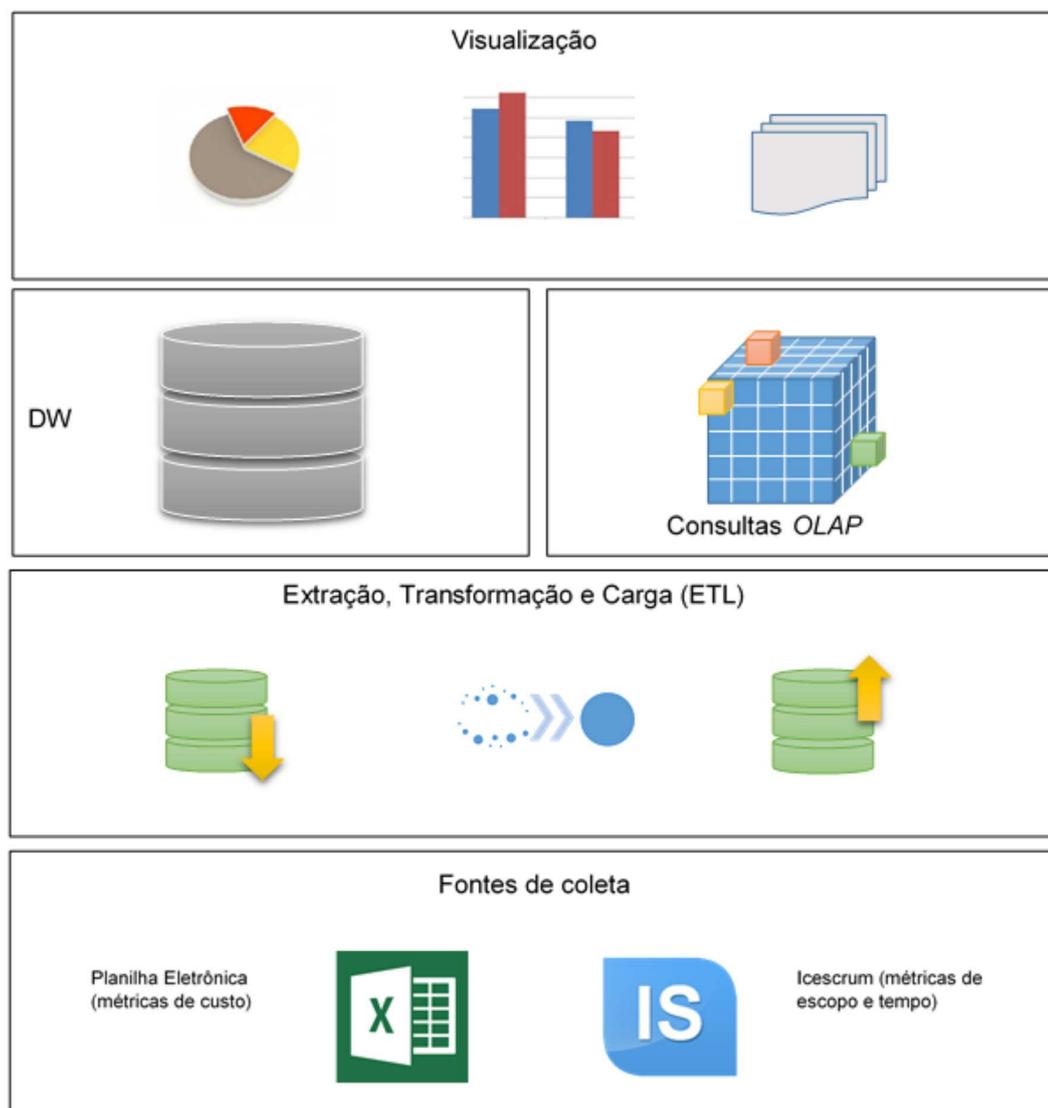


Figura 13 – Visão Arquitetural do ambiente *DW*.

É um processo de total relevância para qualquer empresa, como já citado no capítulo 2, nos quesitos de interesse, por informações relacionadas a prazo, custo e escopo. Baseado nisso, e como já detalhado no capítulo 3, os conceitos do processo de negócio úteis para a modelagem dimensional podem ser observados na Tabela 12.

Logo, algumas respostas (consultas) que o negócio quer obter são:

- Qual o valor agregado (EV) por release?
- Qual a quantidade de pontos planejados por release?
- Qual o total de horas trabalhadas por um profissional em uma release?

Essas consultas são conhecidas como consultas OLAP e seus conceitos foram apresentados no Capítulo 4. Para poder fornecer essas consultas a ferramenta SAIKU foi escolhida para apresentá-las e serão apresentadas na seção 5.5.

Tabela 12 – Processos de Negócio

SCRUM	AgileEVM
<ul style="list-style-type: none"> • Projeto • Release • Sprint • Histórias <ul style="list-style-type: none"> – Pontos • Tarefas <ul style="list-style-type: none"> – Horas Planejadas – Horas Realizadas • Profissionais <ul style="list-style-type: none"> – Valor (R\$) da Hora Trabalhada 	<ul style="list-style-type: none"> • Valor Agregado (EV) • Valor Planejado (PV) • Custo Atual (AC) • Orçamento ao Final (BAC) • Variação do Prazo (SV) • Variação do Custo (CV) • Indicador de Desempenho de Prazo (SPI) • Indicador de Desempenho de Custo (CPI) • Estimativa para Concluir (ETC) • Estimativa à Conclusão (EAC)

5.3.6.3 Granularidade

Declarar a granularidade significa especificar exatamente o que cada registro em uma tabela de fatos representa, ou seja, é especificar a unidade de medida da representatividade do registro. Refere-se ao nível de detalhe de cada unidade de dados do *DW*.

A granularidade permeia todo o ambiente do *DW* e é ela quem define como será a modelagem do esquema, além de interferir diretamente no volume de dados e nos tipos de consultas que podem ser realizadas. Isso quer dizer que quanto maior o nível de granularidade, menos versátil é a realização de consultas (*query*), e quanto menor o nível de granularidade, mais versátil é a realização de consultas (*query*).

Ao realizar a integração de dados, advindo de tabelas diferentes é importante observar que sempre o menor nível de granularidade deve ser respeitado em todas as tabelas.

Para manter a mesma granularidade, foi necessário particionar os dados entre tabelas, permitindo então uma forma de manipulação independente. O particionamento é benéfico para: carga dos dados, acesso aos dados, armazenamento de dados, monitoramento, deleção, entre outros. (INMON, 2005)

Diante disso, a granularidade foi definida a nível de Sprint, pois a partir dela é

possível realizar agregações a nível de *releases* e projetos. Sendo assim, as métricas de tamanho, esforço e custo são primariamente consolidadas no nível da Sprint.

5.3.6.4 Dimensões

A partir da análise de negócio, onde foram identificados os atributos do negócio e, conseqüentemente, das dimensões, além das considerações de granularidade. Com isso foram identificadas três dimensões que fazem parte do processo de negócio. São elas: projeto, usuário e tempo. Essas dimensões com seus atributos podem ser visualizados na Figura 14.

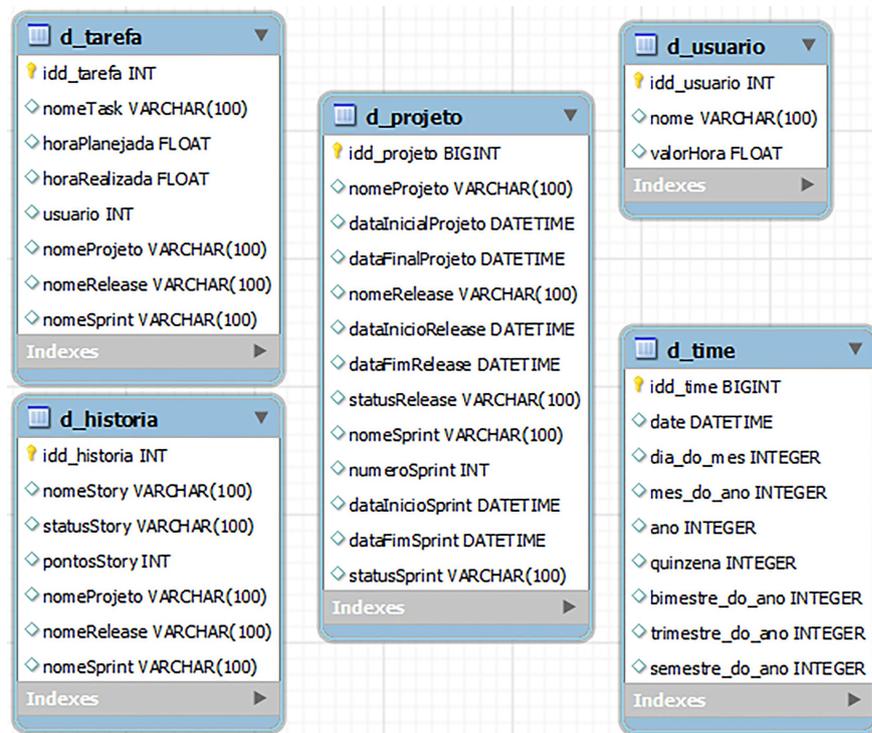


Figura 14 – Tabelas Dimensão do DW.

5.3.6.5 Fatos

A partir das dimensões e suas granularidades, foi possível definir as tabelas fato com suas respectivas medidas.

Na modelagem realizada foi observado que existiam três escopos de métricas. Esses escopos identificados nomearam as tabelas fato de: fato tamanho, fato esforço e fato AgileEVM, referentes às métricas de tamanho, esforço e custo, respectivamente. As tabelas fato e suas medidas, juntamente com as chaves primárias e estrangeiras podem ser visualizadas na Figura 15. De acordo com KIMBALL e ROSS (2002), e como citado no capítulo 4, há três tipos de fatos. A seguir, na Tabela 13, são apresentados os fatos identificados divididos pelos três tipos citados por KIMBALL e ROSS (2002).

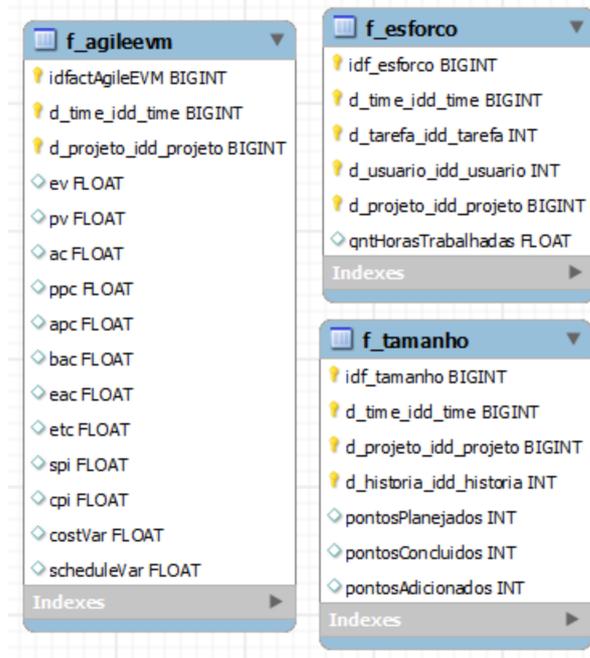


Figura 15 – Tabelas Fato do DW.

Não houveram fatos não-aditivos identificados.

5.3.6.6 AgileEVM Data Warehouse

O modelo final do *Data Warehouse* é mostrado na Figura 16



Figura 16 – AgileEVM DW.

Tabela 13 – Tipos de Tabela Fato

Fatos Aditivos	Fatos Semi-aditivos
<ul style="list-style-type: none"> • qntHorasTrabalhadas (f_esforco) • pontosPlanejados (f_tamanho) • pontosConcluídos (f_tamanho) • pontosAdicionados (f_tamanho) 	<ul style="list-style-type: none"> • ev (f_AgileEVM) • pv (f_AgileEVM) • ac (f_AgileEVM) • ppc (f_AgileEVM) • apc (f_AgileEVM) • bac (f_AgileEVM) • eac (f_AgileEVM) • etc (f_AgileEVM) • spi (f_AgileEVM) • cpi (f_AgileEVM) • costVar (f_AgileEVM) • scheduleVar (f_AgileEVM)

5.3.7 Coleta de Dados

No planejamento inicial do projeto COMIGRAR, foram planejadas quatro Sprints divididas em três releases.

Os dados referentes ao gerenciamento de projeto do projeto COMIGRAR, provenientes do processo e do quadro KANBAN, foram adicionados à ferramenta iceScrum, que por sua vez os armazenava em seu banco de dados. Neste TCC o SGBD utilizado é o MySQL.

É importante ressaltar que o iceScrum não estava sendo utilizado pela equipe do COMIGRAR. Ela foi introduzida no estudo de caso como objeto de análise às questões de pesquisas apresentadas.

Os dados referentes ao gerenciamento de projeto do Projeto COMIGRAR podem ser observados nas figuras 17 e 18.

A figura 17 mostra as funcionalidades do Projeto COMIGRAR e a Figura 18 mostra as histórias de usuário do software.

Nas figuras 19, 20 e 21 são exibidas as informações sobre as releases do projeto COMIGRAR.



Figura 17 – Funcionalidades do projeto COMIGRAR.

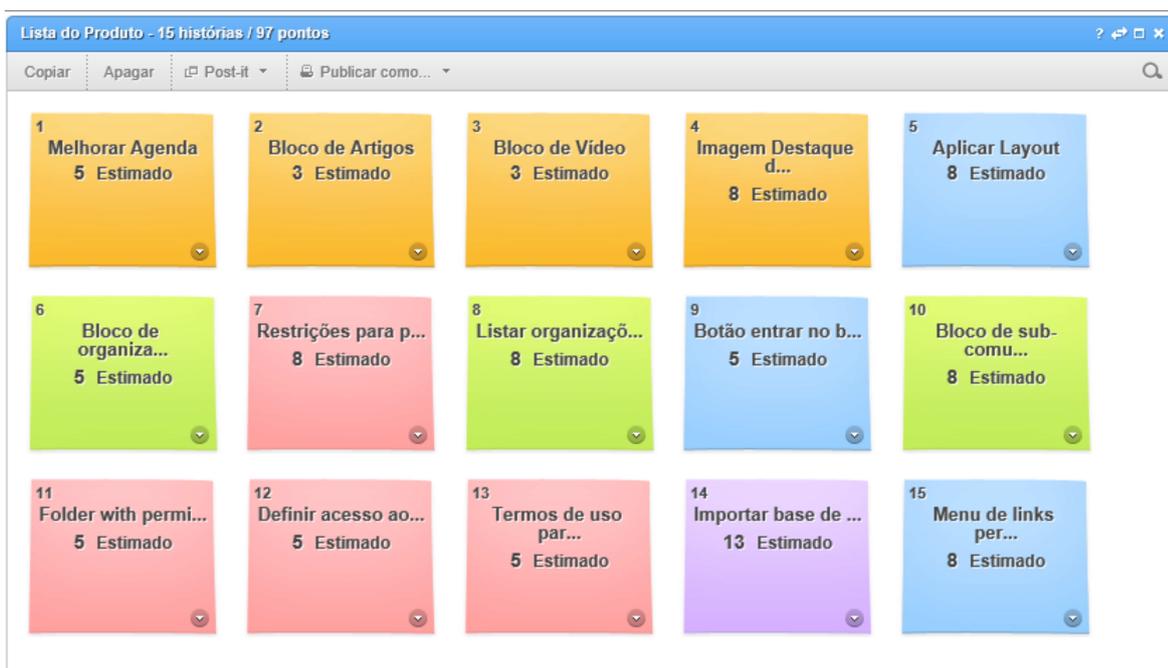


Figura 18 – Histórias de Usuário do projeto COMIGRAR.

Do lado esquerdo das imagens 19, 20, 21 é possível visualizar as histórias de usuário e seus respectivos pontos no planejamento da sprint e da release, e do lado direito as histórias de usuário que foram concluídas. É possível observar nas figuras 19, 20, 21 a visão da releases. Como por exemplo, no topo dessas figuras, a inscrição "Planejar Entrega R3". As sprints também podem ser observadas, pois são descritas como "Iteração X", onde X é substituído pelo número da sprint dentro da release.

Circunscrito em vermelho, é possível visualizar a quantidade de pontos planejados e a quantidade de pontos concluídos do projeto, conforme apresentado nas Figuras 19, 20, 21

Uma limitação da ferramenta iceScrum é que a mesma não gerencia a quantidade de horas reais trabalhadas por um profissional. É planejado a quantidade de horas a serem trabalhadas em uma tarefa, e essa mesma quantidade é associada ao atributo "horas

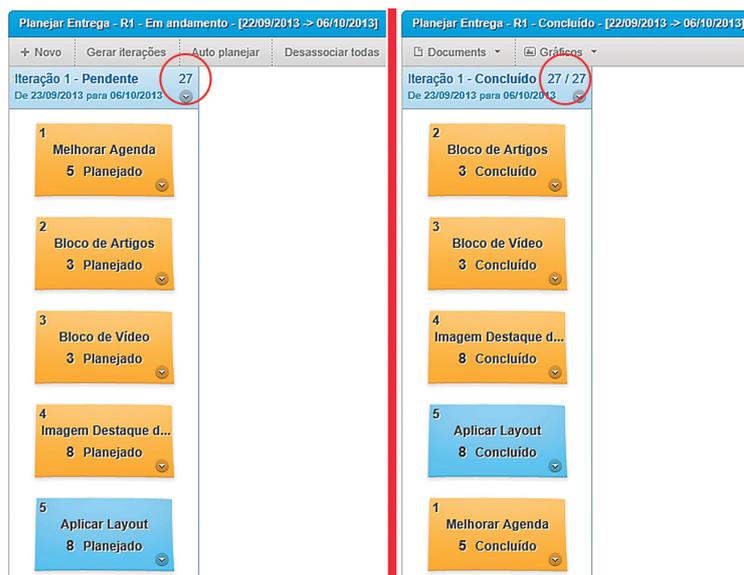


Figura 19 – Release 1 com uma Sprint

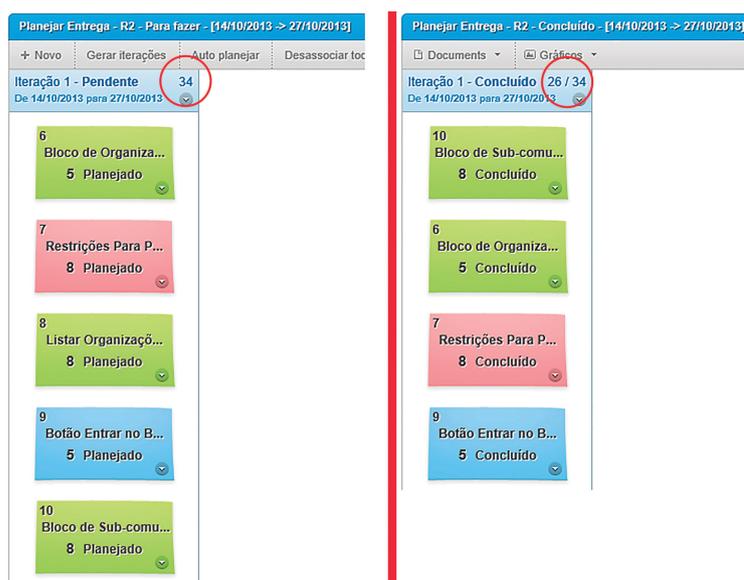


Figura 20 – Release 2 com uma Sprint

restantes" que é gerenciado de forma decremental, e ao concluir a história, este atributo é setado como 0 (zero) automaticamente. Ou seja, se houver variação para mais ou para menos, não é possível saber, pois o atributo, no final, será sempre zero, não sabendo qual o valor inicial ou a variação do mesmo.

A outra fonte de coleta utilizada foi a planilha eletrônica, que possuía informações de custo dos membros do time COMIGRAR. A planilha pode ser visualizada na Figura 22.

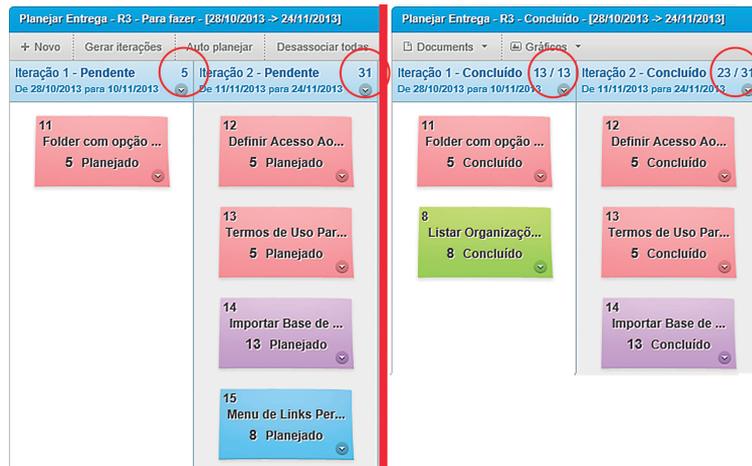


Figura 21 – Release 3 com duas Sprints.

	A	B
1	nome	valorHoraTrabalhada
2	Artur	24,25
3	Carlos	21,25
4	Daniel	24,25
5	David	24,25
6	Fabio	21,25
7	Gabriela	21,25
8	Leandro	21,25
9	Luiz	21,25
10	Marcos	20,00
11	Tales	21,25
12	Valessio	10,75

Figura 22 – Lista de profissionais do time Scrum

5.3.7.1 ETL

O processo de ETL, como apresentado no Capítulo 4, é composto de três passos: extração, transformação e carga. A ferramenta PDI fornece dois recursos: 1 - Transformação (*transformation*, em inglês), e 2 - Serviço (job, em inglês). Esses recursos são úteis, pois, ao realizar as transformações é possível realizar o ETL, e com os *jobs* é possível definir um fluxo de execução das transformações e conseqüentemente a de carga do *DW*.

A Figura 23 ilustra a transformação de ETL para a dimensão projeto. Pela transformação é possível observar que ela também contempla as dimensões tarefa e historia. Essa característica desta transformação ajuda a garantir a integridade referencias dos atributos.

O *step* info projeto é a etapa de extração, E, do ETL. Os *steps* 1) selec.registros

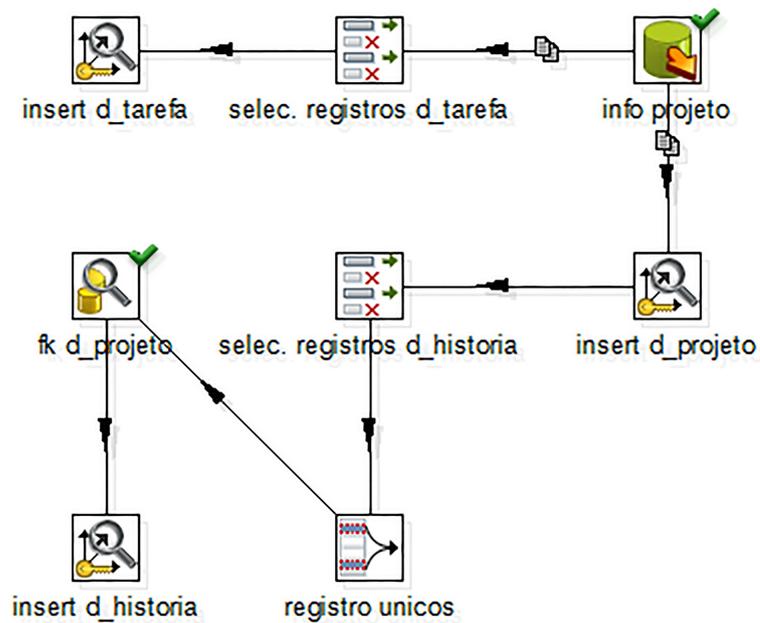


Figura 23 – Transformação da Dimensão Projeto, História e Tarefa

d_tarefa; 2) selec. registros d_historia; e, 3) registros unicos, fazem parte da etapa T, do ETL. E por fim, os steps 4) insert d_tarefa; 5) insert d_historia; e, 6) insert d_projeto, fazem parte da etapa L, do ETL. A Figura 24 ilustra também o processo ETL para a fato tamanho. O step "extraíndo info de BAC, AC, PPC e APC", é a etapa E, do ETL. Os steps Calculator e Select values, são a etapa T, do ETL. E os steps "insert d_time" e "insert f_agilevm" são a etapa L, do ETL.

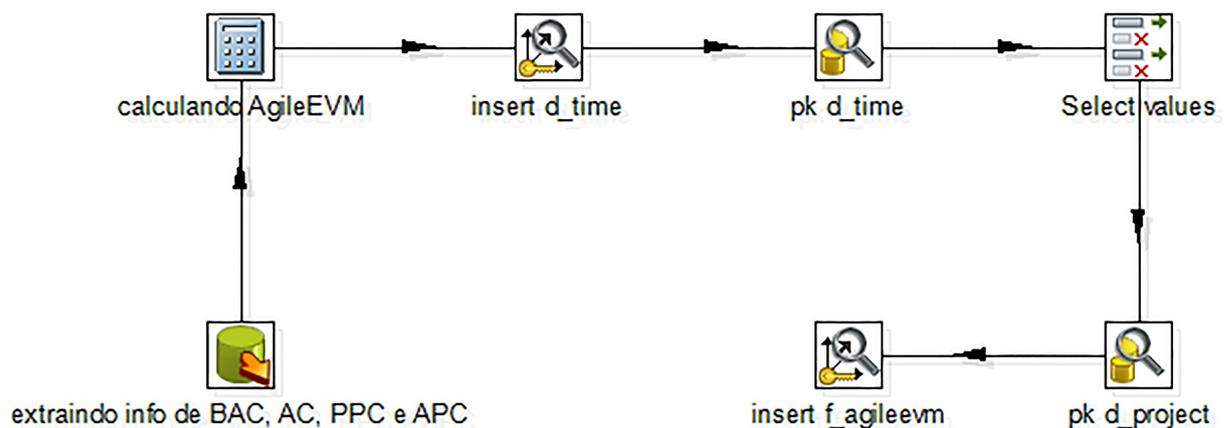


Figura 24 – Transformação da Fato AgileEVM

Um job, Figura 25 foi criado para a execução automática das transformações criadas. Neste job é possível visualizar o fluxo de execução das mesmas e conseqüentemente da carga do DW.



Figura 25 – Job do Projeto COMIGRAR.

As demais transformações podem ser visualizadas no Apêndice A, e estão disponíveis no repositório https://github.com/fernandosouzs/repositorio_COMIGRAR.

Estão também no repositório os *scripts* do banco de dados do icescrum (com as informações do projeto COMIGRAR) e do *DW* utilizado neste TCC.

5.4 Analisar dados coletados

O objetivo da análise é tirar conclusões a partir dos dados, mantendo uma clara cadeia de evidências. Inclui a análise estatística, a análise correlacionada, o desenvolvimento de modelos preditivos, e testes de hipóteses. (RUNESON; HOST, 2009)

Como as métricas deste TCC vieram de metodologia SCRUM e do AgileEVM, essa fase, de analisar dados coletados, foge do escopo deste TCC, pois as métricas já possuem uma análise, um nível de formalismo e validade consolidada no mercado corporativo.

5.5 Reportar

Como apresentado na subseção 5.3.6.2, abaixo é possível visualizar algumas consultas.

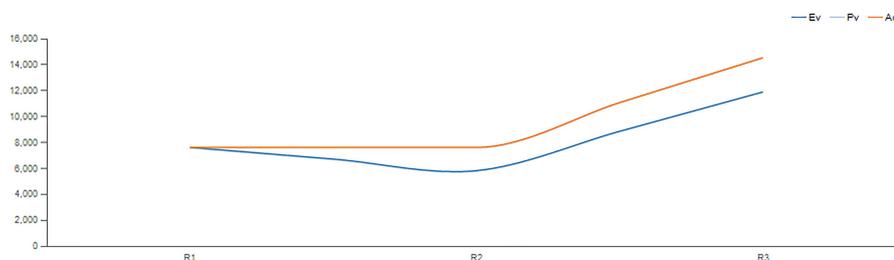


Figura 26 – Valor Planejado, Valor Agregado, Custo Atual e BAC por Release gráfico de linhas

Na Figura 26 é apresentada a curva dos valores *Earned Value*, *Planned Value* e *Actual Cost* ao longo das Releases 1, 2 e 3.

O Projeto COMIGRAR não apresenta variação nos custos, pois os mesmos são calculados considerando as horas trabalhadas por cada profissional que possui horas fixas de trabalho. Ou seja, devido à característica do projeto de conceder bolsas aos profissionais, independente da quantidade de horas que se trabalha, o salário é o mesmo, logo é costume trabalhar a quantidade de horas contratadas e portanto não há horas extras ou sobra de horas.

Devido a isso, de não haver variação em horas realizadas em relação a horas planejadas, as métricas PV e AC serão sempre iguais. Portanto, estas curvas, representadas em um gráfico, se sobrepõem, como pode ser visualizado na Figura 26.

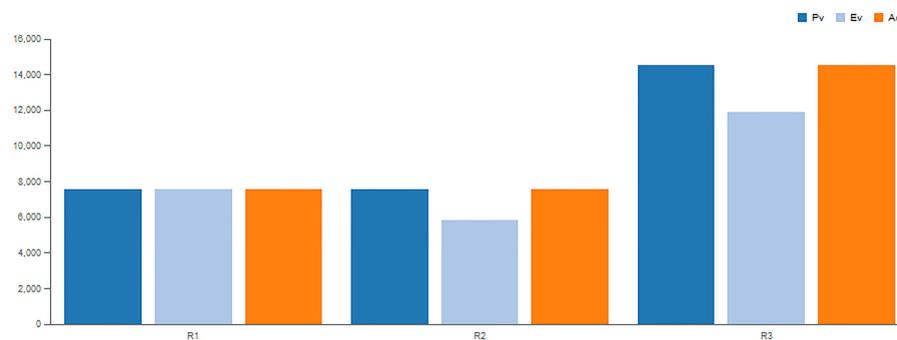


Figura 27 – Valor Planejado, Valor Agregado e Custo Atual por Release, gráfico de barras

No gráfico 27 é apresentado um comportamento previsível devido ao contexto em que o projeto COMIGRAR se enquadra.

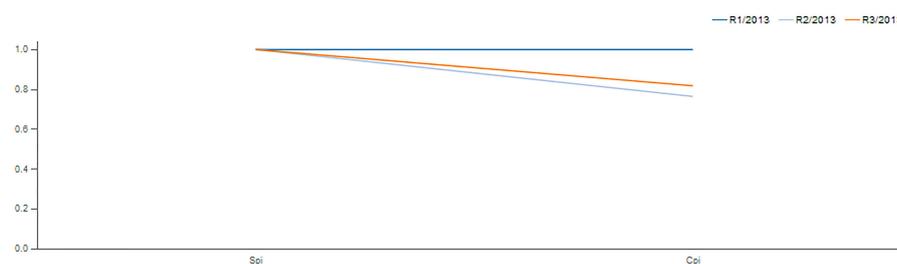


Figura 28 – Indicadores de Desempenho - CPI e SPI

A Figura 28 apresenta os valores SPI e CPI nas Releases 1, 2 e 3.

É possível observar que não houve variação do indicador SPI, logo, de acordo com a teoria apresentada no Capítulo 3, o índice indica que houve um equilíbrio entre a quantidade de horas realizadas em relação às horas planejadas.

Já o CPI indica que, com valores menores do que um (1), o desempenho de custo está acima do limite planejado.

5.5.1 Resultados do Estudo de Caso - COMIGRAR

A partir do processo de ETL e com as funcionalidades da ferramenta SAIKU é possível responder as questões de pesquisas apresentadas na seção 5.1.4.

1. É possível coletar métricas do AgileEVM a partir do projeto COMIGRAR?

Sim. Com as informações coletadas do projeto COMIGRAR foi possível extrair todas as métricas primitivas e calcular as métricas derivadas, relacionadas ao AgileEMV.

2. É possível aplicar as métricas coletadas do projeto COMIGRAR no *Data Warehouse* desenvolvido no TCC?

Sim. O modelo dimensiona desenvolvido foi aplicado ao contexto do projeto COMIGRAR. Uma vez resolvida a etapa de ETL, o modelo dimensional proposto se mostrou tecnicamente viável, uma vez que, possibilitou obter informações relacionadas a custo, tamanho e esforço, em projetos que utilizam métodos ágeis. Dessa maneira, pode-se afirmar que a partir do modelo proposto, é possível se utilizar diferentes fontes de coleta, inclusive diferentes das propostas neste estudo de caso, que será possível visualizar informações de métricas obtidas pela aplicação da técnica AgileEVM.

6 Conclusão

Neste trabalho foi apresentado, a partir das referências teóricas, a construção e um exemplo de uso da solução proposta. Nesse sentido, foi utilizado um conjunto de ferramentas que auxiliam e estabelecem um ambiente *DWing* perpassando pelas etapas de extração, transformação, carga, agregações, e visualização de métricas, medidas e indicadores de tamanho, esforço e custo em projetos de desenvolvimento de software com uso de métodos ágeis. Este ambiente se mostra não dependente de ferramentas de ETL, sendo possível sua implementação com várias outras fontes de coleta.

O trabalho realizado foi capaz de atender os objetivos específicos:

1. Descrever as características do gerenciamento de projetos tradicionais e projetos ágeis, no Capítulo 2.
2. Utilizar a adaptação da técnica tradicional de monitoramento de projetos, EVM, para o domínio de projetos ágeis. A utilização foi apresentada no Capítulo 5
3. Descrever as principais características da aplicação da técnica de análise de valor agregado nos contextos de projetos tradicionais e ágeis, no Capítulo 3
4. Utilizar ferramentas de software livre, no Capítulo 5 foi apresentado comparações e os critérios de escolha para as ferramentas.
5. Coletar dados de um projeto real de desenvolvimento de software, o estudo de caso presente no Capítulo 5 foi capaz de demonstrar.

Atendeu também aos objetivos específicos, 4) com o uso de ferramentas de software livre, onde foi possível realizar a construção de um ambiente *DWing*. Neste ambiente foi possível 5) coletar métricas do AgileEVM a partir de um projeto real de desenvolvimento de software, o COMIGRAR. Essas métricas foram facilmente aplicadas no ambiente *DWing*, pois o mesmo, baseado em SCRUM e AgileEVM, mostrou-se capaz de satisfazer a solução.

Dessa maneira, com o atendimento aos objetivos específicos, o objetivo geral do TCC foi alcançado: Implementar uma proposta de solução que permita a coleta automática e semiautomática de métricas de gerenciamento; o armazenamento em um repositório centralizado; definição de um cubo de dados; uma forma de visualização destas; com o suporte de um ambiente de *DWing*.

Com o objetivo alcançado pode-se então afirmar, de acordo com as afirmações a serem alcançadas na subseção 1.3 que: a) o ambiente *DWing* desenvolvido é uma solução

para automatizar a coleta de métricas gerenciais de custo, tempo e escopo, que oferece uma diminuição da interferência humana na coleta das métricas, pois o processo de extração, transformação e carga é realizado executando-se apenas um *step*: o *job*, como apresentado no Capítulo 5. O modelo *DW* construído do tipo Constelação de Fatos tem sua parcela de contribuição, pois foi construído baseado em uma granularidade que independe das fontes de coleta, podendo ser utilizado em qualquer outro projeto que utilize a metodologia Scrum.

Esse ambiente confirma a afirmação de que é possível integrar as ferramentas: planilha eletrônica, iceScrum e Pentaho para a interpretação de métricas de gerenciamento de projeto de software, porém não confirma que a ferramenta iceScrum satisfaz a utilização da técnica AgileEVM, pois não é completa.

Essa percepção pode ser reforçada devido ao fato de que os membros do time citados no estudo de caso, não fizeram uso da ferramenta iceScrum, conforme discutido no capítulo 5 na subseção 5.3.7. Pode-se pontuar que as métricas foram coletadas diretamente do banco de dados da ferramenta e através de pequenos e curtos diálogos com o Scrum-Master do time. Isso evidencia que o time não teve nenhum tipo de sobrecarga, para que fosse possível coletar e visualizar métricas de processo e projeto. Utilizou-se os próprios eventos previstos no Scrum para que os dados fossem definidos e fornecidos pelo time.

A solução apresentada se mostra relevante à organizações que desejem gerenciar seu portfólio de projetos coletando grandes quantidades de dados e sendo possível visualizá-los facilmente.

6.1 Trabalhos Futuros

Para trabalhos futuros este TCC deixa o legado de um ambiente *Data Warehousing* capaz de coletar métricas de custo, tamanho e esforço em projetos ágeis.

Este legado estimula ao incremento do ambiente em relação a métricas de processo, produto e projeto, além da integração com outras ferramentas, como por exemplo, de bug tracking, teste, entre outras.

Referências

- 5CQUALIBR. *Desenvolvimento Ágil*. 2010. Disponível em: <<http://www.softwarepublico.gov.br/5cqualibr/xowiki/DesenvolvimentoAgil>>. Citado na página 31.
- BALLAR, C. et al. *Dimensional Modeling: In a Business Intelligence Environment*. 1ª. ed. New York: IBM, 2006. Citado 4 vezes nas páginas 15, 56, 58 e 59.
- BARCELLOS, M. P. Medição de software: Um importante pilar da melhoria de processos de software. *Engenharia de Software Magazine*, n. 24, p. 31–36, 2010. Citado na página 23.
- BASILI, V. R.; CALDIERA, G.; ROMBACH, H. D. *The Goal Question Metric Approach*. [S.l.]: Encyclopedia of Software Engineering, 1996. Citado na página 39.
- BECK, K.; GAMMA, E. Test infected: Programmers love writing tests. *Java Report*, v. 3, n. 7, p. 51–56, 1998. Citado na página 40.
- BECK, K. et al. *Manifesto para Desenvolvimento Ágil de Software*. 2001. Disponível em. Disponível em: <<http://www.agilemanifesto.org/>>. Citado 2 vezes nas páginas 24 e 36.
- BOUMAN, J. V. D. R. *Pentaho Solutions: Business Intelligence and Data Warehousing with Pentaho and MySQL*. 1ª. ed. Indianapolis, IN: Wiley, 2009. Citado na página 49.
- CASTELLANOS, M. et al. Predictive business operations management. In: BHALLA, S. (Ed.). *Databases in Networked Information Systems*. [S.l.]: Springer Berlin Heidelberg, 2005, (Lecture Notes in Computer Science, v. 3433). p. 1–14. Citado na página 24.
- CHULANI, S. et al. Metrics for managing customer view of software quality. In: *Software Metrics Symposium, 2003. Proceedings. Ninth International*. [S.l.: s.n.], 2003. p. 189–198. Citado na página 24.
- CMMI. [S.l.]. Citado na página 40.
- CODD, E. F.; CODD, S. B.; SALLEY, C. T. *Providing OLAP (On-Line Analytical Processing) to User-Analysis: An IT Mandate*. [S.l.]: E. F. Codd & Associates, 1993. Citado na página 57.
- COHN, M. *Succeeding With Agile: Software Development Using SCRUM*. 1ª. ed. Boston, USA: Addison-Wesley: Pearson Education, 2012. Citado na página 34.
- COMIGRAR. *I Conferência Nacional sobre Migrações e Refúgio - COMIGRAR*. 2013. Disponível em: <<http://participa.gov.br/comigrar>>. Citado na página 61.
- FENTON, N.; PFLEEGER, S. L. *Software Metrics: A Rigorous and Practical Approach*. 2ª. ed. Boston, MA, USA: PWS Publishing Company, 1997. Citado na página 39.
- FLEMING, Q. W.; KOPPELMAN, J. M. Earned value project management: A powerful tool for software projects. *CrossTalk: The Journal of Defense Software Engineering*, p. 19–23, 1998. Citado 3 vezes nas páginas 41, 42 e 43.

- FOLLECO, A. et al. Learning from software quality data with class imbalance and noise. In: *SEKE*. [S.l.]: Proceeding of the 19th International Conference on Software Engineering and Knowledge Engineering, 2007. p. 487–493. Citado na página 24.
- GOPAL, A.; MUKHOPADHYAY, T.; KRISHNAN, M. The impact of institutional forces on software metrics programs. *Software Engineering, IEEE Transactions on*, IEEE, v. 31, n. 8, p. 679–694, 2005. Citado na página 59.
- HAN, J.; KAMBER, M. [S.l.: s.n.]. Citado 4 vezes nas páginas 15, 57, 58 e 59.
- HELDMAN, K. *Gerência de projetos: Guia para o exame oficial do PMI*. 5ª. ed. São Paulo: Elsevier, 2009. Citado 3 vezes nas páginas 41, 43 e 44.
- IAMAMOTO, E.; GAILLAND, R. *A Revolução Digital e a Sociedade do Conhecimento*. IME - Universidade de São Paulo. Disponível em: <<http://www.ime.usp.br/~is/ddt/mac333/aulas/tema-3-08abr99.html>>. Citado 2 vezes nas páginas 35 e 41.
- IEEE, S. E. S. C. *IEEE Standard for a Software Quality Metrics Methodology*. Piscataway, NJ, USA, 1998. Citado 2 vezes nas páginas 29 e 39.
- INMON, W. *Building the data warehouse*. [S.l.]: Wiley, 2005. (Wiley technology publishing). ISBN 9780764599446. Citado 2 vezes nas páginas 25 e 70.
- ISO/IEC. *ISO/IEC 15939:2007: Systems and software engineering - Measurement process*. Geneva, 2007. Citado 2 vezes nas páginas 23 e 39.
- ISO/IEC. *ISO/IEC 25023:2011 Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - Measurement of system and software product quality*. Geneva, 2011. Citado 3 vezes nas páginas 15, 39 e 40.
- KAGILUM. *Getting Start with Icescrum*. 2013. Disponível em: <<http://www.icescrum.org/>>. Citado na página 65.
- KAN, S. H. *Metrics And Models In Software Quality Engineering*. 2ª. ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2002. Citado na página 23.
- KATAYAMA, E. T. A contribuição da indústria da manufatura no desenvolvimento de software. *Dissertação (Mestrado em Ciência da Computação) - Instituto de Matemática e Estatística, Universidade de São Paulo*, 2011. Citado 2 vezes nas páginas 35 e 68.
- KEITH, C. *Agile Game Development With SCRUM*. 1ª. ed. Boston: Addison-Wesley, 2010. Citado na página 38.
- KIMBALL, R.; ROSS, M. *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*. 2ª. ed. New York: Wiley, 2002. Citado 10 vezes nas páginas 15, 24, 49, 50, 51, 52, 53, 54, 56 e 71.
- LORENZ, E. N. *The Essence Of Chaos*. [S.l.]: Taylor & Francis, 1995. (Jessie and John Danz lectures). Citado na página 35.
- MARTINS, J. C. C. *Técnicas Para Gerenciamento de Projetos de Software*. 1ª. ed. Rio de Janeiro: Brasport, 2007. Citado 2 vezes nas páginas 30 e 34.
- MOUNTAINGOAT. *What is SCRUM methodology?* 2013. Disponível em. Disponível em: <www.mountaingoatsoftware.com>. Citado 3 vezes nas páginas 23, 36 e 37.

- NERI, H. R. Análise, projeto e implementação de um esquema molap de data warehouse, utilizando o sgbd-or oracle 8.1. In: . [S.l.]: Universidade Federal da Paraíba, 2002. Citado na página 24.
- NOOSFERO. *Blog do Projeto Noosfero*. 2013. Disponível em: <<http://softwarelivre.org/noosfero>>. Citado na página 62.
- OLIVEIRA, P. A. *Ferramenta de Construção de Data Warehouse*. 2007. Citado na página 24.
- PALZA, E.; FUHRMAN, C.; ABRAN, A. Establishing a generic and multidimensional measurement repository in cmmi context. In: *Proceeding of the 28th Annual NASA Software Engineering Workshop (SEW-28)*. [S.l.]: IEEE Computer Society Press, 2003. p. 12–20. Citado na página 24.
- PETERSON, C. D.; OLIVER, M. E. Ev-lite earned value control for fast paced projects. In: *32th Annual Project Management Institute Seminars and Symposium*. Nashville, TN: [s.n.], 2001. Citado 2 vezes nas páginas 43 e 44.
- PFLEEGER, S. L. *Engenharia de Software: Teoria e Prática*. 2ª. ed. [S.l.]: Pearson/Prentice Hall, 2004. Citado na página 29.
- PMI. *Um Guia do Conhecimento em Gerenciamento de Projetos (Guia PMBOK®)*. Em português. 4ª. ed. Pennsylvania, EUA, 2008. Citado 9 vezes nas páginas 15, 23, 29, 31, 32, 41, 42, 43 e 44.
- POPPENDIECK, M.; POPPENDIECK, T. *Implementando o Desenvolvimento Lean de Software: Do conceito ao dinheiro*. [S.l.: s.n.], 2011. Citado na página 40.
- PRESSMAN, R. S. *Engenharia de Software: Uma Abordagem Profissional*. 7ª. ed. New York: McGraw Hill, 2011. Citado na página 39.
- RUNESON, P.; HOST, M. Guidelines for conducting and reporting case study research in software engineering. *Empirical Softw. Engg.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 14, n. 2, p. 131–164, abr. 2009. Citado 3 vezes nas páginas 61, 62 e 78.
- SATO, C. E. Y.; DERGINT, D. E. A.; HATAKEYAMA, K. O papel evolutivo do gerente de projetos. *XI Seminário Latino-Iberoamericano de Gestión Tecnológica.*, ALTEC, 2005. Citado na página 35.
- SCHWABER, K. *Agile Project Management With Scrum*. 1ª. ed. Washigton: Microsoft-O'Reilly Media, 2004. Citado 4 vezes nas páginas 15, 36, 37 e 38.
- SCHWABER, K. What is scrum? *VOLARO With AgileSparks*, VOLARO, 2010. Citado na página 30.
- SCHWABER, K.; SYTHERLAND, J. Guia do scrum. *As regras do Jogo*, 2011. Disponível em: <<https://www.scrum.org>>. Citado na página 36.
- SCRUMALLIANCE. *What is Scrum?* 2013. Disponível em. Disponível em: <www.scrumalliance.org>. Citado na página 36.
- SCRUMMETHODOLOGY. *What's unique about SCRUM?* 2013. Disponível em. Disponível em: <www.scrummethodology.com>. Citado na página 36.

- SILVA, R. O. *Teorias da Administração*. São Paulo: Pioneira Thomson Learning, 2001. Citado 4 vezes nas páginas 30, 31, 34 e 35.
- SILVEIRA, P. S. Processo de etc. orientado a serviço para um ambiente de gestão de qualidade de software. PUCRS, 2007. Citado na página 24.
- SILVEIRA, P. S.; BECKER, K.; RUIZ, D. D. Spdw+: A seamless approach for capturing quality metrics in software development environments. *Software Quality Journal*, v. 18, n. 2, p. 227–268, 2012. Citado 5 vezes nas páginas 23, 24, 40, 49 e 59.
- SPEAR, S.; BOWEN, H. K. Decoding the dna of the toyota production system. *Harvard Business Review*, v. 77, p. 96–108, 1999. Citado na página 68.
- SPIES, E. H.; RUIZ, D. D. Proposta de um plano de métricas para o monitoramento de projetos scrum. *Brazilian Workshop on Agile Methods*, v. 3, p. 84–95, 2012. Citado na página 40.
- STACEY, R. D. *Complexity and creativity in organizations*. 1. ed. [S.l.]: Berrett-Koehler Publishers, 1996. Citado na página 35.
- SUGIMORI, Y. et al. *International Journal of Production Research*, v. 15, n. 6, p. 553–564, 1977. Citado na página 68.
- SULAIMAN, T.; BARTON, B.; BLACKBURN, T. Agileevm - earned value management in scrum projects. *AGILE '06 Proceedings of the conference on AGILE 2006*, IEEE, 2006. Citado 4 vezes nas páginas 24, 40, 45 e 46.
- VARGAS, R. V. *Análise de valor agregado (EVA) em projetos*. Rio de Janeiro: Brasport, 2002. Citado 2 vezes nas páginas 23 e 40.
- VARGAS, R. V. *Gerenciamento de Projetos: Estabelecendo Diferenciais Competitivos*. 7ª. ed. Rio de Janeiro: Brasport, 2009. Citado 6 vezes nas páginas 15, 29, 32, 33, 43 e 44.
- VARGAS, R. V. *Análise de Valor Agregado: Revolucionando o Gerenciamento de Prazos e Custo*. 5ª. ed. Rio de Janeiro: Brasport, 2011. Citado 4 vezes nas páginas 23, 29, 41 e 43.

Apêndices

APÊNDICE A – ETL



Figura 29 – ETL Dimensão Usuário



Figura 30 – ETL Fato Esforço

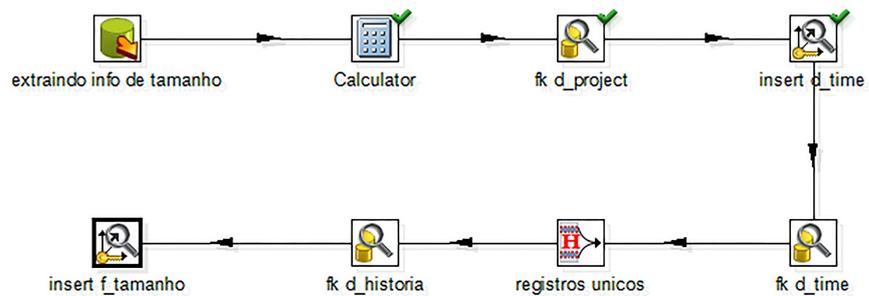


Figura 31 – ETL Fato Tamanho

APÊNDICE B – Gráficos do Negócio

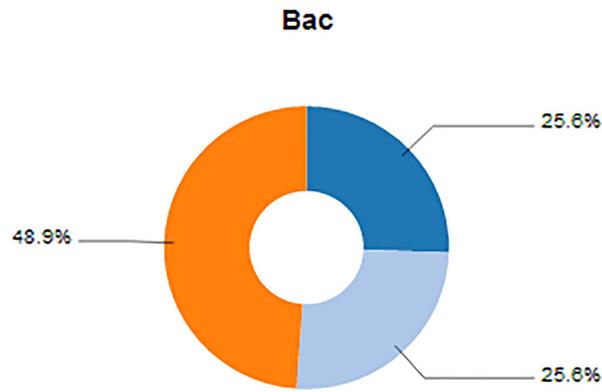


Figura 32 – Porcentagem da representatividade do BAC por Release

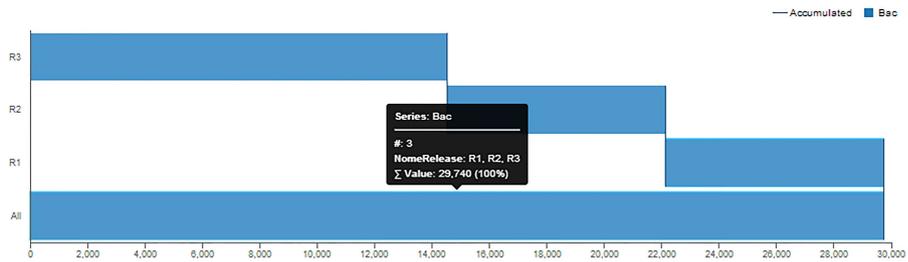


Figura 33 – BAC acumulado do projeto COMIGRAR

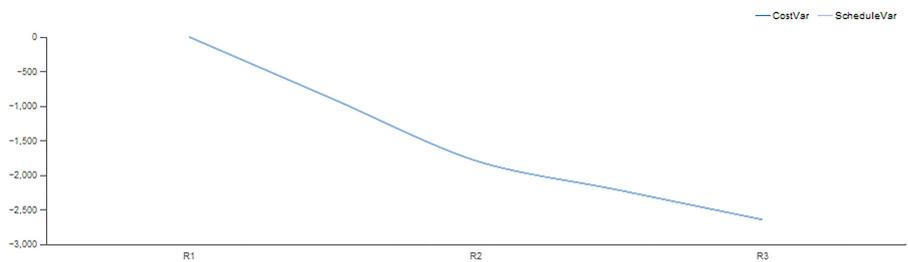


Figura 34 – Variações de prazo e escopo

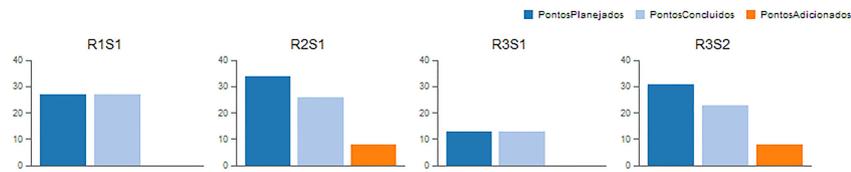


Figura 35 – Fato Tamanho: Pontos Planejados, Concluídos e Adicionados

NomeSprint	QntHorasTrabalhadas										
	Artur	Carlos	Daniel	David	Fabio	Gabriela	Leandro	Luiz	Marcos	Tales	Valessio
R1S1	14	16	14	14	16	16	16	16	20	16	32
R2S1	14	16	14	14	16	16	16	16	20	16	32
R3S1	14	16	14	14	16	16	16	16	20	16	32
R3S2	14	16	14	14	16	16	16	16	20	16	32

Figura 36 – Fato Esforço: quantidade de horas trabalhadas por Sprint

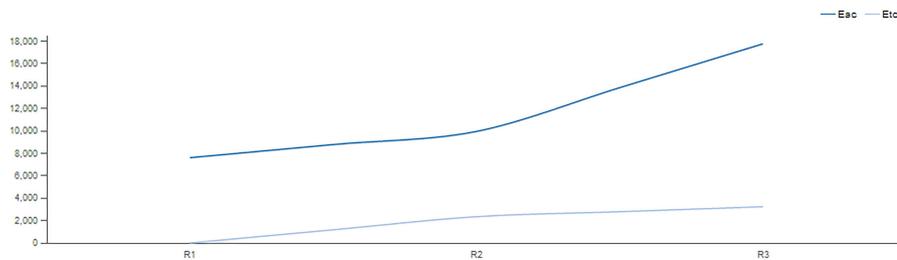


Figura 37 – Estimativas por Release