

**UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ADMINISTRAÇÃO (EA)
DEPARTAMENTO DE CIÊNCIAS ADMINISTRATIVAS (DCA)
COMISSÃO DE GRADUAÇÃO EM ADMINISTRAÇÃO (COMGRAD – ADM)**

Gabriel Hoff

**PROPOSIÇÃO DE UM SISTEMA DE MEDIDAS DE DESEMPENHO PARA
ACOMPANHAMENTO DO DESENVOLVIMENTO DE SOFTWARE EM UMA
FÁBRICA DE SOFTWARE**

**Porto Alegre
2007**

Gabriel Hoff

**PROPOSIÇÃO DE UM SISTEMA DE MEDIDAS DE DESEMPENHO PARA
ACOMPANHAMENTO DO DESENVOLVIMENTO DE SOFTWARE EM UMA
FÁBRICA DE SOFTWARE**

**Trabalho de conclusão de curso de graduação apresentado ao
Departamento de Ciências Administrativas da Universidade
Federal do Rio Grande do Sul, como requisito parcial para a
obtenção do grau de Bacharel em Administração.**

Orientadora: Prof^a. Dr^a. Ângela Freitag Brodbeck

Porto Alegre

2007

Gabriel Hoff

**PROPOSIÇÃO DE UM SISTEMA DE MEDIDAS DE DESEMPENHO PARA
ACOMPANHAMENTO DO DESENVOLVIMENTO DE SOFTWARE EM UMA
FÁBRICA DE SOFTWARE**

**Trabalho de conclusão de curso de graduação apresentado ao
Departamento de Ciências Administrativas da Universidade
Federal do Rio Grande do Sul, como requisito parcial para a
obtenção do grau de Bacharel em Administração.**

Conceito Final:

Aprovado em de de

BANCA EXAMINADORA

Prof. Dr. – UFRGS

Prof. Dr. – UFRGS

Sr.

Orientadora – Prof^ª. Dr^ª. Ângela Freitag Brodbeck – UFRGS

AGRADECIMENTOS

Agradeço a todos aqueles que de uma forma ou de outra estiveram presentes durante a minha passagem pelo curso de Administração de Empresas.

Aos meus pais pelo esforço feito para me dar uma formação com base sólida, sendo de fundamental importância para o meu ingresso em uma faculdade federal.

À minha namorada, Raquel, que esteve sempre presente durante o desenvolvimento deste trabalho e muitas outras dificuldades e felicidades, não só no curso, mas também na vida pessoal.

À minha orientadora, professora Ângela Brodbeck, pela ajuda e comunicação constante, fundamentais para a realização deste trabalho.

À empresa onde foi possível desenvolver este trabalho, em especial ao sr. Luiz Fernando Schröer.

E aos amigos, mesmo os mais distantes, que de alguma forma eu sei que posso contar.

LISTA DE ILUSTRAÇÕES

Figura 1 – Performance Pyramid	16
Quadro 1 – Comparação entre Dados de Natureza Operacional e Informacional	20
Figura 2 – Representação do modelo clássico de desenvolvimento de software	22
Quadro 2 – Exemplos de medidas de desempenho operacionais	25
Quadro 3 – Exemplos de medidas de desempenho relacionadas aos objetivos de desempenho	26
Quadro 4 – Informações sobre Participantes da Pesquisa	30
Figura 3 – Diagrama do fluxo de trabalho na Fábrica de Software	35
Quadro 5 – Respostas à questão 1: “Que etapas do processo de desenvolvimento são executadas na Fábrica de Software?”	39
Quadro 6 – Respostas à questão 2: “Estas etapas estão formalizadas? Através de que forma?”	40
Quadro 7 – Respostas à questão 3: “Quais os indicadores que são utilizados pela Fábrica de Software atualmente? De que forma estes indicadores foram levantados?”	41
Quadro 8 – Respostas à questão 4: “Como é feito o planejamento das tarefas demandadas à Fábrica de Software?”	42
Quadro 9 – Respostas à questão 5: “Na concepção da Fábrica de Software, é mais importante: tempo, qualidade ou custo?”	43
Quadro 10 – Respostas à questão 6: “Como é estimado o tempo necessário para o desenvolvimento das tarefas?”	44
Quadro 11 – Respostas à questão 7: “É feita alguma previsão de demanda de atendimento? Se sim, de que forma?”	45
Quadro 12 – Respostas à questão 8: “Na organização em que está inserida a Fábrica de Software, no que pode impactar um erro em um <i>software</i> que já foi publicado?”	46
Quadro 13 – Respostas à questão 9: “Dos indicadores abaixo, marque os 5 que você considera mais importantes para o contexto que está inserido a Fábrica de Software?”	47
Quadro 14 – Respostas à questão 10: “Existem indicadores que você julga necessário a Fábrica de Software ter, porém ainda não implantou? Que benefícios estes indicadores poderiam trazer, se adotados?”	49
Quadro 15 – Indicadores selecionados para compor o modelo	50
Figura 4 – Protótipo do <i>Dashboard</i> - Parte 1 de 2	53
Figura 5 – Protótipo do <i>Dashboard</i> - Parte 2 de 2	54

SUMÁRIO

1	INTRODUÇÃO	7
1.1	PROBLEMA	8
2	OBJETIVOS	11
2.1	OBJETIVO GERAL	11
2.2	OBJETIVOS ESPECÍFICOS	11
3	REVISÃO BIBLIOGRÁFICA	12
3.1	PLANEJAMENTO ESTRATÉGICO	12
3.2	SISTEMAS DE MEDIDAS DE DESEMPENHO	13
3.2.1	Modelos de sistemas de medidas de desempenho	15
3.2.1.1	SMART	15
3.2.1.2	PMQ	16
3.2.1.3	Balanced Scorecard	17
3.2.2	Business Intelligence	18
3.2.2.1	Data Warehouse	19
3.2.2.2	Dashboard	20
3.3	SOFTWARE	21
3.3.1	O processo de desenvolvimento de software	21
3.3.2	Fábrica de Software	23
3.3.3	Medidas de produtividade e desempenho em desenvolvimento de software	24
4	MÉTODO	27
4.1	ETAPAS DE PESQUISA	27
4.2	PROTOCOLO DE ENTREVISTA	28
4.3	EMPRESA DO ESTUDO DE CASO	29
4.4	PARTICIPANTES DO ESTUDO DE CASO	30
4.5	PROCEDIMENTOS DE COLETA E ANÁLISE DE DADOS	30
5	ESTUDO DE CASO	32
5.1	A EMPRESA	32
5.1.1	Planejamento Estratégico	33
5.2	O PROCESSO DE DESENVOLVIMENTO NA FÁBRICA DE SOFTWARE	34
5.2.1	Processo de inspeção de tarefa	

5.2.2	Processo de codificação	37
5.2.3	Processo de inspeção de software	38
5.3	ENTREVISTAS	38
5.3.1	Análise dos dados	39
5.4	CONCLUSÕES DA ANÁLISE DOS DADOS	50
6	PROTOTIPAÇÃO DO SISTEMA	52
7	CONSIDERAÇÕES FINAIS	55
7.1	PRINCIPAIS CONTRIBUIÇÕES	55
7.2	LIMITAÇÕES E PESQUISAS FUTURAS	55
	REFERÊNCIAS	57
	ANEXO A – FORMULÁRIO DE ENTREVISTA	59
	ANEXO B – MODELO DE DOCUMENTO DE DEFINIÇÃO DE TAREFA	62
	ANEXO C – MODELO DE DOCUMENTO DE CASOS DE TESTE ..	65

1 INTRODUÇÃO

Há alguns anos atrás, a preocupação das empresas em fazer medições de desempenho baseava-se apenas em saber se o negócio está indo bem em termos financeiros (ANTHONY, 1965). Todavia, a partir da década de 80, Kaplan e Norton (1983) alertaram para a importância de se medir índices que expressassem não só a saúde financeira da empresa, mas também o quanto ela estivesse preparada para o futuro.

Medidas de desempenho são instrumentos de planejamento, gerenciamento e mobilização, essenciais para saber se um determinado processo apresenta progresso, através de uma comparação do momento atual com momentos passados (TIRONI, 1993). Estas medidas são importantes em processos produtivos tanto de manufatura quanto de serviços.

Segundo Fernandes (2004), uma gestão para melhoria somente se concretiza através de medições, permitindo um entendimento do comportamento do processo. Desta forma, com o aumento constante na indústria do desenvolvimento de software, torna-se fundamental a aplicação de um sistema de medidas de desempenho em empresas deste setor. O mercado brasileiro de software e serviços ocupa a 12ª posição no mercado mundial. Em 2005, movimentou em torno de US\$ 7,41 bilhões, equivalente a 0,95% do PIB naquele ano. Conforme estudos, a média de crescimento anual deste mercado, até 2009, será superior a 11% (Associação Brasileira das Empresas de Software, 2006).

O estudo de caso deste trabalho, executado em uma Fábrica de Software, apresenta um modelo de empresa prestadora de serviço de desenvolvimento de software voltada para uma maior produtividade e maior eficiência na redução dos prazos entre a concepção e a implantação dos sistemas (FERNANDES, 2004). Desta forma, a utilização de indicadores de desempenho em uma organização deste tipo torna-se imprescindível.

As medições de software, segundo Sommerville (2003), podem ser utilizadas para coletar dados quantitativos sobre o software e seu processo. Com isso, o autor enfatiza a importância da utilização de indicadores de desempenho em *software*, visto que os valores coletados podem ser utilizados para fazer inferências sobre a qualidade do produto e de processo.

Pressmann (1995) também aponta como benefícios das medições de desempenho em software: indicar a qualidade do produto; avaliar a produtividade das pessoas que produzem o produto; formar uma linha básica para estimativas; e ajudar a justificar os pedidos de novas ferramentas ou treinamento adicional. O autor também enfatiza que estes indicadores, quando

usados em conjunto com outras atividades, como treinamento e desenvolvimento dos colaboradores, podem estimular ainda mais a busca pela qualidade no produto e aperfeiçoamento do processo.

1.1 O PROBLEMA

A empresa referenciada neste trabalho trata-se de uma Fábrica de Software. Uma fábrica de software deve ser vista como uma unidade de produção especializada nas atividades de construção de software.

A Fábrica de Software da empresa analisada já atende hoje à maioria do desenvolvimento de soluções para os produtos da matriz. Dentre os serviços prestados, encontra-se a criação e manutenção de sistemas que efetuam transações operacionais. Por meio destes sistemas, é feita a ligação entre um estabelecimento conveniado aos servidores e bancos de dados da empresa. Outro trabalho desenvolvido pela Fábrica de Software consiste na pesquisa de novas soluções tecnológicas, como, por exemplo, a busca por um tipo de banco de dados mais ágil e confiável.

Atualmente, os gestores da empresa estão começando a deparar-se com um problema. Apesar da formalidade do processo de desenvolvimento de software ter melhorado, há uma dificuldade em avaliar se a unidade está sendo produtiva ou não. A Fábrica de Software não possui indicadores formalizados e metas que avaliem o desempenho de produtividade. A falta de uma metodologia para calcular estes indicadores e traçar metas acarreta dificuldades aos gestores, quando necessitam tomar decisões estratégicas.

No decorrer deste último ano, a Fábrica de Software desenvolveu diversos sistemas de controle interno, e utilizou outros adquiridos de empresas terceirizadas. Estes sistemas visavam a um melhor acompanhamento da produção dos serviços de Tecnologia da Informação. Todavia, o resultado alcançado neste acompanhamento não foi satisfatório, pois o que se vê na empresa é uma gama de sistemas que não estão integrados, gerando assim informações inconsistentes para os gestores.

Dentre estes sistemas, podemos citar o *IBM RPM (Rational Portfolio Manager)*. Esta ferramenta tem a função de gerenciar projetos, porém para a Fábrica de Software o gerenciamento é feito no nível das tarefas, onde é feita a alocação dos recursos humanos para a execução das mesmas. Este é um dos sistemas mais utilizados no cotidiano, e oferece ao

coordenador a possibilidade de atribuir e acompanhar as tarefas aos desenvolvedores de sua equipe. Esses, por sua vez, registram no sistema todas as ações desenvolvidas nas tarefas. Desta forma, este sistema se torna basicamente um gerenciador do tempo gasto nas tarefas.

Para uso paralelo ao *IBM RPM*, foi desenvolvido o SGR (Sistema de Gerenciamento de Recursos), acessado exclusivamente pelo coordenador. Este sistema foi constituído de forma a dar uma visão gráfica dos recursos humanos. A tela principal do sistema apresenta visualmente a disposição física dos desenvolvedores. Ao lado das fotos de cada pessoa, são apresentadas suas habilidades e sua ocupação atual. Esta ocupação representa a quantidade de horas de tarefas que aguardam pelo desenvolvimento. O SGR torna-se assim um programa auxiliar para que o coordenador visualize o recurso mais apto a desenvolver uma tarefa, que será repassada através do *IBM RPM*.

Um outro sistema a ser destacado é o *Inspector*. Utilizado principalmente pelas equipes de controle de qualidade, esta ferramenta tem como objetivo principal a inspeção de tarefas. Esta inspeção é feita tanto na entrada da tarefa, visando assim uma qualidade da especificação enviada pelo analista de sistemas, quanto na saída da mesma, quando é avaliado o resultado do desenvolvimento. O *Inspector* grava estes dados referentes a inspeções, registrando as inconsistências encontradas nas fases do processo.

De funções distintas, estes três principais sistemas em um primeiro momento podem parecer completos. De certa forma, eles cumprem as funções para as quais foram designados. Porém, quando os gestores da Fábrica de Software necessitam de informações que ilustrem a rentabilidade do negócio, é necessário um grande consumo de mão-de-obra para reunir estes dados. Assim sendo, os sistemas de controle esbarram na falta de integração das informações, gerando o problema aqui mencionado.

Uma das possíveis soluções para este problema seria a integração física de informações, através de um banco de dados centralizado. Neste caso, para resolver o problema apontado por este trabalho, a Fábrica de Software deveria dispensar o uso dos sistemas atuais e centralizar os esforços em uma ferramenta multifuncional. Entretanto, esta alternativa tem um custo muito alto para a unidade, visto que seria necessário um investimento alto em mão-de-obra e treinamento em todos os níveis da empresa. A mão-de-obra seria necessária para a confecção do sistema, enquanto o treinamento seria indispensável àqueles que utilizariam diariamente o sistema, alimentando-o com dados.

Sendo assim, de modo a aproveitar as informações já existentes e dispensando mudanças de cultura na empresa, a Fábrica de Software poderia criar um conjunto de indicadores, reunindo-os em um sistema de avaliação de desempenho empresarial. Com estes

índices, seria possível aos gestores um acompanhamento mais confiável do desempenho da unidade. Desta forma, os resultados apresentados não seriam baseados em conclusões subjetivas por parte de cada pessoa que fizesse uma análise dos dados, aumentando assim a credibilidade das informações.

A compreensão detalhada do que acontece e deixar de acontecer em uma empresa deverá favorecer e tornar as tomadas de decisão dos gestores mais rápidas e alinhadas aos interesses da empresa e de seus clientes. Sendo assim, a criação de indicadores de desempenho e um sistema para acompanhamento dos mesmos parece ser uma alternativa adequada para a solução deste problema.

2 OBJETIVOS

A fim de solucionar o problema exposto acima, foram criados um objetivo geral e três objetivos específicos.

2.1 OBJETIVO GERAL

O objetivo geral deste trabalho é propor um sistema de medidas de desempenho para acompanhamento do processo de desenvolvimento de software em uma Fábrica de Software.

2.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos para este trabalho são:

- a) analisar o processo de desenvolvimento de software da Fábrica de Software;
- b) definir um conjunto de indicadores que melhor representam o desempenho da unidade de negócio;
- c) desenvolver um protótipo de sistema de mediadas de desempenho para visualização, com atualização diária, dos indicadores formulados.

3 REVISÃO BIBLIOGRÁFICA

Para a realização deste trabalho, foram abordados tópicos no que tange aos sistemas de medidas de desempenho e desenvolvimento de software. Primeiramente, é apresentado abaixo o conceito de planejamento estratégico, através do qual se origina a busca pelo aperfeiçoamento de um acompanhamento da unidade de negócio. Logo após são apresentadas algumas metodologias de medição de desempenho e, em seguida, são introduzidos conceitos de software e desenvolvimento de sistemas. O assunto *software* foi abordado pois um dos objetivos deste trabalho baseia-se na prototipação de um sistema para acompanhamento dos indicadores, que neste estudo serão levantados.

3.1 PLANEJAMENTO ESTRATÉGICO

Para Ansoff (1977), estratégia pode ser definida como "um conjunto de regras de tomada de decisão para orientação do comportamento de uma organização". É através de um planejamento estratégico que uma estratégia torna-se realidade e evolui dentro de um ambiente constantemente em mudanças em busca da vantagem competitiva sustentável para a organização.

Existem muitas maneiras de conceituar planejamento estratégico. Stoner (1982) define o planejamento estratégico como o "processo de planejamento formalizado e de longo alcance empregado para se definir e atingir os objetivos organizacionais". Este processo estabelece como a organização se relaciona com o ambiente, lida com os acontecimentos e probabilidades e resulta em um plano estratégico.

Ainda segundo Stoner (1982), o planejamento estratégico pode ser encarado como sendo "uma técnica que, através da análise do ambiente de uma organização, cria consciência das oportunidades e ameaças, dos seus pontos fortes e fracos para o cumprimento de sua missão e, através desta consciência, estabelece o propósito e direção que a organização deverá seguir para aproveitar as oportunidades e evitar riscos".

Oliveira (2001) conceitua planejamento estratégico como um processo gerencial que possibilita ao executivo estabelecer o rumo a ser seguido pela empresa, com vistas a obter um nível de otimização na relação com o seu ambiente. Meyer (1988, p.55) o define como

“processo continuado e adaptativo através do qual uma organização define (e redefine) sua missão, objetivos e metas, selecionando as estratégias e meios para atingi-los, num determinado período de tempo, através da constante interação com o ambiente externo”. Entre seus maiores objetivos figura a necessidade de definir, claramente, para cada instituição: seu negócio, sua missão, seu marco diferencial, seus recursos, suas condições de funcionamento.

Também é um instrumento para se estabelecer estratégias competitivas. Cada vez mais, os administradores o utilizam como ferramenta para auxiliar no desenvolvimento das organizações. O planejamento estratégico ajuda a formular um conceito claro e definido das organizações, permitindo a elaboração de planos e a determinação das atividades que farão com que as organizações se aproximem de seus objetivos.

3.2 SISTEMAS DE MEDIDAS DE DESEMPENHO

Medidas de desempenho são instrumentos de planejamento, gerenciamento e mobilização, essenciais para saber se um determinado processo apresenta progresso, através de uma comparação do momento atual com momentos passados (TIRONI, 1993). No ambiente atual, dinâmico e competitivo, os sistemas de medidas de desempenho passam a ser instrumentos gerenciais de grande valia para que os objetivos organizacionais possam ser alcançados com sucesso.

Sistemas de medidas de desempenho são uma parte integral do controle da administração. O sistema reflete a filosofia e cultura organizacionais e descreve o quanto o trabalho é bem feito em termos de custo, tempo e qualidade. Para serem efetivas, as medidas de desempenho necessitam refletir variações ocorridas na competitividade, além de proporcionar capacitação aos gestores para identificar o cumprimento de atividades programadas (TATIKONDA e TATIKONDA, 1998).

Até alguns anos atrás, a preocupação de muitos gestores baseava-se basicamente em medidas financeiras. Anthony (1965) argumenta que as medidas financeiras são importantes na avaliação do desempenho, pois tais medidas facilitam a comunicação dos objetivos financeiros entre as diversas áreas da empresa. Segundo ele, também possibilitam a elaboração de um resumo instantâneo sobre o desempenho global da empresa. Desta forma, estas medidas permitem que os níveis de gestão mais elevados possam tomar decisões sobre a

continuidade de estratégias adotadas pelas filiais, evitando custos adicionais para preparar relatórios financeiros que já estão prontos.

Todavia, na década de 80, Kaplan (1983) alertou para a necessidade de se melhorar o sistema de medidas de desempenho através do uso de medidas não-financeiras. Como exemplo destas medidas, podemos citar índices de produtividade, qualidade, custos de estoque, flexibilidade na manufatura e desempenho de entrega. A justificativa para essa necessidade se deu devido ao desenvolvimento de modernas técnicas de gestão, como a da qualidade total. Estas técnicas necessitam de medidas não-financeiras para que possam ser avaliadas, visto que as medidas financeiras não captam benefícios a longo prazo. Torna-se difícil identificar e relacionar atividades do passado com os gerentes atuais. Como consequência, ações corretivas são ainda mais difíceis de serem desenvolvidas. Medidas financeiras também não dão informações sobre o grau de satisfação dos clientes e aspectos relativos ao aprendizado organizacional.

Os indicadores, com relação ao controle gerencial, têm como finalidade mostrar se os resultados, no tempo, estão dentro de uma faixa aceitável de desempenho. Caso contrário, ou seja, se os resultados em um dado instante fugir dos padrões, ações corretivas devem ser tomadas para estabelecer o padrão. A periodicidade de análise de um indicador depende da complexidade do projeto, do índice ser crítico no processo decisório, dos custos e das dificuldades técnicas de sua obtenção.

Bevan e Thompson (1991) sugerem que uma organização com um sistema de gerenciamento de desempenho deveria:

- a) ter uma visão detalhada de seus objetivos, ou uma missão declarada, que comunica todos seus funcionários;
- b) ter objetivos individuais de gerenciamento de desempenho no que diz respeito a operar unidades e objetivos organizacionais mais extensos;
- c) conduzir uma revisão regular, formal de progresso para esses objetivos;
- d) usar a revisão de processos para identificar treinamento, desenvolvimento e com efeito de recompensa;
- e) avaliar a efetividade do processo total e sua contribuição para o desempenho organizacional total para permitir que mudanças e melhorias sejam feitas.

3.2.1 Modelos de sistemas de medida de desempenho

A seguir, são apresentados alguns modelos de sistemas de medidas de desempenho formulados a partir da mudança de pensamento na década de 80, onde as medidas financeiras passaram a ser apenas uma parte dos indicadores, e não a sua totalidade.

3.2.1.1 SMART

O SMART (*Strategic Measurement Analysis and Reporting Techniques*) consiste em um sistema de medição desenvolvido pela Wang Laboratories, Inc. Foi criado, assim como o BSC, devido à insatisfação com os indicadores restritamente financeiros, que traduzem uma perspectiva do passado da empresa, sem projetar índices que possam mostrar o desempenho da empresa no futuro. É um modelo baseado nos conceitos de gerenciamento da qualidade total, engenharia industrial e contabilidade por atividade.

O modelo cria condições para comunicação em duas direções. Os objetivos fluem de cima para baixo e as medidas fluem de baixo para cima. Objetivos e medidas são os elos entre a estratégia da empresa e suas atividades. O objetivo deste sistema de gerenciamento de controle é fornecer indicadores de desempenho que proporcionem o sucesso sustentado, baseando-se na necessidade de incluir medidas de performance interna e externa na avaliação de desempenho (CROSS e LYNCH, 1990).

SMART é representado por uma pirâmide (*Performance Pyramid*) e adiciona a noção de “cascateamento”: medições através de diferentes níveis hierárquicos na organização, fazendo com que as medidas dos departamentos e grupos de trabalho reflitam a visão corporativa, assim como os objetivos das unidades internas e externas de negócios. Segundo Martins (1998) o foco é a integração entre as estações de trabalho de forma que elas administrem as medidas de desempenho operacionais que sustentam a visão da corporação. Os níveis de unidade de negócios e de sistemas de operação do negócio intermediam a ligação entre os departamentos e a alta administração.



Figura 1 - Performance Pyramid.

Fonte: Cross e Lynch (1990, p. 57).

3.2.1.2 PMQ

O PMQ (*Performance Measurement Questionnaire*) é um modelo proposto por Dixon, em 1994, e tem como base 24 requisitos que compõem um modelo. É um instrumento de diagnóstico para avaliar a efetividade de um sistema de desempenho. Segundo McMann e Nanni Jr. (1994), para uma organização ter sucesso e sustentar sua competitividade, suas medidas de desempenho internas devem corresponder e dar assistência para a tarefa que ela está enfrentando.

O PMQ é formado por quatro fases. A primeira gera vários dados para que os entrevistados sejam selecionados em grupos. A segunda fase avalia as prioridades competitivas e o sistema de medição de desempenho utilizado atualmente, sendo importantíssima para a identificação das áreas de melhoria. A terceira fase é semelhante a segunda, mas seu foco está nas medidas de desempenho. E a quarta fase pergunta aos entrevistados quais as melhores medidas para se avaliar o desempenho.

Realizadas estas quatro etapas, algumas considerações podem ser feitas em quatro perspectivas: alinhamento, congruência, consenso e confusão. O alinhamento permite investigar como as ações da organização e medidas complementam a estratégia. A congruência consiste na análise detalhada de como o sistema suporta as ações e estratégias.

A análise do consenso é executada por um grupo de dados para gerenciar um nível ou grupo funcional. E o objetivo da análise de confusão é determinar a extensão do consenso, considerando cada área de melhoramento e cada medida de desempenho.

3.2.1.3 Balanced Scorecard

Originado de um estudo entre diversas empresas, motivado pela crença de que os métodos existentes para avaliação do desempenho empresarial, em geral apoiados nos indicadores contábeis e financeiros, estavam se tornando obsoletos, o Balanced Scorecard evoluiu para um uso muito mais amplo do que apenas avaliação de desempenho.

As discussões em torno do mesmo resultaram em uma ampliação do conceito inicial de Scorecard, resultando no Balanced Scorecard (BSC), com quatro perspectivas distintas:

- a) financeira;
- b) do cliente;
- c) de processos internos;
- d) de inovação e aprendizado.

O nome reflete o equilíbrio entre os objetivos de curto e longo prazos, entre as medidas financeiras e não financeiras, entre indicadores de tendências e ocorrências e entre as perceptivas interna e externa de desempenho.

O BSC complementa as medidas financeiras do desempenho passado com medidas dos vetores que impulsionam o desempenho futuro. Os objetivos e medidas focalizam o desempenho nas quatro perspectivas já mencionadas. Estas perspectivas formam a base da estrutura do Balanced Scorecard.

Os princípios que permitem a integração entre as medidas de resultados com os vetores de desempenho de um Balanced Scorecard são:

- a) relação de causa e efeito;
- b) resultados e vetores de desempenho;
- c) relação com os fatores financeiros.

Segundo Kaplan e Norton (2000), “a estratégia é um conjunto de hipóteses sobre causas e efeitos. O sistema de medição deve tornar explícitas as relações (hipóteses) entre os

objetivos (e as medidas) nas várias perspectivas, para que elas possam ser gerenciadas e validadas”. Devem estar explícitas as seqüências de hipóteses de causas e efeitos entre as medidas de resultados e os vetores de desempenho desses resultados, nas diversas perspectivas que compõem o scorecard. Todas as medidas identificadas devem estar relacionadas a uma causa que conduz ao efeito desejado.

Ainda segundo os autores, as medidas de resultado refletem as metas comuns de muitas estratégias, bem como estruturas semelhantes entre setores e empresas. Os vetores de desempenho, por sua vez, são indicadores de tendências geralmente específicos para uma determinada unidade de negócios.

Kaplan e Norton (1997) ainda destacam que “em última análise, as relações causais de todas as medidas incorporadas ao scorecard devem estar vinculadas a objetivos financeiros”.

Comentam ainda que sua utilização hoje se dá como um sistema de gestão estratégica para a administração da estratégia a longo prazo. As empresas adotam a filosofia do scorecard para viabilizar processos críticos:

- a) esclarecer e traduzir a visão e a estratégia;
- b) comunicar e associar objetivos e medidas estratégicas;
- c) planejar, estabelecer metas e alinhar iniciativas estratégicas;
- d) melhorar o feedback e o aprendizado estratégico.

3.2.2 Business Intelligence

O conceito de *Business Intelligence* (BI) é relativamente novo. A caracterização do BI que hoje é conhecido começa na década de 70, quando alguns produtos de foram disponibilizados para analistas de negócio. Estes produtos, porém, exigiam intensa e exaustiva programação, a informação não era flexível e rápida. Aliado a isto, o alto custo de implantação tornou-se uma barreira para a evolução do conceito.

No entanto, a revolução tecnológica, com implantação de interfaces gráficas nos computadores, e o aumento da complexidade dos negócios fizeram com que surgissem produtos específicos a este ramo. Estes produtos acabaram proporcionando uma maior flexibilidade e rapidez na disponibilização das informações.

O BI engloba diversas fontes de dados, transformando-as em depósitos estruturados de informações (*Data Warehouse*), independentes de sua origem (BARBIERI, 2001). Sendo

assim, pode ser considerado que um BI contém diversos sistemas de medidas de desempenho, cada qual para um devido fim. Estes sistemas, por sua vez, são voltados para a inteligência do negócio, pois facilitam a descoberta de relações entre estes dados extraídos (HAN e KAMBER, 2001).

Para Elmasri e Navathe (2005) as ferramentas de BI devem oferecer interfaces que facilitem ao usuário o entendimento das relações entre os dados (descritivo), a fim, por exemplo, de prover melhores informações para a tomada de decisão. Assim, o *Business Intelligence* tem uma ligação com a vertente tecnológica da gestão do conhecimento. Além disso, ao prover informações para tomada de decisão, o BI pode ajudar tanto nos processos descritivos quanto nos normativos de inteligência empresarial.

3.2.2.1 Data Warehouse

Um dos componentes do *Business Intelligence* é o *Data Warehouse*. Este conceito consiste em um banco de dados separado para fins analíticos de uma empresa. Com isto, torna-se perfeitamente compreensível o porquê do *Data Warehouse* ser tão importante em BI.

A idéia de criar um banco de dados consolidado surgiu pois os primeiros bancos de dados foram desenvolvidos com ênfase em um processamento de dados operacionais. Uma vez compreendida a diferença entre dados analíticos e operacionais, as empresas resolveram fazer esta separação, explicam Gray e Watson (1998). É inevitável a perda de produtividade e confiabilidade quando os dados são retirados de diversas fontes em sistemas desintegrados. Assim, o conceito de *Data Warehouse* propicia integração e consolidação da informação.

O *Data Warehouse* extrai, de uma maneira eficiente, informações valiosas dos bancos de dados operacionais que estão espalhados por toda a empresa e que, certamente, contêm dados duplicados ou parcialmente duplicados. O grande objetivo do *Business Intelligence* está em definir regras e técnicas para a formatação adequada dessas informações, transformando-as em depósitos estruturados de dados. O conceito está, na sua essência, relacionado com formas alternativas de tratamento de informações (BARBIERI, 2001).

Os dados operacionais e informacionais constituem fontes importantes para o estabelecimento dos conceitos de *Business Intelligence*, porém com características distintas, como apresentado no Quadro 1 abaixo:

Características	Dados operacionais	Dados informacionais
Conteúdo	Valores correntes	Valores sumariados, calculados e integrados de várias fontes
Organização dos dados	Por aplicação/sistema de informação	Por assuntos/negócios
Natureza dos dados	Dinâmica	Estática até a sincronização dos dados
Formato das estruturas	Relacional, próprio para computação transacional	Dimensional, simplificado, próprio para atividades analíticas
Atualização dos dados	Atualização campo a campo	Acesso, sem atualização nos dados
Uso	Altamente estruturado, processamento repetitivo	Desestruturado, com processamento analítico/heurístico
Tempo de resposta	Otimizado para 2 a 3 segundos	Análises mais complexas, com tempos de respostas maiores

Quadro 1 – Comparação entre Dados de Natureza Operacional e Informacional

Fonte: Barbieri, 2001

Gray e Watson (1998) destacam como características destes bancos:

- a) são orientados a determinado assunto;
- b) são integrados;
- c) não são violáveis, ou seja, seus dados não são alterados, só incluídos;
- d) representam um histórico;
- e) são condensados;
- f) representam um longo período de tempo (de 5 a 10 anos);
- g) não são normalizados, ou seja, podem ter dados redundantes;
- h) não são baratos e têm implantação demorada; e
- i) são grandes e continuam crescendo ao longo do tempo.

3.2.2.2 Dashboard

Para a representação dos dados consolidados em um *Data Warehouse*, explicado no item anterior, uma das formas mais utilizadas tem sido o *Dashboard*. *Dashboard* é uma ferramenta que fornece informações imediatas sobre o desempenho dos negócios em toda a empresa. Tipicamente, são gerados para os gerentes e executivos que precisam de uma visão

geral do negócio e consideram primordial dispor de uma visualização intuitiva e oportuna dos dados estratégicos, financeiros e operacionais.

O *Dashboard* também é conhecido Painel de Controle, pois dá uma visão gráfica dos gestores da situação dos indicadores da unidade. Esta tecnologia tem sido considerada por muitos CEO's como tendência para o futuro (BUSINESS WEEK, 2006). Isto se deve ao fato que a análise dos indicadores torna-se muito mais ágil, uma vez que os dados são agrupados e apresentados com clareza.

Todavia, estes executivos também citam o perigo de utilizar apenas essa ferramenta como controle de uma unidade. Atualmente, ainda não há como controlar certos aspectos de uma linha produtiva através da tecnologia. Desta forma, estes CEO's sugerem que em uma equipe tenham pessoas que acompanhem estes indicadores, e outras se preocupem com questões que necessitam de participação efetiva (BUSINESS WEEK, 2006).

3.3 SOFTWARE

Por tratar-se de um trabalho onde foi desenvolvido um protótipo de software, torna-se fundamental elucidar aqui o conceito de software, segundo Pressman (1995):

“(1) Instruções (programas de computador) que, quando executadas, produzem a função e o desempenho desejados; (2) Estruturas de dados que possibilitam que os programas manipulem adequadamente a informação; (3) Documentos que descrevem a operação e o uso dos programas.”

No próximo tópico, é abordado o processo de desenvolvimento de software, bem como suas etapas.

3.3.1 O Processo de Desenvolvimento de Software

O processo de desenvolvimento de software, ou processo de software, corresponde ao conjunto de atividades relacionadas que são desempenhadas pelos desenvolvedores, que levam à produção de um produto de software (SOMMERVILLE, 2003). Como forma de analisar e amadurecer tal processo, é sugerida uma descrição do processo, a qual consiste de

um modelo de processo de software. A descrição formal de um processo de software é a atividade que permite que o mesmo seja analisado, compreendido e automatizado.

Existem diversos modelos de etapas de desenvolvimento de software (ciclo de vida). Para explicá-lo, neste trabalho, será apresentado o modelo clássico conhecido como *Waterfall* (PRESSMAN, 1995).

Este modelo é o mais didático, e apresenta as etapas de forma sequencial e bastante lógica, porém não é recomendado para elucidar a iteração entre os estágios. As etapas deste modelo são as seguintes:

- a) análise e engenharia de sistema;
- b) análise de requisitos de software;
- c) projeto;
- d) codificação (desenvolvimento);
- e) testes;
- f) manutenção.

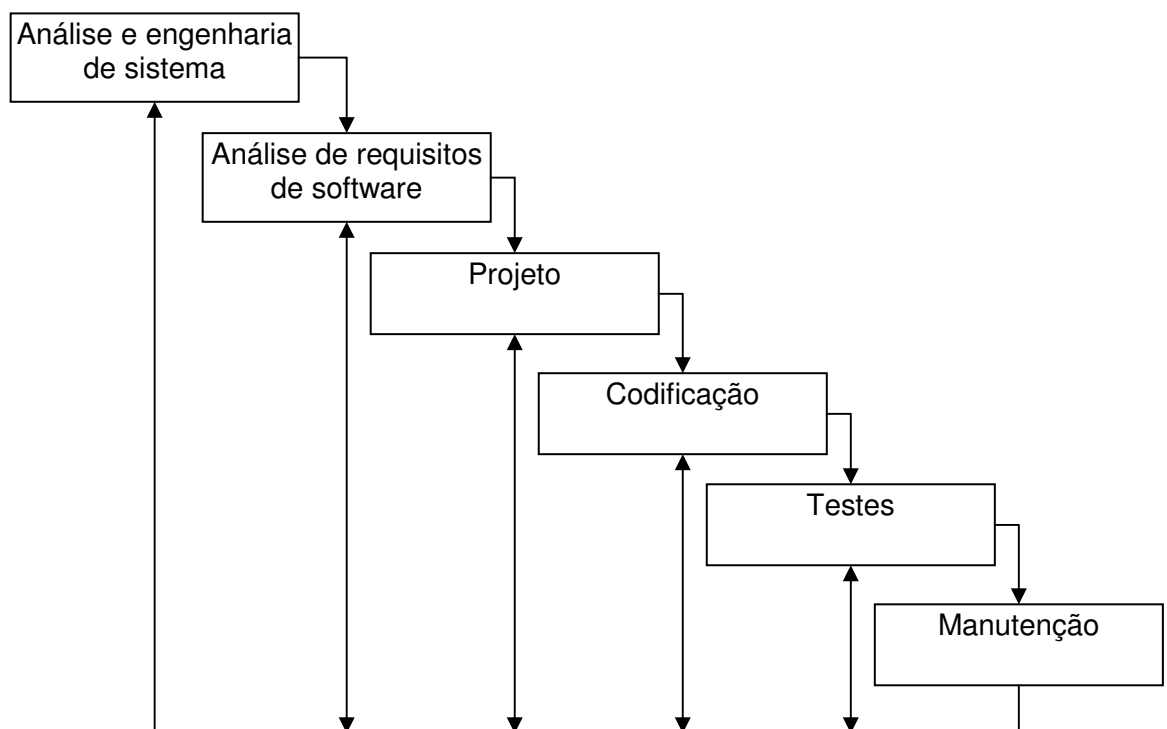


Figura 2 – Representação do modelo clássico de desenvolvimento de software

Fonte: elaborado pelo autor

Um dos problemas analisados com este modelo é o de que os requisitos devem ser completamente especificados e congelados antes do projeto (PRESSMAN, 1995). Para driblar este problema, há uma adaptação do modelo clássico *Waterfall*, incluindo uma fase de prototipação antes do desenvolvimento, de fato, do sistema. Os protótipos podem ser, por exemplo, dos seguintes tipos:

- a) protótipo em papel: é a forma mais rudimentar, utilizado normalmente para representar a interface homem-máquina, através de seqüência de figuras correspondentes às telas;
- b) protótipo executável: é obtido através da implementação, simplificada ou não, de um subconjunto de funções do sistema em um computador que não é necessariamente a máquina onde será instalado o sistema final;
- c) programa existente: consiste no uso de um programa existente que permita executar parte ou a totalidade das funções desejadas; é o caso, por exemplo, de um sistema cuja ampliação ou modificação justifica um novo desenvolvimento.

Do ponto de vista do protótipo executável, este modelo traz, aparentemente, muito mais vantagens do que o clássico, devido às correções que podem ser feitas até a obtenção do sistema final. No entanto, a sua aplicação ao processo de desenvolvimento exigem, da gerência, o estabelecimento de regras bastante precisas, para que um protótipo não acabe sendo o produto definitivo. Neste trabalho, foi desenvolvido um protótipo de um sistema para acompanhamento dos indicadores de desempenho de uma fábrica de software. O protótipo desenvolvido é um executável, que apresenta algumas das características propostas para o sistema.

3.3.2 Fábrica de Software

Uma Fábrica de Software deve ser vista como uma unidade de produção especializada nas atividades de construção de software. Uma unidade de produção caracteriza-se por modelo semelhante a um processo industrial, visando principalmente a qualidade e a produtividade. É na proposta de trabalho de uma Fábrica de Software que a tecnologia da informação (TI) transforma-se num processo de produção efetivo (FERNANDES, 2004).

As empresas especializadas em software, por questão de competitividade, tem maior propensão à utilização de técnicas e ferramentas mais modernas. Dessa forma apresentam maior produtividade e maior eficiência na redução dos prazos entre a concepção e a implantação dos sistemas. Hoje a relação “cliente-fornecedor” de muitas destas empresas, além de estarem embasadas nestes modelos, encontram-se reguladas através de acordos de níveis de serviço e indicadores de projetos e processos que permitem uma transparência muito maior às partes envolvidas (COSTA, 2003).

Fábrica de Software é também designada como um conjunto de recursos (humanos e materiais), processos e metodologias estruturados de forma semelhante às indústrias tradicionais, utilizando as melhores práticas criadas para o processo de desenvolvimento de software. Em sua operação, são usados indicadores de qualidade e produtividade em cada etapa do ciclo de desenvolvimento de software. Também tem como objetivo maximizar a reutilização de componentes anteriormente desenvolvidos (componentização) (FERNANDES, 2004).

3.3.3 Medidas de produtividade e desempenho em desenvolvimento de software

A preocupação com a qualidade no desenvolvimento de software tem tido um aumento constante. Segundo Fernandes (2004), uma gestão para melhoria somente se concretiza através de medições, permitindo um entendimento do comportamento do processo. Para Sommerville (2003), a medição de software se ocupa em obter um valor numérico para alguns atributos de um produto ou de um processo de software. Com isto, a empresa fica possibilitada de estabelecer metas de melhorias, pois saberá ao certo o que realmente ocorre em seu ambiente.

As medições de desempenho de software dividem-se em três grupos, com propósitos diferentes (FERNANDES, 2004):

- a) medições estratégicas: têm como objetivo o *benchmarking* da organização de software em relação a padrões internacionais e em relação a competidores, além de possibilitar a implementação de projetos de melhoria contínua e estudos econômicos acerca do acervo de software;

- b) medidas táticas: objetivam a possibilidade da gestão do ambiente de software em sua totalidade, em termos de projetos, processos e produtos. Verificam tendências do ambiente em termo de qualidade, custo, produtividade, ciclos de tempo, além de analisar o impacto de mudanças de processo e tecnologia.
- c) medidas operacionais: ocorrem no nível de cada projeto ou produto de software (em operação e pelos usuários). As medições desse nível, uma vez coletadas e agrupadas, fornecem subsídios para a avaliação de desempenho do nível superior (tático).

Neste trabalho, foram focadas medidas de desempenho operacionais, visto que estas são o primeiro passo para a maturidade da empresa, possibilitando futuramente uma medição de desempenho nos grupos estratégico e tático. No caso da Fábrica de Software analisada, o nível da análise foi feito tanto por projeto quanto por tecnologia empregada, uma vez que esta unidade desenvolve software em diferentes linguagens.

Ainda segundo Fernandes (2004) são exemplos de medições operacionais:

Tipos de medições	Medições
Medições realizadas para o desenvolvimento	Estimativa de tamanho de software
	Estimativa de prazo
	Estimativa de esforço
	Estimativa de recursos
	Estimativa de custo
	Estimativa de qualidade
	Estimativa de confiabilidade
	Estimativa de produtividade
	Estimativa de defeitos
	Estimativa de retrabalho
Medições realizadas para a gestão do produto	Número de defeitos pós- <i>release</i>
Medições no desenvolvimento	Desvio das estimativas
	Efetividade das revisões
	Progresso na remoção de defeitos
	Progresso físico
	Custo realizado
	Composição de tipos de defeitos
	Densidade de defeitos
	Complexidade do software
	Origem de defeitos
Medições realizadas para a gestão do produto	Falhas esperadas para o software
	Intensidade atual das falhas
	Estimativa do número de defeitos pós- <i>release</i>
	Estimativas de atendimento
	Custo de produtividade do atendimento

Quadro 2 – Exemplos de medidas de desempenho operacionais

Fonte: Fernandes, 2004

Assim como uma manufatura tradicional, o processo de desenvolvimento de software também pode ser orientado por objetivos de desempenho, conforme exemplificado no quadro abaixo:

Objetivos de desempenho	Medições
Qualidade do produto de software	Densidade de defeitos (número de defeitos no software sobre o tamanho em horas do software)
Qualidade do processo de software	Eficiência de remoção de defeitos (número de defeitos reportados pelos clientes na operação do software dividido pelo somatório do número de defeitos encontrados e removidos no teste de pré-release, mais o número de defeitos reportados pelos clientes)
Confiabilidade de entrega	% de projetos ou solicitações de serviços (manutenção ou programa) entregues no prazo acordado com o cliente, conforme um período de tempo determinado
Tempo	% de redução do tempo de entrega de produtos no período
	Tempo de absorção de um novo método ou linguagem de programação
	Produtividade do desenvolvimento/manutenção
Flexibilidade	Número de ambientes de desenvolvimento
	Número de ciclos de vida utilizados
	Número de técnicas de análise utilizadas
	Número de linguagens de programação
Inovação	Número de habilidades do <i>staff</i>
	% de novos ambientes de software incorporados no período
Custo	% de novos processos incorporados
	Custo por linha de código
	Custo por hora de desenvolvimento
	% de redução de custo do processo no período
	% de projetos e serviços dentro do orçamento ou abaixo

Quadro 3 – Exemplos de medidas de desempenho relacionadas aos objetivos de desempenho

Fonte: Fernandes, 2004

4 MÉTODO

Neste capítulo será descrito o método que foi utilizado neste trabalho, de forma a alcançar os objetivos. Como forma de validação do modelo apresentado, foi feito um estudo de caso na empresa referenciada no item problemática.

O método do estudo de caso é propício para pesquisas que “investigam um fenômeno contemporâneo dentro de um contexto da vida real, quando a fronteira entre o fenômeno e o contexto não é claramente evidente e onde múltiplas fontes de evidência são utilizadas” (YIN, 2001). A partir deste método consegue-se compreender e distinguir o método do estudo de caso de outras estratégias de pesquisa como o método histórico e a entrevista em profundidade, o método experimental e o *survey*.

A ênfase deste método está na compreensão, fundamentada basicamente no conhecimento tácito (STAKE, 1995). Segundo o autor, o estudo de caso tem uma forte ligação com intencionalidade, o que não ocorre quando o objetivo é meramente explanação, baseada no conhecimento proposicional. Assim, quando o centro de um estudo for a explanação, ou a busca de um conhecimento proposicional, o estudo de caso pode não ser vantajoso. Por outro lado, quando o objetivo do estudo de caso é a compreensão e ampliação da experiência, esta desvantagem desaparece.

4.1 ETAPAS DA PESQUISA

As etapas desta pesquisa de estudo de caso foram:

- a) levantamento bibliográfico sobre o assunto, onde foram explorados assuntos base deste trabalho, tais como: estratégia, sistemas de medidas de desempenho e fábrica de software. O objetivo da pesquisa refere-se ao produto final desta pesquisa ser um protótipo de sistema de acompanhamento dos indicadores que serão levantados;
- b) geração de um roteiro para as entrevistas individuais, com a formulação de perguntas que conduziram as mesmas, de forma a explorar as necessidades da

empresa em relação aos indicadores que são consideradas importantes pelos gestores;

- c) estudo de caso, onde foram levantadas as características da empresa, bem como da Fábrica de Software, unidade de negócios onde será desenvolvido o trabalho. Após isto, é relatado o processo de desenvolvimento de *software*, focando na parte operacional deste. Este foco se deve pelo motivo que os indicadores a levantados são essencialmente operacionais; e
- d) prototipação, onde foi obtida uma prévia do sistema de indicadores, através da implementação de um subconjunto de funções, apresentando os indicadores levantados no estudo de caso. O desenvolvimento deste protótipo foi feito em linguagem de programação *Cold Fusion*, da empresa de Tecnologia da Informação *Macromedia*. Este protótipo foi desenvolvido no ambiente da unidade de negócio aqui estudada e será mostrado, em imagens, neste trabalho, e também ao vivo na apresentação do trabalho à banca examinadora. Para confecção deste protótipo, será utilizado o conceito de *Dashboard* que, conforme será visto na revisão bibliográfica, é uma forma de apresentação de dados no BI (*Business Intelligence*, ou inteligência do negócio).

4.2 PROTOCOLO DE ENTREVISTA

Para levantamento dos dados nas reuniões realizadas, foi elaborado um formulário para documentar os assuntos tratados (anexo A), buscando identificar a necessidade de determinados indicadores. As perguntas foram baseadas, primeiramente, na revisão bibliográfica e no relato do processo de desenvolvido de *software* da Fábrica de Software, que será abordado no capítulo 5 (Estudo de Caso).

O roteiro de entrevista, detalhado no anexo A, aborda os seguintes questionamentos:

- a) primeiramente, o entendimento do que a Fábrica de Software já possui de artifícios hoje (**planejamento estratégico**), mesmo que não sejam aplicados, visando compreender o ambiente a ser estudado é o primeiro grande passo para fazer um plano de execução (WOILER e MATHIAS, 1996);

- b) a seguir, com o ambiente compreendido, o entendimento do que é importante para a Fábrica de Software, visando entender melhor o **processo de desenvolvimento de software** da empresa.
- c) por último, a exploração da percepção do entrevistado quanto aos **indicadores** que não são úteis e aos que são úteis para uma empresa com a estrutura da Fábrica de Software.

Além das informações obtidas por entrevistas, foi analisada toda a documentação que a Fábrica de Software mantém sobre o processo de desenvolvimento de software. Desta forma, buscou-se ter mais subsídios para a realização do estudo de caso e determinação da relevância dos indicadores averiguados a partir da revisão conceitual, bem como novos indicadores que porventura aparecerão.

4.3 EMPRESA DO ESTUDO DE CASO

Este estudo de caso foi conduzido em uma unidade de negócios de uma empresa prestadora de serviços que atua nos ramos de gestão de frotas e benefícios. A unidade estudada é responsável pelo desenvolvimento de software de toda empresa. A matriz foi fundada em 1999, visando desde então fornecer soluções com tecnologia avançada aos seus clientes. A empresa possui como uma bandeira de cartões, que são vendidos aos seus clientes (em geral empresas de médio e grande porte), que fornecem este serviço aos seus funcionários. Estes então podem utilizar o cartão na rede conveniada (principalmente supermercados e postos de gasolina), como forma de benefício.

Há cerca de um ano atrás, a empresa inaugurou uma nova unidade física, localizada na Zona Norte de Porto Alegre. Esta filial, denominada Fábrica de Software, foi constituída com o objetivo de centralizar o desenvolvimento de aplicativos para todos os setores da empresa, separando assim as responsabilidades da organização: a tecnologia passa a ser administrada pela nova unidade, enquanto que o negócio (cartões de benefício) fica a cargo das demais áreas da empresa.

Maiores detalhes sobre a empresa, encontram-se descritos no capítulo 5.

4.4 PARTICIPANTES DO ESTUDO DE CASO

As entrevistas do estudo de caso foram feitas com quatro pessoas que ocupam cargos administrativos na empresa e que trabalham diretamente no processo de desenvolvimento da Fábrica de Software, conforme apresentado no quadro abaixo:

Ocupação	Escolaridade	Experiência na ocupação
Gerente da unidade	Bacharel em Análise de Sistemas (Unisinos); Pós-graduado em Administração da Tecnologia da Informação (Unisinos).	6 anos
Consultor de processos	Bacharel em Ciências da Computação (UFRGS).	7 anos
Coordenador da equipe de desenvolvimento	Bacharel em Ciências da Computação (PUC-RS).	4 anos
Coordenador da equipe de testes	Bacharel em Sistemas de Informação (Centro Universitário Franciscano - Santa Maria); Mestre em Computação Aplicada (Unisinos).	2 anos

Quadro 4 – Informações sobre Participantes da Pesquisa

Fonte: elaborado pelo autor

4.5 PROCEDIMENTOS DE COLETA E ANÁLISE DE DADOS

Para averiguação dos indicadores a serem utilizados, foram feitas entrevistas individuais com o gerente, coordenadores da unidade de negócio (Fábrica de Software) e um consultor de processos. De forma geral, as entrevistas são uma fonte essencial de evidências para o estudo de Caso. O tipo de entrevista utilizada foi a entrevista focada, onde o respondente é entrevistado por um curto período de tempo e pode assumir um caráter aberto-fechado - onde o investigador pode solicitar aos respondentes-chaves a apresentação de fatos e de suas opiniões a eles relacionados - ou se tornar conversacional, mas o investigador deve preferencialmente seguir as perguntas estabelecidas no protocolo da pesquisa (YIN, 2001).

A duração das entrevistas foi de aproximadamente uma hora. Além das questões abertas do roteiro de entrevistas, foram apresentados aos respondentes os indicadores pré-

estabelecidos através das referências bibliográficas. Estes indicadores foram discutidos e selecionados, conforme a realidade observada na empresa pelos entrevistados.

Um ponto importante no uso dos indicadores se deve ao fato que a empresa deverá ter os dados disponíveis para serem extraídos e calculados pelos indicadores, visto ser esta uma necessidade para o desenvolvimento do protótipo.

A Fábrica de Software também já utiliza alguns indicadores, que são divulgados mensalmente em uma reunião gerencial. Estes indicadores, todavia, não são formalizados, sendo necessário um levantamento manual dos dados a cada mês. A pesquisa discute a utilidade destes indicadores, e a necessidade da aplicação dos mesmos no novo sistema, visto que estes indicadores antigos não são extraídos em tempo real.

Ao fim do levantamento dos dados, haverá uma análise de evidências no estudo de caso, com o objetivo de verificar quais os indicadores de desempenho necessários para a Fábrica de Software. Segundo Yin (2001), “o objetivo final da análise é o de tratar as evidências de forma adequada para se obter conclusões analíticas convincentes e eliminar interpretações alternativas”.

Como estratégia para esta análise, Yin (2001) sugere como uma das opções o desenvolvimento da descrição do caso. Em vista disso, será relatado neste estudo o processo de desenvolvimento da Fábrica de Software, para auxiliar na escolha dos indicadores.

Desta forma, o fechamento do estudo de caso se dá com o desenvolvimento do protótipo do sistema de medição de desempenho. As respostas encontradas guiaram o autor a implementar o *Dashboard* na empresa estudada, podendo ser utilizado na unidade de negócios na qual foi aplicado este trabalho.

5 ESTUDO DE CASO

Neste capítulo é relatado o estudo de caso realizado em uma Fábrica de Software. Primeiramente, é descrita a organização onde esta fábrica insere-se, para posteriormente descrever o processo na unidade.

Uma vez identificada a organização, é descrita a condução das entrevistas para que pudesse ser feita a análise final. Como resultado, este capítulo mostra o modelo de indicadores que se propõe para utilização na Fábrica de Software.

5.1 A EMPRESA

A empresa na qual foi conduzido o estudo de caso deste trabalho foi fundada em 1999, por um sólido e experiente grupo empresarial. Tendo como missão desenvolver sistemas e tecnologia para gestão de convênios com uma marca associada, a empresa foi pioneira no uso da internet como meio de disponibilizar aos seus clientes ferramentas para a auto-gestão, onde o próprio cliente altera limites, solicita cartões, emite extratos, cancela, dentre outras funcionalidades.

O maior diferencial para o sucesso do empreendimento foi o desenvolvimento da plataforma tecnológica, com operação totalmente *on-line*, permitindo já no ano de 2000 a conquista de importantes clientes que validaram os produtos junto ao mercado nacional. Casas Bahia, Danone, TAM e o grupo Sílvio Santos, entre outros.

Com o seu foco voltado para o cliente e atendendo à demanda de desenvolvimento, a empresa aqui estudada lançou, em 2001, um sistema para controle e acompanhamento de tratamento médico prescrito e compra de medicamentos, logo implantado em grandes empresas como Gerdau e Eaton do Brasil.

Com uma rede de postos credenciados em nível nacional, um sistema inteligente, flexível e que provoca redução nas despesas para clientes que possuem frotas de veículos, a empresa lançou em 2002 um cartão para abastecimento de frotas de seus clientes. Este produto representa, atualmente, 60% do faturamento da organização.

Em agosto de 2005 esta empresa inaugurou uma fábrica de software em Porto Alegre, criada para atender à demanda de todo o grupo de empresas da qual faz parte, com o objetivo

de centralizar as demandas de tecnologia para as mais diversas áreas. Esta unidade será focada neste trabalho, visto que o objetivo principal será a criação de um sistema para acompanhamento dos indicadores de desempenho da fábrica de software.

5.1.1 Planejamento Estratégico

O planejamento estratégico da empresa apresenta os seguintes componentes:

- a) missão: desenvolver soluções e tecnologia para gestão de convênios, com profissionais criativos e motivados, aliando qualidade na prestação dos serviços a preços justos, garantindo a plena satisfação de clientes e acionistas.
- b) negócio: integrar informação e conhecimento em gestão de convênios.
- c) visão: liderar o mercado de gestão de convênios, destacando-se pela competitividade e comprometimento com resultados, e ser reconhecida pela sua solidez e confiabilidade.
- d) valores:
 - a. agilidade;
 - b. determinação;
 - c. humildade;
 - d. trabalho em equipe;
 - e. inovação;
 - f. parceria;
 - g. otimismo;
 - h. transparência;
 - i. comunicação;
 - j. honestidade;
 - k. reconhecimento;
 - l. persistência.
- e) objetivos:
 - a. atender às necessidades dos clientes de forma ágil e inteligente;
 - b. respeitar a individualidade, formar as pessoas e valorizar o trabalho em equipe;

- c. compartilhar de forma clara a responsabilidade pelo processo das metas do negócio;
- d. buscar o lucro como medida para perenização da empresa;
- e. desenvolver e comprometer a equipe com processos produtivos, flexíveis, eficientes e eficazes;
- f. incentivar as práticas de cidadania e responsabilidade social;
- g. faturar R\$ 1 bilhão em 2007.

5.2 O PROCESSO DE DESENVOLVIMENTO DE SOFTWARE NA FÁBRICA DE SOFTWARE

Na Fábrica de Software, desde o início de suas operações, o processo de desenvolvimento de software vem sendo evoluído constantemente. Como já descrito no capítulo de Referências Bibliográficas, uma Fábrica de Software é responsável apenas pela etapa de construção do software. Neste capítulo, é descrito, com base no modelo *Waterfall*, as fases de responsabilidade da Fábrica de Software. Este modelo possui, conforme já foi citado anteriormente, as seguintes etapas:

- a) análise e engenharia de sistema;
- b) análise de requisitos de software;
- c) projeto;
- d) codificação (desenvolvimento);
- e) testes;
- f) manutenção.

Foram focadas apenas as etapas de codificação e testes, que são as de responsabilidade da unidade estudada. A premissa para a execução destas etapas dentro da Fábrica de Software é a entrega, pelas equipes de análise de sistemas e análise de negócios, dos documentos de definição de tarefa (análise do caso de uso¹ e casos de teste).

Quando da necessidade de implementação de uma nova funcionalidade, ou alteração de um programa existente, estas equipes de análise solicitam a execução de uma tarefa ou

¹ Especificação do comportamento do sistema ou parte(s) dele e descrevem a funcionalidade do sistema desempenhada pelos atores.

grupo de tarefas em um projeto. É através destes documentos formulados que fica formalizada a necessidade averiguada nas etapas anteriores à codificação, no processo de desenvolvimento de software.

Cabe então, à Fábrica de Software, a interpretação deste documento para posterior codificação. Os anexos B e C apresentam, respectivamente, a estrutura dos documentos de definição de tarefa e casos de teste. Nestes documentos, os analistas preenchem os requisitos necessários para a execução de uma tarefa de determinado projeto, deixando desta forma todas as alterações e implementações nos sistemas devidamente documentadas.

O processo de desenvolvimento na Fábrica de Software é dividido, basicamente, em três processos:

- a) inspeção de tarefa;
- b) desenvolvimento (codificação);
- c) inspeção de software.

Em alguns casos, quando a codificação apresenta problemas na inspeção de software, há a etapa de correção de software. Depois de completado o fluxo dentro da Fábrica de Software, a tarefa é enviada a um setor externo de homologação. Desta forma, o fluxo de trabalho da Fábrica de Software pode ser representado da seguinte forma:

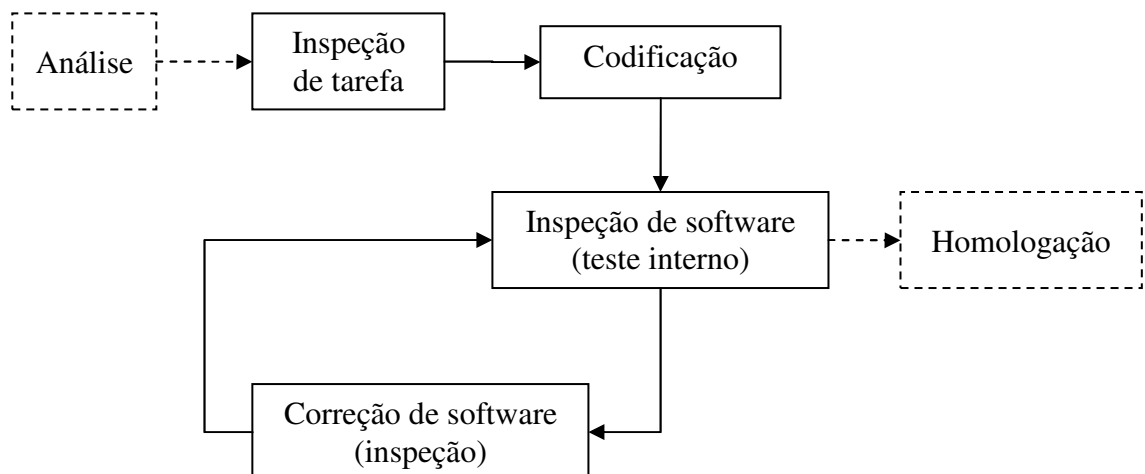


Figura 3 – Diagrama do fluxo de trabalho na Fábrica de Software

Fonte: elaborada pelo autor

5.2.1 Processo de inspeção de tarefa

Antes de codificar o que foi requisitado, há a etapa de inspeção de tarefa. Esta etapa baseia-se na avaliação dos documentos enviados pela análise, verificando se os mesmos encontram-se no padrão exigido pela diretoria. A avaliação é feita por um desenvolvedor habilitado, onde o mesmo segue o *checklist* abaixo:

- a) verificar se os dados gerais de identificação da tarefa estão preenchidos corretamente. Dados obrigatórios: Título, analista de sistemas responsável, aplicação, menu de acesso, número da tarefa (projeto/tarefa);
- b) verificar se a descrição do objetivo está preenchida e se está claro o que deve ser atingido com a execução da tarefa;
- c) verificar se a lista de programas (fontes) está preenchida. Verificar se os programas listados estão nos caminhos indicados, disponíveis e com permissão de acesso pelo desenvolvedor;
- d) verificar se as *stored procedures*² referenciadas na lógica de processamento estão listadas corretamente;
- e) verificar a existência das *stored procedures* listadas;
- f) verificar a permissão de acesso das *stored procedures* listadas;
- g) verificar se as *triggers*³ referenciadas na lógica de processamento estão listadas corretamente;
- h) verificar a existência das *triggers* listadas;
- i) verificar a permissão de acesso das *triggers* listadas;
- j) verificar se os usuários da aplicação e do banco de dados estão informados;
- k) verificar se o item Dados/Métodos está preenchido indicando o nome das tabelas, arquivos ou demais origens de dados, e indicando o tipo de operação a ser realizada;
- l) verificar se as entradas estão definidas e se estas estão devidamente relacionadas com o item Dados, ou seja, as entradas devem pertencer às tabelas, arquivos, dentre outros, descritos no item Dados como entrada;

² Conjunto de comandos, ao qual é atribuído um nome. Este conjunto fica armazenado no Banco de Dados e pode ser chamado a qualquer momento tanto pelo SGBD (sistema Gerenciador de Banco de Dados) quanto por um sistema que faz interface com o mesmo

³ Tipo especial de procedimento armazenado, que é executado sempre que há uma tentativa de modificar os dados de uma tabela que é protegida por ele.

- m) ler a lógica de processamento (funcionamento) e verificar se é possível identificar claramente no processamento que deverá ser feito e a utilização das entradas que foram descritas;
- n) verificar a existência de cálculos e se estes cálculos estão descritos de forma clara, identificando todas as operações que devem ser realizadas;
- o) verificar se as saídas estão definidas e se estas estão devidamente relacionadas com o item Dados, ou seja, as saídas devem pertencer às tabelas, arquivos, etc. descritos no item Dados como saída;
- p) ler a lógica de processamento (funcionamento) e verificar se é possível identificar claramente no processamento que deverá ser feito a geração das saídas que foram descritas;
- q) verificar se o destino das informações processadas está claramente definido, ou seja, se está descrito claramente como os dados de saída devem ser apresentados e onde devem ser armazenados;
- r) verificar se o *layout* de telas e relatórios é apresentado;
- s) verificar a existência dos casos de teste;
- t) verificar a existência dos scripts para geração de dados para teste;
- u) verificar se os casos de teste possuem os dados de entrada, descrição, valor de entrada, resultado esperado e mensagem esperada.

5.2.2 Processo de codificação

O processo de codificação na Fábrica de Software segue padrões pré-estabelecidos, conforme a tecnologia da tarefa. O desenvolvedor de software faz a leitura dos documentos de definição de tarefa e casos de teste e inicia o trabalho de codificação.

No decorrer deste processo, o desenvolvedor, quando tiver dúvidas, consulta o analista de sistemas responsável apenas através de e-mail, de forma que toda comunicação entre as áreas fique documentada. Em caso de necessidade de alteração da definição da tarefa, o analista de sistemas atualiza o documento no *software* de gerenciamento de tarefas.

Após o término da codificação, o desenvolvedor preenche um documento de liberação da tarefa, onde constam informações técnicas acerca do que foi codificado, como por exemplo a localização física do arquivo alterado ou criado. Este documento é enviado à equipe de

testes da Fábrica de Software, que irá validar o produto da tarefa, conforme descrito no próximo item.

5.2.3 Processo de inspeção de software

O processo de inspeção de software consiste em um teste feito pelo inspetor de software, visando garantir a qualidade dos artefatos construídos dentro da Fábrica de Software. Os testes desta atividade são baseados nos casos de teste especificados e entregues juntamente com a especificação da devida tarefa.

Os testes não são executados visando a aplicação como um todo. Foca-se apenas o escopo da tarefa enviada para a Fábrica de Software. Cada tecnologia possui características diferenciadas nesta atividade, porém todas devem ter como saída o produto de software, testado e funcionando sem restrições.

O inspetor de software executa todos os casos de teste e, caso encontre qualquer inconformidade referente ao escopo da tarefa, a mesma será retornada para o desenvolvedor. Este, por sua vez, será alocado em uma tarefa de correção de software, a fim de corrigir o possível problema encontrado pelo inspetor. Caso a tarefa não apresente problemas, ela será enviada para o setor de homologação, externo ao ambiente da Fábrica de Software, finalizando o fluxo de trabalho dentro da unidade.

5.3 ENTREVISTAS

As entrevistas foram realizadas, individualmente, com 4 funcionários da Fábrica de Software estudada, conforme descrito no capítulo 4 (Método). Foi seguido um roteiro de perguntas, tendo como objetivo a expressão de idéias por parte do entrevistado.

5.3.1 Análise dos dados

A análise dos dados das entrevistas busca deixar claro quais são os indicadores necessários para o acompanhamento da produtividade da Fábrica de Software. Sendo assim, abaixo serão analisadas as respostas do roteiro de entrevista.

Análise da questão 1: “Que etapas do processo de desenvolvimento são executadas na Fábrica de Software?”

Nesta pergunta, buscou-se identificar se a Fábrica de Software segue o modelo apresentado no capítulo 3.3.2 (Fábrica de Software), onde são descritas as características de uma empresa deste tipo.

Relembrando, as etapas do processo de desenvolvimento de um projeto de software, conforme modelo *Waterfall* são as seguintes: análise e engenharia de sistema, análise de requisitos de software, projeto, codificação (desenvolvimento), testes e manutenção.

Apesar de neste estudo ter-se buscado observar apenas as etapas de codificação e testes, foi colocada a pergunta aberta aos entrevistados, no intuito de deixar emergir alguma idéia ou etapa, ou outra variável não contemplada pelo modelo. O Quadro 5 abaixo mostra as respostas por entrevistados e posicionamento do entrevistado dentro da Fábrica de Software, buscando mostrar o possível nível de entendimento do mesmo com relação ao questionamento feito:

Entrevistado	Posição na Fábrica de Software	Resposta
Luiz Fernando Schröer	Gerente da unidade	<i>“Inspeção de tarefa, codificação e teste unitário.”</i>
Ângelo Maratía	Consultor de processos	<i>“Programação e inspeção de software (teste unitário).”</i>
Rodrigo Petry	Coordenador da equipe de desenvolvimento	<i>“Inspeção de entrada para validação da tarefa, codificação e teste unitário de software.”</i>
André Tocchetto	Coordenador da equipe de testes	<i>“São realizadas atividades de validação do processo na etapa de inspeção de tarefa, atividades de construção na etapa de desenvolvimento, verificação na etapa de inspeção de software.”</i>

Quadro 5 – Respostas à questão 1: “Que etapas do processo de desenvolvimento são executadas na Fábrica de Software?”

Observando as respostas dos 4 participantes, nota-se que palavras-chave como codificação, desenvolvimento, construção e teste unitário estão presentes em todas respostas. Estas respostas fecham com as etapas codificação e teste unitário do modelo conceitual utilizado para esta pesquisa. Além disso, foi apontado por 3 entrevistados uma etapa anterior, de inspeção de tarefa, descrita no capítulo 5.2.1 (Processo de inspeção de tarefa).

Por outro lado, buscando reforçar este resultado realizou-se uma análise da documentação da metodologia de desenvolvimento de projetos definida pela Fábrica de Software no momento de sua criação. As etapas inspeção de tarefa, codificação e testes aparecem claramente como sendo aquelas que esta empresa de fabricação de *software* se especializa. Com esta análise, pode ser verificado que a unidade de produção pesquisada segue os moldes uma fábrica de *software* conforme definido na teoria, pelos autores citados no capítulo 3 (Revisão Bibliográfica).

Análise da questão 2: “Estas etapas estão formalizadas? Através de que forma?”

Através da formalização de um processo, o conhecimento em uma organização deixa de ser empírico. Desta forma, buscou-se com esta pergunta identificar se a Fábrica de Software possui a documentação dos processos que executa, de forma a garantir que os indicadores criados neste trabalho serão úteis para qualquer pessoa que venha a trabalhar com estas informações. O Quadro 6 abaixo mostra as respostas por entrevistados e posicionamento do entrevistado dentro da Fábrica de Software:

Entrevistado	Posição na Fábrica de Software	Resposta
Luiz Fernando Schröer	Gerente da unidade	<i>“Sim, documentos de Word e apresentações Power Point.”</i>
Ângelo Maratia	Consultor de processos	<i>“Sim, através de documentos, treinamentos e softwares de apoio.”</i>
Rodrigo Petry	Coordenador da equipe de desenvolvimento	<i>“As etapas estão formalizadas em documentos mantidos por uma comissão de melhoria de processos.”</i>
André Tocchetto	Coordenador da equipe de testes	<i>“O processo está documentado na enciclopédia digital da empresa.”</i>

Quadro 6 – Respostas à questão 2: “Estas etapas estão formalizadas? Através de que forma?”

Analisando as respostas, observa-se que os entrevistados apontaram que há a formalização do processo na Fábrica de Software, através de documentos digitais. Estas respostas foram confirmadas com a pesquisa feita na análise da questão anterior, onde foi verificado que o processo está devidamente documentado. Assim sendo, os indicadores aos

quais objetiva-se chegar não serão criados com base na visão de um ou outro funcionário, e sim com base na visão organizacional.

Análise da questão 3: “Quais os indicadores que são utilizados pela Fábrica de Software atualmente? De que forma estes indicadores foram levantados?”

Buscou-se, com esta questão, identificar o nível de preocupação e também a maturidade do processo de acompanhamento do desenvolvimento de software na Fábrica. Assim, as respostas a esta questão ajudam na compreensão do ambiente, item de grande importância conforme citação no capítulo 4 (Método).

Os entrevistados apontaram itens dispersos, cada qual focando em sua área de atuação. O coordenador da qualidade, por exemplo, apontou indicadores utilizados no que se refere aos testes de tarefa, como o número de testes que uma determinada tarefa sofreu. O Quadro 7, apresentado a seguir, mostra as respostas por entrevistados e posicionamento destes dentro da Fábrica de Software:

Entrevistado	Posição na Fábrica de Software	Resposta
Luiz Fernando Schröer	Gerente da unidade	<i>“Percentual de ociosidade, horas produtivas e improdutivas, custo por hora de trabalho, horas por área de negócio e tecnologia. Estes indicadores são extraídos manualmente do banco de dados.”</i>
Ângelo Maratia	Consultor de processos	<i>“Indicadores relativos à qualidade do processo, levantados a partir do banco de dados. Exemplo: retrabalho, ociosidade, número de tarefas que entraram e saíram.”</i>
Rodrigo Petry	Coordenador da equipe de desenvolvimento	<i>“Indicadores de qualidade e desempenho, levantados através de consulta no banco, tais como: ocupação/ociosidade dos recursos, horas de trabalho em cada tecnologia, tarefas entregues e recebidas.”</i>
André Tocchetto	Coordenador da equipe de testes	<i>“Ociosidade dos recursos, retrabalho, tipos de atividade por tecnologia, ocorrências de erro.”</i>

Quadro 7 – Respostas à questão 3: “Quais os indicadores que são utilizados pela Fábrica de Software atualmente? De que forma estes indicadores foram levantados?”

Dos indicadores mais evidenciados pelos entrevistados, e que estão relatados no capítulo 3.3.3 (Medidas de produtividade e desempenho em desenvolvimento de software), estão:

- a) densidade de defeitos (número de defeitos no software sobre o tamanho em horas do software);
- b) custo por hora de desenvolvimento;
- c) quantidade de horas em retrabalho.

Além destes indicadores apontados nas referências bibliográficas, notou-se o destaque dado aos seguintes indicadores:

- a) percentual de ociosidade da equipe;
- b) percentual de tempo trabalhado por tecnologia;
- c) utilização de recursos conforme área de negócio;
- d) quantidade de tarefas recebidas por período;
- e) quantidade de tarefas entregues por período.

É importante também salientar nesta análise que, segundo os entrevistados, o levantamento destes dados é feito manualmente, extraíndo-os direto dos bancos de dados dos sistemas da empresa.

Análise da questão 4: “Como é feito o planejamento das tarefas demandadas à Fábrica de Software?”

Esta pergunta foi formulada visando identificar se a Fábrica de Software possui algum critério para o planejamento das tarefas demandadas. Através das respostas, observou-se que as tarefas são planejadas individualmente, para os recursos que estarão disponíveis nas datas planejadas da tarefa. Também é levado em conta o tipo de atividade e a tecnologia envolvida para a alocação de um determinado recurso. O planejamento das tarefas é feito com o auxílio de um software chamado RPM (*Rational Portfolio Manager*), da multinacional IBM. O Quadro 8 abaixo mostra as respostas por entrevistados e posicionamento do entrevistado dentro da Fábrica de Software:

Entrevistado	Posição na Fábrica de Software	Resposta
Luiz Fernando Schröer	Gerente da unidade	<i>“Individualmente, com auxílio de uma ferramenta de gestão de portfolio de</i>

		<i>projetos.”</i>
Ângelo Maratia	Consultor de processos	<i>“Através de pacotes de trabalho agrupados em uma estrutura. As datas e recursos para execução das tarefas são sugeridos pelo analista do projeto, e confirmada ou não pelo coordenador de recurso, conforme disponibilidade.”</i>
Rodrigo Petry	Coordenador da equipe de desenvolvimento	<i>“A demanda é planejada de acordo com o tipo de tecnologia e a disponibilidade de recursos, controlada pelo RPM (Rational Portfolio Manager).”</i>
André Tocchetto	Coordenador da equipe de testes	<i>“Procura-se um recurso com habilidade para a tecnologia da tarefa, e que tenha datas disponíveis mais próximas.”</i>

Quadro 8 – Respostas à questão 4: “Como é feito o planejamento das tarefas demandadas à Fábrica de Software?”

Analisando os dados levantados, chegou-se à conclusão que este item aponta a necessidade de utilizar um indicador que auxilie no planejamento das tarefas, visto que os entrevistados apontaram que o planejamento é sempre feito por demanda. O indicador de estimativa de atendimento, apontado no quadro de exemplos de indicadores no capítulo 3.3.3 (Medidas de produtividade e desempenho em desenvolvimento de software) foi considerado o mais apropriado para auxiliar no planejamento das tarefas. Desta forma, a gerência da Fábrica de Software pode enviar à área demandante uma previsão de atendimento por períodos, não havendo problemas de indisponibilidade de equipe em alguns momentos.

Análise da questão 5: “Na concepção da Fábrica de Software, é mais importante: tempo, qualidade ou custo?”

Como muitos dos indicadores levantados baseiam-se nestas três variáveis, esta pergunta foi feita de forma a identificar a variável para a qual é dada maior importância na Fábrica de Software. O Quadro 9 abaixo mostra as respostas por entrevistados e posicionamento do entrevistado dentro da Fábrica de Software:

Entrevistado	Posição na Fábrica de Software	Resposta
Luiz Fernando Schröer	Gerente da unidade	<i>“Todos, mas na seguinte ordem (no momento): tempo, qualidade e custo.”</i>
Ângelo Maratia	Consultor de processos	<i>“Tempo.”</i>
Rodrigo Petry	Coordenador da equipe de desenvolvimento	<i>“Tempo e qualidade são importantes, mas a otimização dos custos é mais importante.”</i>
André	Coordenador da equipe	<i>“Qualidade, pois a falta dela acarreta em</i>

Tocchetto	de testes	<i>necessidade de terceirização, o que aumenta o custo e o tempo.</i>
-----------	-----------	---

Quadro 9 – Respostas à questão 5: “Na concepção da Fábrica de Software, é mais importante: tempo, qualidade ou custo?”

Ao analisar as respostas, observa-se que cada entrevistado tende a dar maior importância para a sua área. O gerente, por sua vez, aponta todas variáveis como sendo importantes dentro do processo produtivo da unidade. Com isto, observou-se que na Fábrica de Software não está disseminada a idéia do que é mais importante, mesmo porque todas variáveis são consideradas importantes. Cada equipe foca-se naquilo que é de sua responsabilidade, buscando-se atingir o resultado coletivo. Todavia, haverá uma preocupação maior com os indicadores que apontam tempo e confiabilidade na entrega, como os dois abaixo, averiguados no referencial teórico deste trabalho:

- a) percentual de projetos ou solicitações de serviços (manutenção ou programa) entregues no prazo acordado com o cliente, conforme um período de tempo determinado;
- b) percentual de redução do tempo de entrega de produtos no período.

Análise da questão 6: “Como é estimado o tempo necessário para o desenvolvimento das tarefas?”

Como no levantamento bibliográfico de indicadores foi verificado a importância do indicador que aponta o desvio das estimativas de tamanho de *software*, julgou-se necessário verificar se a Fábrica de Software segue alguma metodologia. Somente desta forma pode ser estimado o esforço em horas necessário para o desenvolvimento de uma tarefa. As respostas para esta pergunta seguem no Quadro 10 abaixo:

Entrevistado	Posição na Fábrica de Software	Resposta
Luiz Fernando Schröer	Gerente da unidade	<i>“Através da metodologia de enquadramento.”</i>
Ângelo Maratia	Consultor de processos	<i>“Tabela de enquadramento.”</i>
Rodrigo Petry	Coordenador da equipe de desenvolvimento	<i>“Atualmente, através de tabela de enquadramento, onde as principais características das tarefas são contabilizadas e um peso é atribuído a cada uma delas.”</i>
André Tocchetto	Coordenador da equipe de testes	<i>“Na etapa de inspeção de tarefa existe o enquadramento, que avalia artefatos quantitativos de uma tarefa e fatores de</i>

		<i>ajuste, para determinar o tempo necessário para a tarefa.”</i>
--	--	---

Quadro 10 – Respostas à questão 6: “Como é estimado o tempo necessário para o desenvolvimento das tarefas?”

A análise das respostas aponta a utilização de uma metodologia própria de estimativa de tamanho de software, denominada tabela de enquadramento. Buscando reforçar este resultado realizou-se uma análise da documentação e confirmou-se a utilização desta metodologia. Este sistema de fato baseia-se em estipular pesos para atributos de cada tarefa, de forma a chegar em um tempo estimado para a construção de determinada tarefa.

Por fim, devido à utilização de uma metodologia por parte da Fábrica de Software, mostra-se importante a utilização de um indicador que aponte a eficácia ou não deste método. Desta maneira, o método poderá ser aperfeiçoado pela unidade, de forma a ter estimativas cada vez mais precisas.

Análise da questão 7: “É feita alguma previsão de demanda de atendimento? Se sim, de quê forma?”

Esta pergunta ajudou a complementar o que foi verificado na análise da questão 4 (Como é feito o planejamento das tarefas demandadas à Fábrica de Software?). Nela, foi verificada a necessidade de criação de um indicador que auxilie no planejamento das tarefas, sendo sugerida a utilização do indicador de estimativa de atendimentos. O Quadro 11, apresentado a seguir, mostra as respostas por entrevistados e posicionamento destes dentro da Fábrica de Software:

Entrevistado	Posição na Fábrica de Software	Resposta
Luiz Fernando Schröer	Gerente da unidade	<i>“Não existe previsão para um horizonte maior que uma semana, geralmente. Somente temos uma previsão depois que há projetos planejados no RPM.”</i>
Ângelo Maratia	Consultor de processos	<i>“Apenas para curto prazo.”</i>
Rodrigo Petry	Coordenador da equipe de desenvolvimento	<i>“Não. A Fábrica de Software depende do planejamento dos clientes.”</i>
André Tocchetto	Coordenador da equipe de testes	<i>“Não há como prever demanda pois depende das requisições da análise.”</i>

Quadro 11 – Respostas à questão 7: “É feita alguma previsão de demanda de atendimento? Se sim, de quê forma?”

Como resposta dos entrevistados, dois deles apontaram com certeza que não há uma previsão de demanda. Por outro lado, observou-se em outras duas respostas que há uma

previsão apenas de curto prazo. Desta forma, o indicador de estimativa de atendimentos, apontado na análise da questão 4, pode servir para facilitar a Fábrica de Software no que diz respeito à informação de disponibilidade de seus recursos, para que a área de análise demande mais ou menos tarefas, dependendo da ocupação atual dos recursos da Fábrica.

Análise da questão 8: “Na organização em que está inserida a Fábrica de Software, no que pode impactar um erro em um *software* que já foi publicado?”

Diante do levantamento de alguns indicadores referentes a erros de *software* apontados pelos autores, esta pergunta mostra-se fundamental para identificar a importância deste tipo de índice para a Fábrica de Software. Dependendo o tipo da organização, um erro de *software* pode ter diferentes tipos de impactos (principalmente financeiros). Abaixo é apresentado o Quadro 12, que mostra as respostas por entrevistados e posicionamento destes dentro da Fábrica de Software:

Entrevistado	Posição na Fábrica de Software	Resposta
Luiz Fernando Schröer	Gerente da unidade	<i>“Perda de faturamento.”</i>
Ângelo Maratia	Consultor de processos	<i>“Indisponibilidade de produtos e serviços aos clientes. O sucesso do negócio depende de funcionamento de software.”</i>
Rodrigo Petry	Coordenador da equipe de desenvolvimento	<i>“Erros podem provocar prejuízos enormes, cancelamento de contratos com clientes, parada nos serviços 24x7 (24 horas, 7 dias por semana).”</i>
André Tocchetto	Coordenador da equipe de testes	<i>“Se encontrados já em ambiente de produção, custará inúmeras vezes mais do que foi gasto para o desenvolvimento, sem contar na insatisfação do cliente.”</i>

Quadro 12 – Respostas à questão 8: “Na organização em que está inserida a Fábrica de Software, no que pode impactar um erro em um *software* que já foi publicado?”

Ao fazer a análise das respostas, pode ser observado, através de termos como “perda de faturamento”, “prejuízos enormes” e “negócio dependente de software”, que a empresa é completamente dependente dos programas produzidos pela Fábrica de Software. Desta forma, um erro na produção, sem ser apontado no processo de testes, pode causar enormes prejuízos financeiros à organização.

Com esta análise, mostra-se fundamental a utilização de indicadores que possam evitar a ocorrência destes erros. Todavia, foi informado pelos entrevistados que a Fábrica de Software não tem a informação de origem dos erros. Logo, não existe forma, atualmente, de relacionar um erro em ambiente de produção com a tarefa executada pela Fábrica de Software. Sendo assim, o resultado da análise desta questão não aponta nenhum indicador que possa ser utilizado no modelo que será proposto.

Análise da questão 9: “Dos indicadores abaixo, marque os 5 que você considera mais importantes para o contexto que está inserido a Fábrica de Software?”

Utilizando-se de exemplos citados na revisão bibliográfica, esta questão enumerou dez indicadores, solicitando ao entrevistado que fossem apontadas aqueles que julgasse de maior importância para a Fábrica de Software. O Quadro 13 apresenta as alternativas apontadas por cada um dos entrevistados:

Entrevistado	Posição na Fábrica de Software	Respostas assinaladas
Luiz Fernando Schröer	Gerente da unidade	Densidade de defeitos (número de defeitos no software sobre o tamanho em horas do software)
		Eficiência de remoção de defeitos
		% de projetos ou solicitações de serviços (manutenção ou programa) entregues no prazo acordado com o cliente, conforme um período de tempo determinado
		Custo por hora de desenvolvimento
		Desvio das estimativas de tamanho de software
Ângelo Maratia	Consultor de processos	Densidade de defeitos (número de defeitos no software sobre o tamanho em horas do software)
		Eficiência de remoção de defeitos
		% de projetos ou solicitações de serviços (manutenção ou programa) entregues no prazo acordado com o cliente, conforme um período de tempo determinado
		Efetividade das revisões (inspeções de software)
		Desvio das estimativas de tamanho de software
Rodrigo Petry	Coordenador da equipe de desenvolvimento	Densidade de defeitos (número de defeitos no software sobre o tamanho em horas do software)
		% de projetos ou solicitações de serviços (manutenção ou programa) entregues no prazo acordado com o cliente, conforme um período

		de tempo determinado
		% de redução do tempo de entrega de produtos no período
		Efetividade das revisões (inspeções de software)
		% de projetos e serviços dentro do orçamento ou abaixo
André Tocchetto	Coordenador da equipe de testes	Densidade de defeitos (número de defeitos no software sobre o tamanho em horas do software)
		% de projetos ou solicitações de serviços (manutenção ou programa) entregues no prazo acordado com o cliente, conforme um período de tempo determinado
		Custo por hora de desenvolvimento
		% de redução de custo do processo no período
		Desvio das estimativas de tamanho de software

Quadro 13 – Respostas à questão 9: “Dos indicadores abaixo, marque os 5 que você considera mais importantes para o contexto que está inserido a Fábrica de Software?”

A análise desta questão aponta uma unanimidade na alternativa “Densidade de defeitos”. Este indicador, já apontado na análise da questão 3 como um dos já utilizados pela Fábrica de Software, mostra-se importante para o modelo que desejasse chegar.

A pontualidade também é verificada como um fator importante para a unidade de negócios. Todos os entrevistados responderam que o indicador “% de projetos ou solicitações de serviços (manutenção ou programa) entregues no prazo acordado com o cliente” está entre os mais importantes para o controle de desempenho da Fábrica de Software. Contempla-se, desta forma, o indicador já levantado na análise da questão 5.

Apontado por três dos entrevistados, o indicador de desvio das estimativas de tamanho de software mostra-se necessário para suprir a necessidade de um acompanhamento da eficiência da estimativa das tarefas, apontada na análise da questão 6. Já com duas respostas cada, aparecem as seguintes alternativas:

- a) efetividade das revisões (inspeções de software), determinada, segundo os entrevistados, através da divisão entre o número de erros encontrados pela inspeção de *software* e o total de erros encontrados (tanto em inspeção de *software* quanto homologação);
- b) custo por hora de desenvolvimento, apontado como um indicador já utilizado pela Fábrica de Software na questão 3;
- c) eficiência de remoção de defeitos que, apesar de ser importante no ponto de vista dos entrevistados, não é viável no momento, segundo os próprios respondentes,

visto que a Fábrica de Software não possui uma rastreabilidade dos erros. Desta forma, não há como identificar qual atividade gera determinados erros, após ter sido publicada em produção.

Há também outros indicadores apontados apenas por um entrevistado cada. Porém, estes indicadores não serão levados em conta nesta análise, devido à necessidade de se fechar um modelo, para a Fábrica de Software, que seja útil para todos aqueles que os utilizarão.

Análise da questão 10: “Existem indicadores que você julga necessário a Fábrica de Software ter, porém ainda não implantou? Que benefícios estes indicadores poderiam trazer, se adotados?”

Para finalizar a entrevista, foi feita uma pergunta aberta, de forma a deixar que o entrevistado expressasse as necessidades de indicadores da Fábrica de Software. As respostas são apresentadas no Quadro 14 abaixo:

Entrevistado	Posição na Fábrica de Software	Resposta
Luiz Fernando Schröer	Gerente da unidade	<i>“Sim, de comparação/benchmarking, posicionamento perante a concorrência.”</i>
Ângelo Maratia	Consultor de processos	<i>“Grau de dependência da Fábrica com os analistas, no que diz respeito ao contato pessoal (extra processo).”</i>
Rodrigo Petry	Coordenador da equipe de desenvolvimento	<i>“Indicadores de produtividade: possibilitam focar os esforços na produtividade e permitem acompanhar a evolução dos índices. Auxilia na redução de custos.”</i>
André Tocchetto	Coordenador da equipe de testes	<i>“Indicadores de defeitos relacionados com o tamanho do software e Defeitos por hora gasta em teste.”</i>

Quadro 14 – Respostas à questão 10: “Existem indicadores que você julga necessário a Fábrica de Software ter, porém ainda não implantou? Que benefícios estes indicadores poderiam trazer, se adotados?”

Analisando as respostas, verifica-se que não houve repetição nas respostas. Apenas um dos entrevistados (coordenador da equipe de testes) apontou indicadores que são possíveis de serem extraídos atualmente. Os indicadores apontados pelos demais entrevistados, segundo eles mesmos, não são viáveis no momento, visto que não há dados suficientes.

Desta forma, podem ser levados em consideração para o fechamento do modelo os seguintes indicadores:

- a) defeitos por hora gasta em teste;
- b) defeitos por tamanho de software (número de defeitos sobre a estimativa do tamanho do software), já apontado nas questões 3 e 9, através do item de densidade de defeitos.

5.4 CONCLUSÕES DA ANÁLISE DOS DADOS

Como conclusão da análise das respostas, podem ser definidos os indicadores a serem utilizados no modelo que objetiva-se chegar neste trabalho. O Quadro 15 apresenta de forma sintética a escolha destes indicadores, informando quais questões foram relevantes para a escolha de cada um:

Indicador	Questões relevantes
Densidade de defeitos (número de defeitos no software sobre o tamanho em horas do software);	3, 9, 10
Custo por hora de desenvolvimento	3, 9
Quantidade de horas em retrabalho	3
Percentual de ociosidade da equipe	3
Percentual de tempo trabalhado por tecnologia	3
Utilização de recursos conforme área de negócio	3
Quantidade de tarefas recebidas por período	3
Quantidade de tarefas entregues por período	3
Estimativa de atendimento	4, 7
Percentual de projetos ou solicitações de serviços entregues no prazo acordado com o cliente	5, 9
Percentual de redução do tempo de entrega de produtos no período	5
Desvio das estimativas	6
Efetividade das revisões	9
Defeitos por hora gasta em teste	10

Quadro 15 – Indicadores selecionados para compor o modelo.

Ficam estipulados, desta forma, os indicadores que serão utilizados como modelo para acompanhar o desempenho da Fábrica de Software. Estes indicadores serão apresentados para os gestores da unidade através de uma ferramenta de *Dashboard*. *Dashboard*, conforme

citado no tópico 3.2.2.2, é uma ferramenta que fornece informações imediatas sobre o desempenho dos negócios em toda a empresa através de um painel de controle.

No próximo capítulo será desenvolvido o protótipo deste sistema de *Dashboard*. Os dados apresentados no sistema de acompanhamento devem ser organizados em um *Data Warehouse*. Porém, o foco deste trabalho será apenas em propor a forma de apresentação destes indicadores (*Dashboard*), ficando o *Data Warehouse* como sugestão para continuidade futura deste estudo de caso.

6 PROTOTIPAÇÃO DO SISTEMA

Como forma de apresentar os indicadores levantados através do estudo de caso, propõe-se neste capítulo a utilização do conceito de *Dashboard* para visualização destes indicadores. A forma gráfica como os dados são apresentadas em um *Dashboard*, conforme citado no item 3.2.2.2, ajuda os gestores pois fornece informações imediatas sobre o desempenho.

Foi desenvolvido um protótipo executável que, como já visto no tópico 3.3.1 (O processo de desenvolvimento de software), é obtido através da implementação, simplificada ou não, de um subconjunto de funções do sistema em um computador que não é necessariamente a máquina onde será instalado o sistema final. Desta forma, buscou-se com este protótipo facilitar à Fábrica de Software uma implementação final, mais completa e sofisticada, de um sistema para acompanhamento dos indicadores.

Abaixo, são apresentadas as telas do protótipo, desenvolvido na linguagem de programação *Cold Fusion*, conforme citado no capítulo 4 (Método). Este protótipo foi implementado para contemplar a apresentação dos indicadores levantados, seguindo o conceito de *Dashboard* (painel de controle).

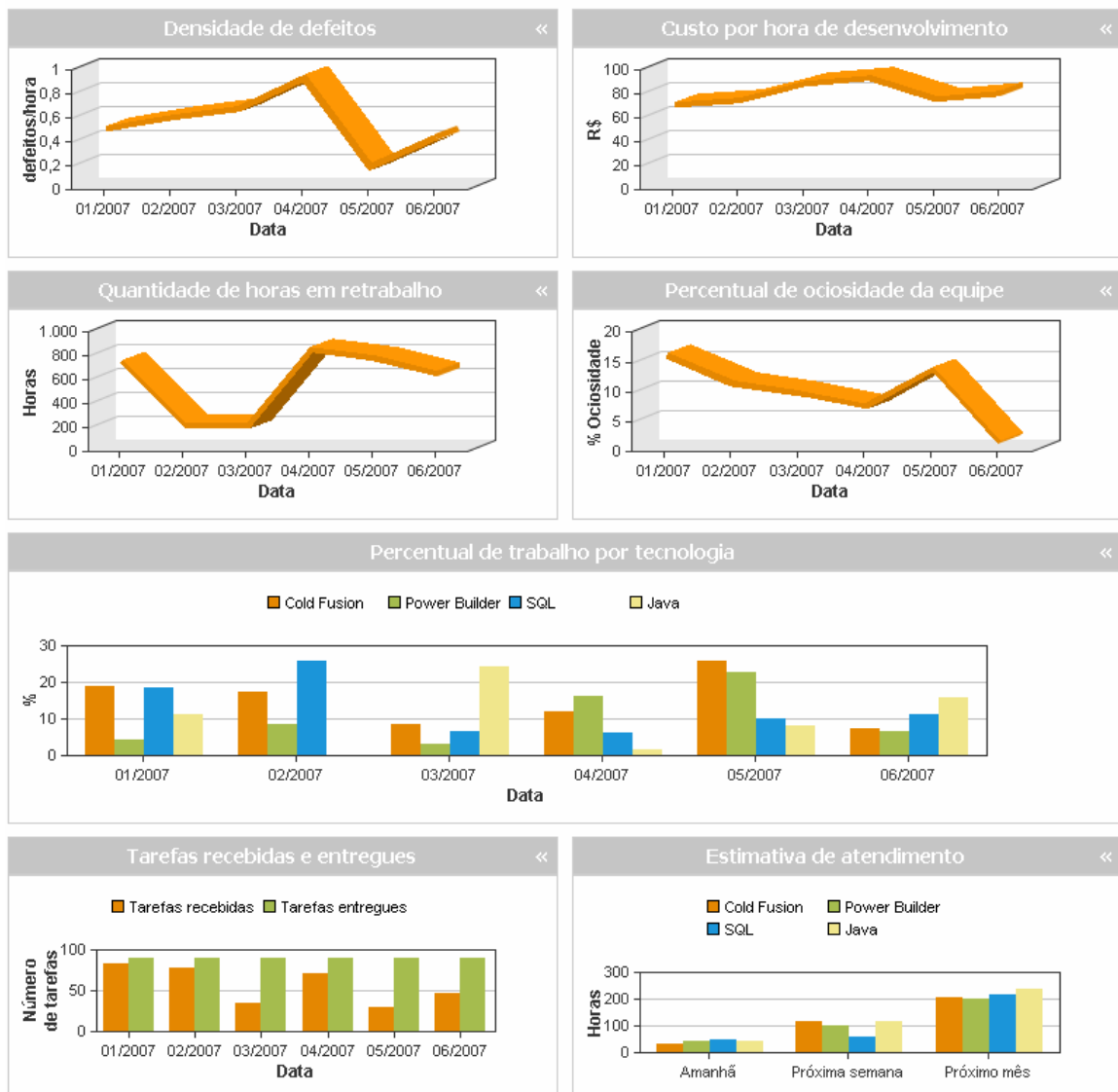


Figura 4 – Protótipo do Dashboard - Parte 1 de 2

Fonte: elaborada pelo autor

Como pode ser observado, a imagem acima apresenta sete gráficos, contendo no total oito indicadores do modelo sugerido, visto que o gráfico de tarefas recebidas e entregues contempla dois indicadores. Na imagem a seguir, são apresentados os demais indicadores do modelo:

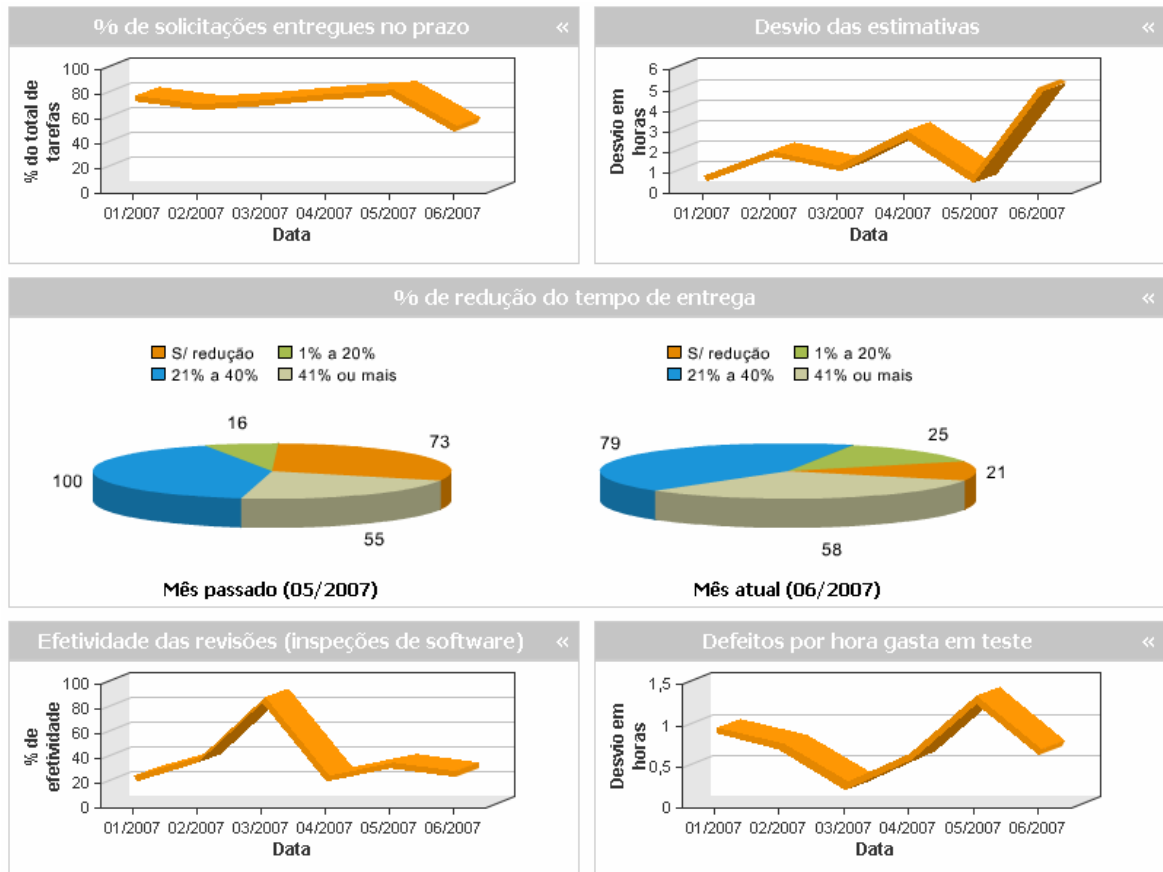


Figura 5 – Protótipo do Dashboard - Parte 2 de 2

Fonte: elaborada pelo autor

7 CONSIDERAÇÕES FINAIS

Este trabalho atingiu seu objetivo principal que era propor um sistema de medidas de desempenho para uma empresa de desenvolvimento de *software*, baseada em um modelo de fábrica de *software*.

Os objetivos específicos também foram atingidos como se pode ver através do desenvolvimento do estudo de caso, onde foi analisado o processo de desenvolvimento de software da Fábrica de Software estudada. Também, para contemplar os objetivos específicos, foi definido um conjunto de indicadores que melhor representam o desempenho da unidade de negócio e, por fim desenvolvido um protótipo de sistema de medidas de desempenho para visualização, com atualização diária, dos indicadores formulados.

7.1 PRINCIPAIS CONTRIBUIÇÕES

Este trabalho teve como contribuição, para a empresa onde foi desenvolvido, a possibilidade de definir quais são os principais indicadores que devem ser levados em conta para a medição de desempenho da unidade de negócio. Além disso, o protótipo desenvolvido pode ser de valia para a empresa, uma vez que se sugere, através dele, como estes indicadores averiguados podem ser apresentados em um sistema.

Para o meio acadêmico, este trabalho torna-se importante pelo fato que reforça a importância de medição em sistemas produtivos de serviços, como é o caso de uma fábrica de *software*. A ênfase na qual este trabalho foi desenvolvido (Produção e Sistemas) explora muito o lado da produção de manufatura. Este estudo, feito em uma empresa prestadora de serviços, mostra a necessidade de haver medidas de desempenho em qualquer sistema produtivo.

7.2 LIMITAÇÕES E PESQUISAS FUTURAS

Uma das principais limitações deste trabalho é o fato de ter sido feito um protótipo

com indicadores em uma empresa específica. Outras empresas que desenvolvam software (seja como terceirizados ou internamente) podem necessitar de outros indicadores e talvez outra plataforma tecnológica para desenvolver seu sistema de medidas de desempenho.

Também pode ser considerado como limitação o tempo de desenvolvimento do trabalho, visto que não foi possível desenvolver uma estrutura de dados para extração e carga das informações para geração dos indicadores. Desta forma, sugere-se que este estudo seja continuado dentro da empresa, de forma a estabelecer esta estrutura, para que sejam completos todos os requisitos para implementação do sistema de medidas de desempenho proposto.

Outra sugestão de pesquisa futura é a busca pelos valores ideais para cada indicador da Fábrica de Software. Com o levantamento de metas, o *Dashboard* poderá ser incrementado, informando os pontos críticos para o gestor, de forma imediata. Este levantamento não pôde ser desenvolvido neste trabalho, visto que é necessário, primeiramente, coletar dados históricos para poder, logo após, compará-los com outras empresas do mercado.

REFERÊNCIAS

ABES, Associação Brasileira das Empresas de Software. Disponível em: <http://www.abes.org.br>. Acessado no dia 04/06/2007.

ANSOFF, I. **Estratégia Empresarial**. São Paulo: McGraw Hill, 1977.

ANTHONY, R. N. **Planning and control systems: a framework for analysis**. Harvard Graduate School of Business, Boston, 1965.

BARBIERI, C. **BI Business Intelligence**. Rio de Janeiro: Axcel, 2001.

BEVAN, S.; THOMPSON, M. **Performance management at the crossroads**. Personnel Management, November 1991.

BUSINESS WEEK, disponível em http://www.businessweek.com/magazine/content/06_07/b3971083.htm. Acessado no dia 11/04/2007.

COSTA, Ivanir. **Contribuição para o aumento da qualidade e produtividade de uma Fábrica de Software através da padronização do processo de recebimento de serviços de construção de software**. Tese de Doutorado em Engenharia de Produção. Universidade de São Paulo, USP, 2003.

CROSS, K. F.; LYNCH, R. L. **Managing the corporate warriors**. Quality Progress, v. 23, n. 4, 1990, p. 54-59.

ELMASRI, Ramez. **Sistemas de banco de dados**, 4ª edição. São Paulo: Pearson/Addison Wesley, 2005.

FERNANDES, A.A., Teixeira, D. S. **Fábrica de Software: Implantação e gestão de Operações**. São Paulo: Atlas, 2004.

GRAY, Paul, WATSON, Hugh J. **Decision Support in the Data Warehouse**. New Jersey: Prentice Hall PTR, 1998.

HAN, J.; KAMBER, M. **Data mining – concepts and techniques**. San Francisco: Morgan Kaufmann Publishers, 2001.

KAPLAN, R. S. e NORTON, D. P. **A Estratégia em Ação: Balanced Scorecard**. Rio de Janeiro: Campus, 1997.

KAPLAN, R. S. **Measuring manufacturing performance: a new challenge for managerial accounting research**. Accounting Review, v. LVIII, n. 4, p. 686-705, 1983.

KAPLAN, R. S. e NORTON, D. P. **Utilizando o balanced scorecard como sistema gerencial estratégico**. Rio de Janeiro: Campus, 2000.

MARTINS, R.A. **Sistemas de medição de desempenho: um modelo para estruturação do uso** - Tese de doutorado apresentada à Escola Politécnica da Universidade de São Paulo, São Paulo, 1998.

McMANN, P.; NANNI, JR., A. J. **Is your company really measuring performance?** Management Accounting, v.76, n.1, pp.55-58, nov. 1994.

MEYER JÚNIOR, Vitor. **Considerações sobre o planejamento estratégico na universidade.** In: FINGER, Almeri Paulo. Universidade: organização, planejamento, gestão. Florianópolis UFSC/ CPGA/NUPEAU. 1988. p.53-69.

MINTZBERG, H; AHLSTRAND, B.; LAMPEL, J. **Safári de estratégia** - um roteiro pela selva do planejamento estratégico. Porto Alegre: Bookman, 2000.

OLIVEIRA, D.P. R. **Planejamento Estratégico, conceitos, metodologias e práticas**, 15ª edição. São Paulo: Atlas, 2001.

PORTER, M. E. **Estratégia Competitiva Empresarial.** Rio de Janeiro: Campus, 1998.

PRESSMAN, Roger S. **Engenharia de software.** São Paulo: Makron Books do Brasil, 1995.

SOMMERVILLE, Ian . **Engenharia de Software**, 6ª edição. São Paulo: Pearson Addison Wesley, 2003.

STAKE, Robert E.. **The art of case study research.** Thousand Oaks: Sage, 1995.

STONER, James A. F.; FREEMAN, R. Edward. **Administração.** 5. ed. (Trad.) Alves Calado. Rio de Janeiro: Prentice-Hall do Brasil, 1982.

TATIKONDA, Lakshmi V., TATIKONDA, Rao J. **We need dynamic performance measures.** Management Accounting. Vol. 80, N° 3, p. 49-51, set. 1998.

TIRONI, Luís Fernando. **Indicadores da Qualidade e Produtividade: Conceitos e Usos**, in Revista Indicadores da Qualidade e Produtividade – ano 1, Fevereiro 93, Programa Brasileiro de Qualidade e Produtividade - PBQP, Instituto de Pesquisa Econômica Aplicada - IPEA.

WOILER, Samsão; MATHIAS, Washington Franco. **Projetos: Planejamento, elaboração e análise.** São Paulo: Atlas, 1996.

YIN, Robert K.. **Estudo de caso: planejamento e métodos.** 2ª edição. Porto Alegre: Bookman, 2001.

ANEXO A – FORMULÁRIO DE ENTREVISTA



UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL
ESCOLA DE
ADMINISTRAÇÃO

**Instrução:**

Por favor, responda as perguntas abaixo com base na sua percepção e conhecimento da Fábrica de Software.

Nome: _____

- 1. Que etapas do processo de desenvolvimento são executadas na Fábrica de Software?**

- 2. Estas etapas estão formalizadas? Através de que forma?**

- 3. Quais os indicadores que são utilizados pela Fábrica de Software atualmente? De que forma estes indicadores foram levantados?**

- 4. Como é feito o planejamento das tarefas demandadas à Fábrica de Software?**

- 5. Na concepção da Fábrica de Software, é mais importante: tempo, qualidade ou custo?**

6. Como é estimado o tempo necessário para o desenvolvimento das tarefas?

7. É feita alguma previsão de demanda de atendimento? Se sim, de quê forma?

8. Na organização em que está inserida a Fábrica de Software, no que pode impactar um erro na produção de *software*?

9. Dos indicadores abaixo, marque os 5 que você considera mais importantes para o contexto que está inserido a Fábrica de Software?

Densidade de defeitos (número de defeitos no software sobre o tamanho em horas do software)

Eficiência de remoção de defeitos (número de defeitos reportados pelos clientes na operação do software dividido pelo somatório do número de defeitos encontrados e removidos no teste de pré-release, mais o número de defeitos reportados pelos clientes)

% de projetos ou solicitações de serviços (manutenção ou programa) entregues no prazo acordado com o cliente, conforme um período de tempo determinado

% de redução do tempo de entrega de produtos no período

Custo por linha de código

Custo por hora de desenvolvimento

% de redução de custo do processo no período

% de projetos e serviços dentro do orçamento ou abaixo

Efetividade das revisões (inspeções de software)

Desvio das estimativas de tamanho de software

10. Existem indicadores que você julga necessário a Fábrica de Software ter, porém ainda não implantou? Que benefícios estes indicadores poderiam trazer, se adotados?

Obrigado pela sua participação!

Gabriel Hoff

ANEXO B – MODELO DE DOCUMENTO DE DEFINIÇÃO DE TAREFA

DADOS GERAIS					
Título:	< nome da tarefa >				
Analista:	< nome completo da analista de sistemas >				
Programador:					
Aplicação:	< nome da aplicação e URL de acesso a aplicação >				
Menu/Acesso:	< localização no menu da aplicação >				
Primeira Definição: dd/mm/aaaa	Última Alteração: 2/7/2007	Tempo estimado: X horas	Previsão de conclusão: dd/mm/aaaa	Número da Tarefa no RPM: < Número do projeto/número da tarefa gerados pelo App no momento do cadastro da tarefa >	Identificação Documento Homologação/Itens: <utilizado para tarefas criadas a partir de problemas identificados pela homologação>
Localização dos demais documentos do projeto:	< nome e localização dos documentos do projeto >				

DEFINIÇÃO DE TAREFA			
Nova Funcionalidade [X]	Alteração na aplicação []	Correção na aplicação []	Alteração tarefa original []
[] Revisar Padrões Web nos Templates Indicados no Item Programas			

DIAGRAMA ER
<diagrama entidade relacionamento>

DIAGRAMA DO COMPONENTE
<diagrama dos componentes relacionados a esta tarefa>

PROGRAMA(S)	
< nome físico dos programas que devem ser alterados ou consultados pelo programador >	< descrição abreviada do programa >

STORED PROCEDURES
< nome físico das stored procedures relacionadas com esta especificação >
IMPORTANTE: toda alteração realizada em procedures deve ser descrita no cabeçalho das mesmas, no item “Revisões”. Deve ser colocado no seguinte estilo: - <data> - <nome do desenvolvedor> <descrição da alteração>

TRIGGERS
< nome físico das triggers relacionadas com esta especificação >

USUÁRIO(S)
<i>< Nome e senha do usuário que deve ser utilizado para login na aplicação. Se existem perfis diferentes de usuários que modificam o comportamento do programa ou as informações apresentadas, identifique-os ></i>

OBJETIVO(S)
<i>< objetivo da implementação/manutenção ></i>

LAY-OUT
<i>< lay-out de telas, relatórios, arquivos, etc...></i>

DADOS/MÉTODOS

Tabelas/Métodos, Arquivos, etc.	E	S	Leitura	Inclusão	Alteração	Exclusão
<i>< nome da tabela, arquivo ou qualquer origem dos dados ></i>						
...

- E** – Marque com um “X” se forem dados de entrada
S – Marque com um “X” se forem dados de saída
Leitura – Marque com um “X” se os dados forem lidos
Inclusão – Marque com um “X” se os dados forem incluídos
Alteração – Marque com um “X” se os dados forem alterados
Exclusão – Marque com um “X” se os dados forem excluídos

ENTRADAS

Ordem	Nome Físico	Descrição/Opções	Formato	Label	Editável	Valor Default	Obrigatório	Validações/Mensagens
<i>#</i>	<i>< nome do dado em termos de negócio ></i>	<i>< descrição do dado e as opções possíveis se existente ></i>	<i>< formato do dado ></i>	<i>< texto que será utilizado na interface ></i>	<i>< “Sim” se editável, caso contrário “Não” ></i>	<i>< valor default do dado ></i>	<i>< “Sim” se obrigatório, caso contrário “Não” ></i>	<i>< validações e mensagens em tempo de tela ></i>
...

- Número seqüencial que determina a ordem de apresentação do dado

< caso existam mais de uma entrada, estas devem ser descritas repetindo-se as informações anteriores >

PRÉ-CONDIÇÃO

< condição ou condições que devem ser atendidas antes da execução da lógica de processamento. Na implementação (codificação) serão as condições que devem ser testadas antes da execução da lógica de processamento. Caso as condições sejam obedecidas, a lógica de processamento é executada, caso contrário, a lógica de processamento não é executada e uma mensagem de advertência deverá ser dada interrompendo a execução da funcionalidade >

FUNCIONAMENTO

< Descrição da lógica de processamento. Procurar especificar através da descrição das informações das entradas, do processamento efetuado e das saídas geradas, nesta ordem. Caso exista mais de um programa/processamento (entenda processamento como uma rotina específica dentro de um programa), especificar para cada processamento as suas entradas, lógica de processamento e as saídas geradas. >

< não use mnemônicos, siglas, ou qualquer outra abreviatura que possa dificultar o entendimento desta especificação. Caso seja necessário, crie uma legenda >
< evite textos longos substituindo-os por diagramas (fluxogramas) para esclarecer a lógica de processamento >

Nome do programa/processamento:

Entradas:

Lógica de Processamento:

Saídas:

Nome do programa/processamento:

Entradas:

Lógica de Processamento:

Saídas:

PÓS-CONDIÇÃO

< condição ou condições que deve ser setadas depois da execução da lógica de processamento >

SAÍDAS

Ordem	Nome Físico	Formato	Descrição
#	< nome do dado de saída >	< formato do dado de saída >	< descrição do dado de saída >
...

- Número seqüencial que determina a ordem de saída do dado

ANEXO C – MODELO DE DOCUMENTO DE CASOS DE TESTE

DADOS GERAIS					
Título:	< nome da tarefa >				
Analista:	< nome completo da analista de sistemas >				
Programador:					
Aplicação:	< nome da aplicação e/ou URL de acesso a aplicação >				
Menu/Acesso:	< localização no menu da aplicação >				
Primeira Definição: dd/mm/aaaa	Última Alteração: 2/7/2007	Tempo estimado: X horas	Previsão de conclusão: dd/mm/aaaa	Número da Tarefa no RPM: < Número do projeto/número da tarefa gerados pelo App no momento do cadastro da tarefa >	Identificação Documento Homologação/Itens: <utilizado para tarefas criadas a partir de problemas identificados pela homologação>
Localização dos demais documentos do projeto:	< nome e localização dos documentos do projeto >				

DEFINIÇÃO DE TAREFA			
Nova Funcionalidade <input checked="" type="checkbox"/>	Alteração na aplicação <input type="checkbox"/>	Correção na aplicação <input type="checkbox"/>	Alteração tarefa original <input type="checkbox"/>
<input type="checkbox"/> Revisar Padrões Web nos Templates Indicados no Item Programas			

PROGRAMA(S)	
< nome físico de todos os programas que devem ser testados pelo homologador >	< descrição abreviada do programa >

USUÁRIO		
Código do usuário	Nome de usuário	Senha

CASOS DE TESTE					
Ordem	Dados de Entrada	Descrição	Valor de Entrada	Resultado Esperado	Mensagem Esperada
< ordem de execução >	< nome do dado de entrada >	< descrição do dado de entrada >	< valor de entrada para o processamento >	< resultado esperado após processamento >	< mensagem esperada após processamento >
...

SCRIPT PARA TESTES
< scripts que devem ser executados para que o desenvolvedor/homologador possam realizar os testes >

TESTES		
Teste efetuado:	Iniciado em:	Concluído em: