



FATEC - Faculdade de Tecnologia de São Paulo
Departamento de Tecnologia da Informação

TELMA CAROLINA MENDES

ESTUDO DA GARANTIA DE QUALIDADE DE SOFTWARE UTILIZANDO A
METODOLOGIA DE DESENVOLVIMENTO DE SISTEMAS CRIADA POR UMA
INSTITUIÇÃO FINANCEIRA

São Paulo, Dezembro de 2011

TELMA CAROLINA MENDES

cah.0101@gmail.com

ESTUDO DA GARANTIA DE QUALIDADE DE SOFTWARE UTILIZANDO A
METODOLOGIA DE DESENVOLVIMENTO DE SISTEMAS CRIADA POR UMA
INSTITUIÇÃO FINANCEIRA

Monografia submetida como exigência
parcial para a obtenção do Grau de
Tecnólogo em Processamento de Dados
Orientador: Carlos Hideo Arima

São Paulo, Dezembro de 2011.

RESUMO

MENDES, Telma Carolina. *Estudo da garantia de qualidade de software utilizando a metodologia de desenvolvimento de sistemas criada por uma Instituição Financeira*. São Paulo, 2011. Trabalho de Conclusão de Curso (Processamento de Dados) FATEC-SP.

Em busca de softwares de qualidade empresas vem fazendo uso de metodologias no processo de desenvolvimento de software como um meio de garantir a qualidade de software. Afim de garantir a produção de sistemas com qualidade é que uma determinada instituição financeira adotou uma metodologia única para todas as empresas do grupo. A metodologia adotada nasceu dentro da própria empresa, e foi elaborada de tal maneira a atender as necessidades e as expectativas da instituição de obter sistemas confiáveis, eficientes e que não apresentassem um volume grande de problemas em produção. O objetivo deste trabalho é identificar, quais as características de garantia de qualidade de software que podem ser encontradas dentro da Metodologia de Desenvolvimento de Sistemas criada pela Instituição Financeira. Dentro desta pesquisa foi feito o estudo sobre o que é a qualidade, como foi elaborada a metodologia que é utilizada pela Instituição Financeira e do que ela é composta. A partir destes estudos foi realizada uma análise para verificar se a metodologia possuía itens suficientes para garantir um software de qualidade e, quais foram as características de garantia de qualidade de software encontradas, ou não, em cada uma destas fases.

Palavras-chave: qualidade; metodologia; software.

ABSTRACT

MENDES, Telma Carolina. *Estudo da garantia de qualidade de software utilizando a metodologia de desenvolvimento de sistemas criada por uma Instituição Financeira*. São Paulo, 2011. Trabalho de Conclusão de Curso (Processamento de Dados) FATEC-SP.

Looking for software quality companies are making use of methodologies in the process of software development as a means of ensuring software quality. In order to ensure the production of quality systems is that a particular financial institution has adopted a unique methodology to all group of the company. The methodology adopted was born within the company, and is designed in such a way that meets the needs and expectations of the institution to obtain reliable systems, efficient and does not present a large amount of problems in production. The objective of this study is identifying which features of quality assurance of software that can be found within the Systems Development Methodology created by the Financial Institution. Within this research was the study done on what is quality, as the methodology that was developed is used by the financial institution and that it is composed. From these studies, an analysis was performed to verify that the methodology has enough items to ensure the quality software, and what were the characteristics of software quality assurance found or not in each of these stages.

Key terms: quality; methodology; software.

LISTA DE FIGURAS

<i>Figura 1 - Template do Roteiro de Teste Funcional</i>	36
<i>Figura 2 - Imagem da janela do sistema de fluxo de aprovação.</i>	39

LISTA DE TABELAS

Tabela 1 - Quantidade de empresas certificadas pela ISO nos últimos dez anos..... 16

Tabela 2 - Matriz de Qualidade 46

SUMÁRIO

1. INTRODUÇÃO.....	8
1.1. Breve histórico do uso das metodologias para o desenvolvimento de sistemas	8
1.2. Identificação e justificativa do problema.....	10
1.3. Objetivos	11
1.3.1. Objetivo geral	11
1.3.2. Objetivos específicos	11
1.4. Metodologia.....	12
2. QUALIDADE.....	14
2.1. Conceitos de qualidade.....	15
2.2. As diferentes percepções de qualidade	19
2.3. Qualidade do processo	21
2.4. O que é qualidade de software	23
2.5. Como é medida a qualidade de um software.....	24
3. METODOLOGIA.....	26
3.1. A Metodologia de Desenvolvimento de Sistemas utilizada pela instituição financeira como garantia de qualidade de seus sistemas	26
3.2. Expectativas na implantação desta Metodologia de Desenvolvimento de Sistemas	29
3.3 Fases da MDS – Metodologia de Desenvolvimento de Sistemas.....	31
3.3.1 Escopo e Estimativa	31
3.3.2 Análise.....	32
3.3.3 Desenho Físico	35
3.3.4. Construção	36
3.3.5 Teste.....	38
3.3.6 Implantação	41
3.3.7 Gestão de Mudanças do Projeto	43
3.4 Checklist da Metodologia de Desenvolvimento de Sistemas	45
3.4.2 O preenchimento dos itens do checklist	45
4. ESTUDO E IDENTIFICAÇÃO DOS PROCESSOS DE GARANTIA DA QUALIDADE NA METODOLOGIA DE DESENVOLVIMENTO DO SISTEMA CRIADO POR UMA INSTITUIÇÃO FINANCEIRA.....	47
4.1 Matriz de Qualidade	48
4.2 Estudo das fases da Metodologia de Desenvolvimento de Sistemas criada por uma determinada Instituição Financeira.....	51
4.2.1. Escopo e Estimativa.....	51
4.2.2 Análise.....	53
4.2.3 Desenho Físico	55
4.2.5 Teste.....	56
4.2.4 Implantação	57
4.2.5 Gestão de Mudanças	57
5. CONCLUSÃO.....	59
REFERÊNCIAS	60

1. INTRODUÇÃO

1.1. Breve histórico do uso das metodologias para o desenvolvimento de sistemas

Segundo PFLEEGER (2004), um método é um procedimento formal, que utiliza técnicas e pode também usar ferramentas para produzir algum resultado. A utilização de uma ferramenta em um método pode trazer resultados mais precisos, melhorando a qualidade do produto resultante.

YOURDON (1991) define metodologia como uma série recomendada de passos e procedimentos que devem ser seguidos para obter-se o desenvolvimento de um sistema de informação. E, ainda segundo a definição de AVISON e FITZGERALD (1997), é o conjunto formado por procedimentos, técnicas, ferramentas e documentação que auxiliará os responsáveis pelo desenvolvimento de sistemas em seus esforços na implementação de um novo sistema de informação. Uma metodologia consistirá de fases, cada uma consistindo de subfases, que orientarão estes responsáveis na escolha das técnicas que deverão ser mais apropriadas a cada estágio do projeto e também auxiliá-los a planejar, gerenciar, controlar e avaliar o projeto do sistema de informação.

As metodologias foram criadas a partir da necessidade de se obter um produto final com um mínimo de qualidade. A Metodologia Kaizen trata justamente de fornecer técnicas para que um produto passe a ter qualidade com custo reduzido. O processo utilizado por essa metodologia envolve desde um funcionário de nível hierárquico baixo até a gerência, onde o foco principal da metodologia é o comprometimento de todos para se atingir o objetivo sugerido. Segundo o criador da metodologia, Masaaki Imai: “Os executivos ocidentais tendem a acreditar que melhorias substanciais custam muito dinheiro. Parecem pensar que devem ter o estado da arte da tecnologia e os equipamentos mais avançados para melhorar.”, quando na realidade um produto de qualidade não está relacionado a uma matéria-prima de altíssima qualidade ou a uma mão de obra especializada, o que é

realmente essencial são os esforços de melhoria contínua, executados por todos, sendo que o seu foco central é a busca pela eliminação dos desperdícios.

Nos primórdios do desenvolvimento de sistemas não eram utilizadas as metodologias, o modelo usado dentro das empresas era o "codifica-corrige", onde o desenvolvedor recebia a tarefa de desenvolver um programa. Ele, sem documentar nada, partia para a codificação; se o programa apresentasse erros, era então realizada uma correção e enviado novamente para o cliente. Com o passar dos anos, notou-se a dificuldade para realizar a manutenção nos programas desenvolvidos dessa maneira, uma vez que apenas o programador que o desenvolveu conseguia realizar essa tarefa. Além disso, era necessário também uma análise mais elaborada do problema, pois a comunicação entre desenvolvedor x cliente era praticamente inexistente no então modelo "codifica-corrige".

Na década de 1960, surge a primeira metodologia para desenvolvimento de sistemas com o modelo Waterfall, ou modelo cascata, que sugere um modelo de desenvolvimento sequencial e composto por cinco fases:

- Comunicação: iniciação do projeto e levantamento de requisitos;
- Planejamento: estimativas, cronogramas e monitoramento;
- Modelagem: análise e projeto;
- Construção: codificação e teste;
- E, por fim, a Implantação: entrega e manutenção.

Já na década de 70, Steven Myers e Constantine propuseram os conceitos do projeto estruturado. Buscou-se então aplicar os conceitos de projeto estruturado à análise de sistemas com o objetivo de desenvolver um método de especificação de requisitos adequado ao projeto estruturado.

GANE e SARSON (1979) apresentaram uma metodologia de desenvolvimento de sistemas chamada STRADIS - Structured Analysis, Design and Implementation of Information Systems - que incorporou a análise, projeto e implementação estruturada

de sistemas de informação. É uma metodologia baseada na decomposição funcional e no uso do diagrama de fluxo de dados, composta pelas seguintes fases:

- Fase 1: Estudo inicial;
- Fase 2: Estudo detalhado;
- Fase 3: Definição e projeto de soluções alternativas;
- Fase 4: Projeto Físico.

Gane e Sarson acrescentam que para completar o desenvolvimento do sistema outras atividades são necessárias como, por exemplo, desenhar um planejamento incluindo planos de teste e aceitabilidade do sistema.

E Yourdon (1989), também dentro da metodologia estruturada, propôs a Metodologia Estruturada Moderna, que possui como base a decomposição funcional e o projeto top-down, através dos quais um problema é sucessivamente decomposto em unidades gerenciáveis.

Atualmente, a metodologia mais utilizada é a orientada a objetos. O termo “Programação Orientada a Objetos” foi criado por Alan Kay, autor da linguagem de programação Smalltalk. Mas, mesmo antes da criação do Smalltalk, algumas das ideias da POO já eram aplicadas, sendo que a primeira linguagem a realmente utilizar esses conceitos foi a linguagem Simula 67, criada por Ole Johan Dahl e Kristen Nygaard em 1967. De acordo com DAVID (2007), este paradigma de programação já é bastante antigo, mas só agora vem sendo realmente aceito nas grandes empresas de desenvolvimento de Software. Alguns exemplos de linguagens modernas utilizadas por grandes empresas em todo o mundo que adotaram esses conceitos: Java, C#, C++, Object Pascal (Delphi), Ruby, Python etc.

1.2. Identificação e justificativa do problema

A utilização de software de qualidade garante a segurança das transações, dos negócios, das pessoas envolvidas e mantém alta disponibilidade dos serviços. Produtos e serviços são considerados aceitáveis se apresentarem desempenho dentro de certos limites. Porém, quando um software passa a apresentar grandes falhas, a

empresa responsável pelo software passa a sofrer com despesas de retrabalho para corrigir os erros detectados e, dependendo da gravidade do erro, as horas gastas pelo desenvolvedor para corrigir a falha serão muito maiores do que o tempo que poderia ter sido dedicado à prevenção desta.

Assim como PFLEEGER (2004) afirma que o uso de uma metodologia é uma maneira de obter um software de qualidade, em determinada Instituição Financeira foi implantada uma Metodologia de Desenvolvimento de Sistemas visando maior qualidade em seus sistemas. A instituição adotou uma metodologia interna para o desenvolvimento de novos sistemas e para projetos de melhoria nos já existentes. A medida foi tomada porque a empresa possui uma grande preocupação com falhas sistêmicas, que acarretam em diversos problemas, tais como: perda financeira, comprometimento de dados sigilosos e multas por não estar de acordo com as normas estabelecidas pelo Banco Central.

A partir da implantação de uma metodologia completamente voltada às necessidades da Instituição Financeira é que surgiu o interesse neste estudo sobre como a Metodologia de Desenvolvimento de Sistemas pode garantir a qualidade de um software. Para obter essa resposta foi desenvolvido este trabalho de pesquisa sobre os pontos, dentro do processo proposto de desenvolvimento dentro da metodologia descrita, em que se pode identificar a garantia de qualidade de software.

1.3. Objetivos

1.3.1. Objetivo geral

Estudar os itens relacionados com a garantia da qualidade de software que compõem a Metodologia de Desenvolvimento de Sistemas criada por uma Instituição Financeira.

1.3.2. Objetivos específicos

Realizar um estudo sobre o que é qualidade de acordo com definições de autores que abordam o assunto, e o que é e como é a qualidade de um produto de software.

Realizar um estudo sobre o uso de metodologia como maneira de obter garantia de qualidade de software, como foi elaborada a Metodologia de Desenvolvimento de Sistemas adotada pela Instituição Financeira, e se a maneira como essa Metodologia de Desenvolvimento de Sistemas foi implantada é suficiente para assegurar um software de qualidade.

A escolha da metodologia desta empresa foi feita baseada nas mudanças de estrutura que estão acontecendo dentro da Instituição Financeira, onde cada vez mais a qualidade nos softwares tem sido cobrada de todos não só pela preocupação de um retrabalho, mas principalmente pelos impactos financeiros que a corporação sofre em função de erros sistêmicos.

1.4. Metodologia

A constatação do problema proposto para estudo foi feita a partir de método indutivo, em que, segundo DIEHL (2006), é um método proposto por Bacon, Hobbes, Locke e Hume que considera que o conhecimento é fundamentado na experiência, realizando a observação de casos na realidade concreta.

Para o desenvolvimento do trabalho foi utilizada a pesquisa bibliográfica, buscando autores dentro das áreas de Qualidade e de Engenharia de Software, para que fossem possíveis as fundamentações teóricas do tema abordado. Além da pesquisa por meios bibliográficos foi utilizada também a pesquisa de documentos oficiais, coletando dados secundários, que segundo DIEHL (2006), são dados existentes na forma de arquivos, bancos de dados e relatórios, onde foi possível ser feita a coleta de dados sobre a metodologia desenvolvida e implantada na Instituição Financeira. Foi necessária também uma pesquisa na internet para buscar o conhecimento a partir de opiniões e experiências de outros profissionais da área de tecnologia. Para alguns pontos sobre a utilização da Metodologia de Desenvolvimento de Sistemas implantada na Instituição Financeira foi utilizada também a observação, onde foi possível a percepção de alguns fatos sobre o cenário antes e pós-implantação da metodologia.

Feito o trabalho de levantamento de dados, foi realizada a organização de todos os dados para que fosse então possível a interpretação destes para a exposição do tema abordado, através de textos.

2. QUALIDADE

A preocupação com a qualidade de bens e serviços não é recente, e de acordo com LONGO (1996), a evolução da qualidade pode ser dividida em três grandes momentos: a era da inspeção, da qualidade estatística e da gestão da qualidade. A preocupação dos consumidores de inspecionar os bens e serviços que recebiam deu origem à chamada era da inspeção, que se voltava para o produto acabado, não produzindo assim qualidade, apenas encontrando produtos defeituosos na razão direta da intensidade da inspeção.

De acordo com LONGO (1996), a era do controle estatístico surgiu com o aparecimento da produção em massa, traduzindo-se na introdução de técnicas de amostragem e de outros procedimentos de base estatística, bem como, em termos organizacionais, no aparecimento do setor de controle da qualidade. Sistemas de qualidade foram pensados, esquematizados, melhorados e implantados desde a década de 30 nos Estados Unidos e, um pouco mais tarde, na década de 40, no Japão e em vários outros países do mundo.

E então, a partir da década de 50, surgiu a preocupação com a gestão da qualidade, que trouxe uma nova filosofia gerencial com base no desenvolvimento e na aplicação de conceitos, métodos e técnicas adequados a uma nova realidade. A gestão da qualidade total, como ficou conhecida essa nova filosofia gerencial, marcou o deslocamento da análise do produto ou serviço para a concepção de um sistema de qualidade. A qualidade deixou de ser um aspecto do produto e responsabilidade apenas de um departamento específico e passou a ser um problema da empresa, abrangendo, como tal, todos os aspectos de sua operação.

A qualidade se torna cada vez mais necessária para garantir que um produto seja competitivo, e em todas as áreas há uma procura por ela: na educação, nos produtos que consumimos e até mesmo nos serviços prestados. Este capítulo apresenta, a partir de diferentes conceitos de qualidade, o que essa expressão significa, o que ela representa para o consumidor final e como tornar possível que um produto ou serviço seja considerado de qualidade.

2.1. Conceitos de qualidade

São inúmeros os autores que abordam o tema qualidade, e nem todos possuem o mesmo conceito pra esse termo. Entre alguns dos nomes mais importantes que tratam sobre esse assunto estão Juran, Deming e Crosby.

Segundo JURAN (1974), dentre todas as definições para o termo duas são as mais importantes para o gerenciamento da qualidade:

- As características de um produto, pois, para os clientes, quanto melhores forem as características de um produto, mais alta será a qualidade dele.
- A ausência de defeito, pois, para o cliente, quanto menos defeitos um produto possuir, maior será a qualidade dele.

A partir disso, JURAN (1974) sugere a reflexão dos gerentes para dois pontos: o primeiro é que produtos com melhores características podem afetar as vendas, pois normalmente custam mais caro; e o segundo é que produtos defeituosos podem afetar os custos, uma vez que haverá um retrabalho.

DEMING (1986) sugere que, para obter a qualidade total, 14 pontos devem ser seguidos. Esses 14 pontos desenvolvidos por ele podem ser aplicados das pequenas às grandes empresas, independentemente de seu setor: serviços ou manufatura. Seus 14 pontos sugerem que as transformações devem atingir todas as áreas da empresa, desde os recursos humanos até a produção, atingindo todos os processos organizacionais da empresa. São eles:

- Criar na organização uma constância de propósitos no sentido da melhoria dos produtos e serviços, de forma que se torne competitiva, mantenha-se no negócio e gere trabalho;
- Adotar uma nova filosofia. Vivemos numa nova era econômica. A administração ocidental deve despertar para o desafio, conscientizar-se de suas responsabilidades e assumir a liderança em direção à transformação;

- Acabar com a dependência de inspeção para a obtenção da qualidade. Eliminar a necessidade de inspeção em massa, priorizando a internalização da qualidade do produto;
- Acabar com a prática de negócios compensadores baseados apenas no preço. Em vez disso, minimizar o custo total. Insistir na ideia de um único fornecedor para cada item, desenvolvendo relacionamentos duradouros, baseados em qualidade e em confiança;
- Aperfeiçoar constante e continuamente todo o processo de planejamento, produção e serviços, com o objetivo de aumentar a qualidade e a produtividade e, conseqüentemente, reduzir os custos;
- Fornecer treinamento no local de trabalho;
- Adotar e estabelecer liderança. O objetivo da liderança é ajudar as pessoas a realizar um trabalho melhor. Assim como a liderança dos trabalhadores, a liderança empresarial necessita de uma completa reformulação;
- Eliminar o medo, para que todos possam trabalhar pela empresa;
- Quebrar as barreiras entre departamentos. Os colaboradores dos setores de pesquisa, projetos, vendas, compras ou produção devem trabalhar em equipe, tornando-se capazes de antecipar problemas que possam surgir desde a produção até a utilização dos produtos ou serviços;
- Eliminar slogans, exortações e metas dirigidas aos empregados pedindo por defeito zero e novos níveis de produção. Isso pode gerar uma competitividade negativa entre os funcionários, atrapalhando na produtividade, uma vez que eles podem passar a não mais trabalhar como uma equipe, apenas visando ser melhores que os demais;
- Eliminar a gestão por objetivos com base em indicadores de quantidades. Esse ponto é controverso, a menos que seja interpretado com o entendimento das variações naturais;

- Remover barreiras que despojem as pessoas de orgulho no trabalho. A atenção dos supervisores deve voltar-se para a qualidade e não para números. Remover as barreiras que usurpam dos colaboradores das áreas administrativas e de planejamento o justo direito de orgulhar-se do produto de seu trabalho. Isso significa a abolição das avaliações de desempenho ou de mérito e da administração por objetivos ou por números;
- Estabelecer um programa rigoroso de educação e aperfeiçoamento para todo o pessoal;
- Colocar todos da empresa para trabalharem de modo a realizar a transformação. A transformação é tarefa de todos.

Crosby é o fundador da teoria Zero Defeito. Essa abordagem se baseia na teoria de que a qualidade é assegurada através de um melhoramento contínuo nos processos de fabricação, o que leva a uma realização correta na primeira tentativa. CROSBY (1985) elaborou 14 pontos que constituem as etapas de implementação de sua abordagem. São eles:

- Dedicção da alta gerência e comprometimento através da elaboração de um documento com a política e os objetivos da empresa;
- Constituição de equipes para melhorias coordenadas pelos gerentes;
- Medição dos resultados;
- Avaliação dos custos da qualidade;
- Comunicação dos resultados aos supervisores e operários;
- Reunião para identificação dos problemas;
- Estabelecimento de um comitê informal para a divulgação do programa;
- Treinamento da gerência e supervisão;
- Instauração do dia Zero Defeito, em que os resultados anuais são divulgados e efetua-se o reconhecimento a todos os participantes do programa;

- Estabelecimento dos objetivos a serem seguidos;
- Consulta aos operários sobre a origem dos problemas;
- Recompensar aqueles que atingiram os seus objetivos;
- Formar os conselhos da qualidade;
- Etapa final: faça tudo de novo.

Qualidade é um assunto cada vez mais discutido dentro das corporações e a preocupação em prestar serviços ou apresentar produtos de qualidade é o desafio para quem pretende ganhar mercado. Cada vez mais o mercado de produtos e serviços procura oferecer qualidade aos seus clientes, mas o que é qualidade? Segundo a norma ISO 9001:2008, qualidade é “o grau no qual um conjunto de características inerentes satisfaz a requisitos”. E, nas últimas décadas, muitas empresas, de vários setores, vêm buscando essa qualidade em seus produtos e serviços. A Tabela 1 mostra a quantidade de empresas que tiraram a certificação ISO 9001:2000 e a ISO 9001:2008 nos últimos dez anos:

Ano	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez	Total
2001	40	11	20	40	44	46	22	42	28	48	75	81	497
2002	93	53	110	111	115	129	150	170	155	154	199	385	1824
2003	244	201	242	311	359	335	389	379	361	494	572	886	4773
2004	565	251	291	289	258	217	255	221	260	254	303	460	3624
2005	387	212	226	267	225	168	213	253	220	275	355	462	3263
2006	397	248	373	282	400	247	351	403	350	406	421	561	4439
2007	400	301	449	344	195	332	321	347	263	398	279	343	3972
2008	354	249	354	274	254	262	319	339	336	359	362	483	3945
2009	374	267	360	295	173	167	293	235	225	220	189	519	3317
* 2010	188	130	124	85	63	59	51	36	19	26	0	0	781

*2010 - Até a presente data
Relatório emitido em: 31/10/2010

Tabela 1 - Quantidade de empresas certificadas pela ISO nos últimos dez anos

Segundo CARPINETTI (2010), a ISO identificou que as empresas que procuram a certificação 9001:2008 são ou as que precisam dela ou as que sabem que, ao exibir tal certificação na embalagem de seus produtos, ganham a confiança do cliente por mostrar a ele que o artigo adquirido é de qualidade.

2.2. As diferentes percepções de qualidade

Qualidade possui definições de inúmeros autores, mas para cada indivíduo ela terá um significado diferente, por se tratar de algo que varia de acordo com as necessidades de cada um. É nesse ponto que entra a questão das diferentes percepções de qualidade.

É possível entender melhor essa questão das diferentes percepções com o seguinte exemplo de Weinberg(1994) em que ele descreve um problema com um editor de textos que prejudicou sua sobrinha, pois o editor “perdeu o texto” após um determinado número de palavras digitadas:

[...] Minha sobrinha me pediu para discutir o assunto com a empresa fabricante do software em minha próxima visita”. Localizei então o gerente de projeto para o editor de textos e ele reconheceu a existência do erro.

– É um erro raro – ele argumentou.
– Eu não diria isso – contestei. Encontrei 25 ocorrências no livro de minha sobrinha.

– Mas isso só poderia ocorrer num projeto do tamanho de um livro. Entre mais de cem mil clientes, estimamos que menos de dez produziram um projeto deste tamanho em um único arquivo.

– Mas minha sobrinha notou. Era seu primeiro livro, e ela ficou arruinada.

– Naturalmente eu sinto por ela, mas não faria sentido tentarmos corrigir um erro para dez clientes.

– Por que não? Vocês anunciam que o editor manipula projetos do tamanho de um livro.

– Tentamos fazê-lo, mas as alterações não funcionaram. Eventualmente faremos as correções, mas, por ora, são grandes as chances de introduzirmos um erro pior, que afetaria centenas de milhares de clientes. Acredito que fizemos a coisa certa. Ao ouvir o gerente de projeto, confesso que fiquei dividido. Como consultor da empresa, eu tinha que concordar, mas, como tio de uma autora, coloquei-me radicalmente contra essa linha de raciocínio. Se alguém me perguntasse: “Esse editor é um produto de qualidade?”, eu ficaria com a língua presa. [...]

O caso acima mostra bem como um produto pode ser de qualidade para muitos e para outros, não, uma vez que ele para de atender aos requisitos do cliente.

GARVIN (1984) define qualidade em seu artigo escrito para a Universidade de Harvard com cinco diferentes percepções:

- Transcendente: em que qualidade é sinônimo de "excelência inata". Parte-se da premissa de que há algo de intemporal e duradouro nas obras de qualidade. Está muito próximo aos produtos artesanais, rejeitando a produção em massa. O mais frequente é alegar que a qualidade não pode ser conceituada, nem medida, e nem ser reconhecida exclusivamente pela

experiência. Essa visão gera uma enorme dificuldade, pois não oferece orientação prática alguma. Essa definição se sustenta na ideia de que um gerente ou qualquer pessoa conhece a qualidade quando se defronta com exemplos dela, ou seja, a qualidade só é reconhecida quando é vista.

- Baseada no produto: essa visão é completamente diferente da visão transcendente. A visão baseada no produto sustenta que a qualidade é precisa e mensurável. Porém, sustenta que "diferenças de qualidade refletem diferenças nas quantidades de algum ingrediente ou atributo de um produto". Por exemplo, um sorvete de alta qualidade possui um alto conteúdo de gordura. Sendo assim, forma-se uma dimensão vertical à qualidade, pois o produto pode ser classificado pela quantidade do atributo desejado que ele possui. Dentro desta visão, devem ser destacados dois fatores importantes. O primeiro é que, como a qualidade reflete a quantidade de atributos contidos e como estes envolvem custos, produtos de melhor qualidade serão obrigatoriamente mais caros. O segundo ponto é que a qualidade é vista como característica inerente ao produto e não como algo atribuído a ele. Essa abordagem também apresenta limitações, uma vez que nem sempre existe uma correspondência entre os atributos do produto e a qualidade.
- Baseada no usuário: esta visão fundamenta-se no princípio de que um produto com qualidade é aquele que melhor atende aos desejos ou às necessidades do consumidor. Isso quer dizer que um produto possui maior qualidade à medida que está atendendo às necessidades do público-alvo. Um problema desta abordagem é que ela iguala qualidade à satisfação máxima. Ambas estão intimamente relacionadas, porém não são de forma alguma idênticas. Um exemplo prático disso pode ser uma lista de livros mais vendidos: esses livros são obviamente os preferidos pelos leitores, porém podem não se traduzir no mais alto nível de literatura.
- Baseada na produção: Enquanto a visão baseada no usuário apoia-se na preferência do consumidor, a visão baseada na produção, por sua vez, busca o lado da oferta, interessando-se basicamente pela engenharia e produção do

bem. Todos os conceitos ligados a essa visão se resumem em uma frase: "qualidade é a conformidade com as especificações". Seu enfoque é basicamente interno. Admite o interesse do consumidor por qualidade, mas sustenta que um produto que não está dentro das conformidades não lhe atende às necessidades. Talvez seja essa a sua grande deficiência, abandonar o elo que o consumidor observa entre qualidade e característica do produto, além da conformidade. É cercada de várias técnicas modernas, mas sempre com o intuito de redução de custos.

- Baseada no valor: As definições de qualidade baseadas no valor refletem um avanço das visões anteriores, já que se fundamentam em custos e preços. Consumidores cada vez mais avaliam a qualidade de um produto pelo seu valor. Por exemplo, um restaurante que tem em seu cardápio um prato a R\$ 200,00 por pessoa, independentemente da excelência da sua confecção, pode não ser um produto de qualidade, pois, pelo preço, acaba atraindo poucos compradores. Essa visão envolve dois conceitos distintos - excelência e valor -, mas sem limites definidos, e pode muitas vezes se perder na subjetividade.

Várias percepções de qualidade podem ser insatisfatórias quando aplicadas individualmente em um processo, segundo GARVIN (1984) "As empresas precisam utilizar diferentes perspectivas para atingir um alto nível de qualidade". Por exemplo, não é porque um produto foi fabricado com qualidade atingindo o objetivo de redução de custos da empresa que ele será bem recebido pelos consumidores atendendo todas as suas necessidades.

2.3. Qualidade do processo

De acordo com JURAN (1992), processo é uma série sistemática de ações dirigidas à realização de uma meta. Um processo pode auxiliar no planejamento de qualidade como, por exemplo, os processos de pesquisa de mercado, desenvolvimento do produto, desenvolvimento do processo e de controles do processo.

Existem inúmeras atividades dentro de um processo que podem afetar a qualidade do produto final, caso uma ou mais delas não sejam bem realizadas.

PFLEEGER (2004) considera que a qualidade do processo de desenvolvimento e manutenção é tão importante quanto a qualidade do produto justamente por essa interferência que o processo pode causar no resultado final.

Para que seja possível avaliar um processo para identificar o seu estado atual e em que pontos podem ser melhorados, existem inúmeras abordagens de avaliação de processo, entres elas PRESSMAN (2006) cita:

- SCAMPI (Standard CMMI Assessment Method for Process Improvement), que fornece um modelo de processo de avaliação com cinco passos: a iniciação; o diagnóstico; o estabelecimento; a ação e o aprendizado. Esse método utiliza o CMMI como base para avaliação;
- CBA IPI (CMM-Based Appraisal for Internal Process Improvement), que fornece uma técnica de diagnóstico para avaliar a maturidade relativa de uma organização de software usando o CMM como base para avaliação;
- Norma SPICE (ISSO/IEC 15504), que define um conjunto de requisitos para a avaliação de processo de software. Essa norma tem por objetivo assistir às organizações no desenvolvimento de uma avaliação objetiva da eficácia de qualquer processo definido de software;
- ISO 9001:2000 para software, uma norma genérica que se aplica a qualquer organização que deseje aperfeiçoar a qualidade global dos produtos, sistemas ou serviços que ela fornece, sendo assim a norma é aplicável também a organizações e empresas de software. Essa norma trabalha dentro de um ciclo de planejar, fazer, verificar e agir, que é aplicado aos elementos de gestão de qualidade de um projeto de software. Aplicando esse ciclo ao contexto de software, tem-se que o “planejar” é onde são estabelecidos os objetivos do processo, suas atividades e as tarefas necessárias para a obtenção de softwares de ótima qualidade, resultando na satisfação do cliente. O “fazer” é a implementação do processo. “Verificar” monitora e mede o processo para garantir que os requisitos estabelecidos tenham sido atingidos. Por fim, o “agir” inicia as atividades de melhoramento contínuo do processo.

Todas essas abordagens estão sempre sofrendo atualizações para que seja possível acompanhar as mudanças e as novas necessidades do mercado.

2.4. O que é qualidade de software

Qualidade de software é uma das áreas de conhecimento da engenharia de software cujo objetivo é garantir a qualidade do software através da definição de processos de desenvolvimento. Apesar dos modelos aplicados na garantia da qualidade de software atuar principalmente no processo, o objetivo mais importante é garantir um produto final que satisfaça às expectativas do cliente, dentro daquilo que foi acordado inicialmente.

Segundo SWEBOOK (2004), essa área é o conjunto de atividades relacionadas com garantia de qualidade de software, entre elas as atividades de verificação e validação, onde as particularidades de um modelo de qualidade de software variam para outro modelo, pois cada um tem um número diferente de níveis hierárquicos e um número total de diferentes características.

PRESSMAN (2006) define qualidade como a conformidade com os requisitos funcionais e de desempenho previamente declarados, as normas de desenvolvimento documentadas e as características implícitas que são esperadas de todo software que é desenvolvido profissionalmente. PRESSMAN (2006) acredita que a partir dessa definição é possível enfatizar três pontos importantes:

- É a partir dos requisitos de software que se pode medir a qualidade dele, uma vez que a falta de conformidade com os requisitos é falta de qualidade;
- A definição de normas auxilia no modo como o software é submetido à engenharia, uma vez que as normas se constituem de um conjunto de critérios de desenvolvimento. Se esses critérios não são seguidos, existe uma grande possibilidade de se obter a falta de qualidade;
- Os requisitos implícitos, na maior parte dos casos, não são mencionados, como, por exemplo, o bom desempenho ou a fácil manutenção do sistema, porém são requisitos esperados. Portanto, se o software satisfaz os requisitos explícitos, mas não satisfaz os implícitos, a qualidade do software pode ser considerada suspeita.

Uma vez que são satisfeitos tanto os requisitos explícitos como os implícitos as chances de se obter um software de qualidade são maiores.

2.5. Como é medida a qualidade de um software

Para avaliar a qualidade de um software ou produto é utilizada como apoio a medição. A medição da qualidade de um software pode ser realizada a partir de métricas de qualidade, essas métricas indicam o quanto um software se adapta às exigências implícitas e explícitas do cliente. As métricas podem ser as medidas de: tamanho do produto de software; número de pessoas necessário para implementar um caso de uso; número de defeitos encontrados por fase de desenvolvimento; esforço para a realização de uma tarefa; tempo para a realização de uma tarefa; custo para a realização de uma tarefa; grau de satisfação do cliente. Segundo PRESSMAN (2006), ainda que as métricas de produto para software não sejam sempre absolutas, elas tornam possível avaliar de modo sistemático a qualidade com base em um conjunto de regras claramente definidas.

JURAN (1992) diz que um software de qualidade é aquele que não apenas satisfaz as exigências, mas também é implementado a tempo e de acordo com o orçamento. Quando um software possui muitos erros ou falhas, perde parte do valor para o usuário, porém, um software isento de erros não é, necessariamente, um software de qualidade, pois ele pode se apresentar como um programa isento de erros que, porém, não satisfaz as necessidades do usuário. A ausência de funções necessárias para o usuário não pode ser compensada por funções auxiliares genéricas não solicitadas, como uma calculadora, um bloco de notas, um layout diferenciado ou qualquer outro aplicativo do gênero. E, segundo PRESSMAN (2006), as necessidades do cliente são expressas nos requisitos explícitos descritos ao analista, e também nos implícitos, tais como: ser flexível, fácil de operar, barato, construído no prazo, além de ter que ser mais prático, mais rápido, portátil, realizar integração com outros softwares, ter fácil manuseio, ser mais seguro e mais barato em relação ao processo anterior, fosse ele manual ou já informatizado.

Em PFLEEGER (2004), conta-se que no início da década de 1990 a comunidade da engenharia de software buscava consolidar todas as inúmeras

visões de qualidade em um modelo que pudesse ser utilizado como padrão mundial para medir a qualidade de software. Como resultado disso surgiu a ISO 9126.

A ISO 9126 (2003) define que são características de qualidade os seguintes itens:

- Funcionalidade: É a capacidade do produto de software de prover funções que atendam às necessidades explícitas e implícitas quando o software estiver sendo utilizado sob condições especificadas;
- Confiabilidade: É a capacidade do produto de software de manter um nível de desempenho especificado, quando usado em condições especificadas;
- Usabilidade: É a capacidade do produto de software de ser compreendido, aprendido, operado e atraente ao usuário, quando usado sob condições especificadas;
- Eficiência: É a capacidade do produto de software de apresentar desempenho apropriado, relativo à quantidade de recursos usados, sob condições especificadas;
- Manutenibilidade: É a capacidade do produto de software de ser modificado. As modificações podem incluir correções, melhorias ou adaptações do software devido a mudanças no ambiente e nos seus requisitos ou especificações funcionais;
- Portabilidade: Capacidade do produto de software de ser transferido de um ambiente para outro.

Sendo assim, quando um produto de software atende à todas as especificações acima pode ser considerado um produto de qualidade.

3. METODOLOGIA

Segundo MACENAS (2005), para se produzir um software de qualidade é necessário pensar também em um processo eficiente para que isso ocorra. E para entender, de maneira simples, o que é metodologia, é necessário pensar em procedimentos organizados que conduzem a um produto final de forma padronizada.

Existem atualmente no mercado inúmeras metodologias, onde cada uma representa visões e experiências de seus criadores e, para que seja possível a adoção de uma, algumas vezes é necessário adequar a metodologia à realidade da empresa.

De acordo com MACENAS (2005), a adoção de uma metodologia não implica necessariamente na obtenção de um produto de qualidade, porém não se pode pensar em qualidade sem o apoio de uma metodologia. Ou seja, ainda que a adoção de uma metodologia não garanta que o resultado final de um processo será de qualidade, é ela quem auxilia com um processo de desenvolvimento padronizado e organizado a fim de alcançar o objetivo de obter um produto final de qualidade.

3.1. A Metodologia de Desenvolvimento de Sistemas utilizada pela instituição financeira como garantia de qualidade de seus sistemas

Visando não só a padronização de processos como também a busca por um melhor processo de desenvolvimento, resultando assim em um produto final de qualidade, foi adotada a Metodologia de Desenvolvimento de Sistemas por uma determinada Instituição Financeira. De acordo com GMQS (2009), essa Metodologia de Desenvolvimento de Sistemas criada por uma instituição financeira foi elaborada por 22 representantes de todas as diretorias de desenvolvimento de sistemas, em que o objetivo era o de integrar em uma única metodologia todos os processos de desenvolvimento existentes após a integração de duas instituições financeiras e todas as empresas do grupo resultante. Além disso, com a adoção dela espera-se também:

- Definir as fases de um desenvolvimento de um sistema, de modo a torná-lo repetível;

- Formalizar o envolvimento das demais áreas no processo de desenvolvimento ou manutenção de sistemas. Como exemplo, cita-se o envolvimento da equipe de administração de dados, equipe de arquitetura de TI, equipe de gestão de mudança, equipe de gestão de liberação, etc.;
- Orientar o desenvolvedor no cumprimento de normas corporativas e/ou de controle para mitigação de risco operacional de tecnologia;
- Institucionalizar os processos de desenvolvimento ou manutenção de sistemas, possibilitando a repetição de práticas de sucesso no ciclo de desenvolvimento ou manutenção de sistemas;
- Padronizar a documentação a ser elaborada pelo desenvolvedor, facilitando a manutenção dos sistemas;
- Maximizar a boa comunicação com os clientes através da formalização da solicitação deles, declaração de escopo e requisitos, e aceite pelo cliente durante o processo de desenvolvimento ou manutenção de sistemas;
- Declarar os papéis e responsabilidades no processo de desenvolvimento ou manutenção de sistemas.

Ainda que dentro na documentação elaborada da Metodologia não há referência à nenhuma outra metodologia como base para o desenvolvimento desta, é possível encontrar algumas semelhanças com a metodologia RUP - Rational Unified Process.

A Metodologia de Desenvolvimento de Sistemas desenvolvida pela Instituição Financeira analisada pode ser comparada à Metodologia de Desenvolvimento RUP – Rational Unified Process. Alguns pontos em comum entre essas metodologias são:

- Assim como é descrito em IBM (1998), a metodologia Rational Unified Process fornece uma abordagem disciplinada para atribuir tarefas e responsabilidades dentro de uma organização de desenvolvimento garantindo a produção de alta qualidade. O desenvolvimento da metodologia adotada pela Instituição Financeira foi elaborada sob a mesma abordagem de obter um software

que atenda as necessidades de seus usuários finais, dentro de um cronograma e orçamento previsíveis;

- Em IBM (1998) é descrito que a equipe de desenvolvimento para o Rational Unified Process trabalha em estreita colaboração com clientes, parceiros, produtos da Rational, grupos, bem como a organização Rational do consultor, para garantir que o processo é continuamente atualizado e melhorado para refletir as experiências recentes e em desenvolvimento e as melhores práticas comprovadas. O mesmo ocorre com a Metodologia de Desenvolvimento adotada pela Instituição Financeira, que de acordo com ARQUITETURA (2009) possui a equipe CMPS – Comitê de Melhoria do Processo de Software responsável pela revisão e aprimoramento constante da metodologia, através de reuniões periódicas com os mesmos representantes das diretorias que participaram da criação da metodologia. Essas reuniões acontecem com uma frequência mínima de uma vez ao mês;

- Segundo IBM (1998), o Rational Unified Process aumenta a produtividade da equipe, fornecendo a cada membro da equipe, com fácil acesso a uma base de conhecimento com as diretrizes, modelos e mentores de ferramentas para todas as atividades de desenvolvimento crítico. Por ter todos os membros da equipe acessando a base de conhecimento, não importa se você trabalha com os requisitos, projeto de teste design, gestão ou gerenciamento de configuração, podemos garantir que todos os membros da equipe partilham uma língua comum, o processo e sua visão de como desenvolver software. A metodologia adotada pela Instituição Financeira também possui seu repositório de arquivos, onde é possível que todos os envolvidos no projeto tenham acesso aos documentos e outros artefatos produzidos. O repositório de arquivos utilizados é o MTFS – Microsoft Team Foundation System;

- Segundo IBM (1998) o Rational Unified Process cria e mantém modelos ao invés de focar na produção de grande quantidade de documentos em papel, o processo unificado enfatiza o desenvolvimento e manutenção de modelos, representações semanticamente ricas do sistema de software em desenvolvimento, assim como descrito nas fases da Metodologia de Desenvolvimento de Sistemas em **4.2.2** e **4.2.3**. Para a criação dos modelos da Metodologia utilizada pela Instituição financeira é utilizada a ferramenta Power Designer;

- Em IBM (1998) é descrito que o Rational Unified Process captura muitas das melhores práticas em desenvolvimento de software moderno, de forma que é apropriado para uma ampla gama de projetos e organizações. Da mesma forma foi elaborada a Metodologia de Desenvolvimento de Sistemas utilizada pela Instituição, de acordo com ARQUITETURA (2009) a proposta de metodologia que considerou as necessidades identificadas e considerou os modelos existentes e a experiência dos desenvolvedores da Instituição Financeira no processo.

Dentro da instituição financeira há uma equipe de qualidade que avalia a documentação gerada em cada fase dessa metodologia a fim de garantir que o produto gerado atende à Metodologia de Desenvolvimento de Sistemas através de Checklists, que são listagens que conferem se itens obrigatórios na documentação foram devidamente preenchidos durante as fases da metodologia.

3.2. Expectativas na implantação desta Metodologia de Desenvolvimento de Sistemas

Ao elaborar e implantar essa metodologia na instituição financeira havia uma expectativa de introduzir com sucesso os sistemas, atendendo à Resolução do Banco Central de Nº 2.554 BANCO CENTRAL (1998), que determina que as instituições financeiras e demais instituições regularizadas por ele devem implantar e implementar controles internos, a fim de garantir que os sistemas e as atividades da instituição cumpram normas corporativas ou de controle para diminuir riscos operacionais. Os controles internos devem ser revisados e atualizados periodicamente, de modo a sempre incorporar novos riscos ou riscos antes não identificados.

Com a implantação desta Metodologia de Desenvolvimento de Sistemas, a Instituição Financeira espera que os projetos de TI atendam às expectativas dos demandadores com previsibilidade de escopo ou de correção, custo e prazo. Sendo assim, uma vez que essas expectativas são atendidas, obtém-se um sistema de qualidade.

As três primeiras áreas-foco durante o desenvolvimento dessa metodologia foram:

- Requisitos: Nesta área, o objetivo era obter uma maior clareza dos requisitos, garantindo requisitos explícitos comprometidos entre TI e negócios;
- Desenho da Arquitetura Técnica e Funcional: Nesta área, buscou-se um melhor uso de recursos de desenvolvimento e processamento, garantindo funcionalidades claras entre os sistemas e no uso da tecnologia;
- Modelagem de Dados: Durante a modelagem dos dados, o objetivo era garantir que modelos de dados lógicos e físicos fossem desenhados de maneira explícita, juntamente com os desenvolvedores e alinhados com a produção a fim de se obter a consistência e integridade de informações e dados.

A Instituição Financeira estudada tinha por objetivo também que essa nova metodologia seguisse o modo de trabalho da empresa, sendo constituída então de um processo único que fosse adequado ao tamanho do projeto, simples, mas com alta utilização e, por fim, controlado onde fosse essencial.

Antes da implantação da metodologia foi realizada uma pesquisa de aceitação dos analistas e diretores, identificando a receptividade ou rejeição em relação a artefatos e disciplinas envolvidas nas boas práticas da metodologia, de modo a permitir o esclarecimento de dúvidas, correção de alguns aspectos ou preparar melhor pontos de rejeição.

As fases que compõem a versão final da Metodologia de Desenvolvimento de Sistemas são:

- Escopo e Estimativa;
- Análise;
- Desenho Físico;
- Construção;
- Teste;
- Implantação;
- Gestão de Mudanças do Projeto.

Essas fases são grupos de atividades organizadas em ordem cronológica de execução. Cada fase é composta de atividades que devem ser executadas. Cada documento gerado durante a execução das atividades de uma fase da metodologia passa pela aprovação de determinadas áreas responsáveis para que dessa forma tenha uma garantia de que o proposto no projeto ou mudança está de acordo com as necessidades do cliente.

3.3 Fases da MDS – Metodologia de Desenvolvimento de Sistemas

Segundo a GMQS (2009), as fases de desenvolvimento da metodologia criada são grupos de atividades organizadas em ordem cronológica de execução. Cada uma dessas fases é composta por atividades que são desenvolvidas a partir de um material de apoio que é disponibilizado dentro da intranet da empresa.

Também de acordo com a GMQS (2009), além dos produtos que devem ser resultados de cada fase, há também dentro de cada fase um checklist para que seja analisado se os produtos gerados naquela fase são ou não satisfatórios. Abaixo são descritas cada uma das fases e as atividades que cada uma produz:

3.3.1 Escopo e Estimativa

A GMQS (2009) descreve que o objetivo desta fase é analisar a demanda de modo rápido e com profundidade suficiente para gerar uma estimativa do esforço do projeto. É aberta pelo analista uma solicitação de serviço para a área afetada e então são gerados dois produtos: O DEP – Demanda, Estimativa e Planejamento e o DAP – Direcionamento da Arquitetura ao Projeto.

3.3.1.1 DEP – Demanda, Estimativa e Planejamento

O DEP é o documento de Demanda, Estimativa e Planejamento. Segundo a GMQS (2009), ele é o principal documento do projeto, uma vez que contém as informações de estimativas, desde a ideia até o planejamento da entrega, o detalhamento da demanda, a solução proposta pela área de TI e o planejamento das entregas do projeto.

As informações contidas no documento de Demanda, Estimativa e Planejamento dão uma visão macro das necessidades do solicitante, permitindo avaliar a viabilidade do projeto e estudar uma solução de TI a ser implementada.

3.3.1.2 DAP – Direcionamento da Arquitetura ao Projeto

O DAP é o documento de Direcionamento da Arquitetura ao Projeto, que de acordo com GMQS (2009) é elaborado quando o analista responsável pelo projeto solicita à equipe de Arquitetura de TI uma análise da arquitetura funcional, arquitetura técnica e o direcionamento para o projeto. A equipe de arquitetura é quem determina se há ou não a necessidade de envolvimento no projeto seguindo determinados critérios como:

- Projetos de novas funcionalidades em um domínio funcional;
- Projetos de novas interfaces entre sistemas;
- Projetos de aquisição, alteração de tecnologias ou pacotes em sistemas novos ou legados;
- Projetos de identificação, construção ou reutilização de códigos corporativos.

Caso seja necessário o envolvimento da equipe de Arquitetura, é ela que será responsável pela elaboração do documento de Direcionamento da Arquitetura ao Projeto, sendo que esse documento é emitido apenas depois de um total entendimento do documento da Demanda, Estimativa e Planejamento.

3.3.2 Análise

De acordo com a GMQS (2009), essa fase define os requisitos em detalhes, determinando o que será desenvolvido. A análise é feita após a aprovação da estimativa e do DEP e DAP pelo solicitante da demanda. Essa fase gera até quatro produtos: DRS, PROT, RTF e MLD.

3.3.2.1 DRS – Diagrama de Requisitos de Sistema

O DRS é o artefato de Requisitos de Sistema, que tem por objetivo representar graficamente e em detalhes as funcionalidades do sistema.

Esse documento é constituído por um Diagrama de Funcionalidades do Sistema e a especificação textual de cada funcionalidade, onde o Diagrama deve representar o escopo da solução de TI, identificando o conjunto de todas as funcionalidades do sistema de forma gráfica através do uso de Diagrama de Casos de Uso. A especificação textual descreve textualmente cada uma dessas funcionalidades desenhadas no diagrama.

3.3.2.2 PROT – Protótipo

O Protótipo de Interface permite que a partir de imagens dos layouts, arquivos e gráficos, os envolvidos no projeto entendam e possam sugerir mudanças para que o sistema satisfaça tanto visualmente como em relação às funcionalidades. Além das imagens de tela da solução proposta para atender às funcionalidades esperadas para o sistema na análise, é válida também a elaboração de um protótipo executável das telas desenvolvidas.

3.3.2.3 RTF – Roteiro de Teste Funcional

O Roteiro de Teste Funcional tem por objetivo testar os requisitos funcionais e não funcionais, e é necessário detalhar exatamente o que será testado para as funcionalidades do sistema. Este artefato é composto por:

- Roteiro Geral: É o documento em que o analista de TI deve descrever os cenários de testes que serão contemplados. O Solicitante Aprovador da área de Negócios vai validar este roteiro e, caso necessário, pode complementar com outros cenários de teste relevantes;
- Roteiro de Segurança: É de responsabilidade do Gestor do Projeto, deve conter um roteiro que atenda as exigências da Área de Segurança e implementa um controle SOX necessário;
- Roteiro BI: É preenchido pelo analista de TI, e é necessário apenas para projetos que contemplem sistemas da área de BI – Business Intelligence;
- Roteiro Campanhas: É preenchido pelo analista de TI, e é necessário apenas para projetos que contemplem sistemas da área de Campanhas;

- Log de Erros: É um documento de controle de uso opcional, voltado somente a registrar os erros detectados durante execução dos ciclos de testes.

A GMQS (2009) define que o Roteiro de Teste Funcional deve ser elaborado pelo Analista de Sistemas com a colaboração do Gestor do Projeto durante a Fase de Análise, lembrando que este será utilizado tanto para testar a aplicação com foco em funcionalidades, como também no momento de homologação pelo usuário/solicitante. Durante a elaboração do Roteiro de Teste Funcional deve se atentar muito à linguagem utilizada, pois não deve possuir uma linguagem muito técnica ou conteúdo muito irrelevante que possa dificultar o seu entendimento. Esse documento deve descrever condições de teste, dados de entrada e resultados esperados e as condições ou cenários de teste para cada funcionalidade, visando assegurar o seu correto funcionamento, e deve atingir um nível de representação suficiente para que o artefato deixe claro o objetivo dos testes e de que forma eles serão conduzidos.

Para cada cenário descrito no roteiro devem ser definidos também os dados de entrada a serem utilizados e os respectivos resultados a serem obtidos em cada um deles.

3.3.2.4 MLD – Modelo Lógico de Dados

O Modelo Lógico de Dados representa graficamente e documenta os dados tratados em um sistema de informação. Ele é feito antes do início da fase de Desenho Físico da Metodologia de Desenvolvimento de Sistemas, que é quando serão analisadas as formas de armazenamento e acesso.

Segundo a GMQS (2009), para o desenvolvimento do Modelo Lógico de Dados, a ferramenta homologada é o Power Designer, e o documento é elaborado utilizando conceitos de Modelagem Entidade Relacionamento, como chave primária, normalização, entre outros. A partir do Modelo Lógico de Dados pode-se obter o Modelo Físico de Dados para diversos Sistemas Gerenciadores de Banco de Dados.

3.3.3 Desenho Físico

De acordo com GMQS (2009), nesta fase é definida a estrutura do aplicativo, a dinâmica de execução dos componentes e a lógica algorítmica detalhada de cada componente. Para a elaboração deste Desenho físico podem ser gerados como principais produtos: MFD, DFIS, MAL, RTI.

3.3.3.1 DFIS – Desenho Físico

O documento do Desenho Físico é composto por um conjunto de diagramas e suas especificações. Os desenhos físicos são realizados de acordo com a arquitetura do projeto, podendo envolver tanto soluções de alta quanto de baixa plataforma. O diagrama é a representação visual da solução, podendo utilizar padrão UML ou estruturado, e a especificação é a descrição textual detalhada dos elementos do próprio diagrama. Essa documentação serve de apoio para implementação do código, portanto, tanto a diagramação quanto a especificação deverão ser elaboradas de maneira esclarecedora e detalhar o que deverá ser construído.

Para se certificar da qualidade do documento, é aconselhado como uma boa prática que um membro da equipe do projeto revise os documentos físicos elaborados por outro membro. É mais comum que outra pessoa identifique falhas ou inconsistências com mais facilidade que aquele que elaborou o desenho físico da aplicação.

3.3.3.2 RTI – Roteiro de Teste Integrado

O Roteiro de Teste Integrado tem como objetivo planejar os testes de integração de modo a garantir que todos os componentes do sistema, quando colocados juntos em funcionamento, executem suas funções corretamente, considerando suas interfaces com usuários ou outros sistemas. O Teste Integrado visa identificar e corrigir possíveis problemas na integração. Este documento deve ser elaborado com a lista das condições ou situações que devem ser testadas para validar a integração entre os elementos que fazem parte do sistema, podendo haver uma ordem em que devem ser executadas. O planejamento dos Testes Integrados deve ser feito na medida em que o sistema vai sendo construído, considerando

requisitos, o uso de determinados recursos e o desempenho da aplicação, suas interfaces com usuário e outros sistemas. E todas as funcionalidades, testadas ou não, devem ser retestadas, considerando a integração com outros módulos ou componentes, com outros recursos e usuários. Os itens contidos no RTI devem validar as integrações entre programas, rotinas e sistemas. Geralmente os tipos de falhas encontradas nos testes de integração são de envio e recebimento de dados.

3.3.3.3. MFD – Modelo Físico de Dados

O Modelo Físico de Dados representa a estrutura física das tabelas do sistema e os relacionamentos entre elas, considerando os requisitos de desempenho e eficiência.

Ele é construído também na ferramenta PowerDesigner, a partir do Modelo Lógico de Dados.

Quando ocorre uma alteração no Modelo Lógico, como desnormalização de tabelas, definição de índices e views, e criação de stored procedures, ela deve refletir no Modelo Físico de Dados.

3.3.3.4 MAL – Mapa de Acesso Lógico

O MAL - Mapa de Acesso Lógico é o registro dos acessos com suas volumetrias, que irão atender as necessidades de informações e, ou atualizações, de Bases de Dados para as funcionalidades de um projeto. É através da análise dos acessos, que o administrador do banco de dados poderá definir a criação de índices secundários, particionamento de bases e recomendar ajustes nas funções para garantia de performance.

3.3.4. Construção

Segundo a GMQS (2009), nesta fase é construída a solução conforme definida no desenho físico, testando-a de modo unitário e integrado. Para que seja realizada a construção, é necessário que a demanda tenha sido aprovada pela Fábrica e que o código esteja testado (teste unitário e integrado). As principais atividades desta fase são a construção da base de dados, o recebimento da

documentação, elaboração e execução do RTU, implementação do código fonte, preparação do ambiente de integração e a execução dos testes integrados.

3.3.4.1 Construir a Base de Dados

A finalidade desta atividade é complementar a especificação da Estrutura de Dados e implementar o resultado no Banco de Dados.

Este complemento da especificação contempla o acréscimo de atributos físicos na especificação das tabelas que contribuam com a otimização do banco de dados, sem, contudo, alterar a estruturação de dados.

3.3.4.2 Receber toda a documentação

O responsável pelo projeto recebe toda a documentação para validar os documentos necessários ao envio do projeto à fábrica de software ou desenvolvimento interno.

3.3.4.3 Elaboração e execução do RTU

Os testes unitários fazem parte do ciclo de desenvolvimento de sistemas, especificamente da codificação da aplicação, e concentram-se na verificação da menor unidade do sistema.

A finalidade do Roteiro de Testes Unitários é planejar e definir as condições de testes unitários a serem executadas para a validação dos sistemas que serão construídos ou alterados. Embora os testes de unidade consumam tempo do desenvolvedor, sua adoção no ciclo de desenvolvimento pode trazer benefícios como encontrar defeitos nas fases iniciais da construção do sistema, considerando que corrigi-los é mais fácil e barato. Garantir que todo código de um sistema passe por testes unitários antes de ser liberado reduz a probabilidade da ocorrência de erros em ambiente de produção. Depois de cada pequena mudança, os testes unitários podem ser reexecutados para confirmar se esta mudança provocou o surgimento de um defeito.

3.3.4.4 Implementar Código Fonte

O Código Fonte deve ser implementado pela Fábrica ou internamente, de acordo com as especificações feitas. Os padrões de desenvolvimento variam de acordo com a linguagem que será utilizada para implementar a solução.

3.3.4.5 Preparação do ambiente de integração.

Nesta atividade deve ser definida a infraestrutura de hardware, software e as configurações de ambiente onde ocorrerá a homologação do sistema.

Cada configuração de ambiente de teste especifica uma configuração adequada e controlada, necessária para a condução dos testes e avaliação dos resultados. Prover um ambiente controlado significa garantir que os resultados do esforço de teste sejam exatos, válidos e com uma grande probabilidade de serem reproduzidos em outras execuções, ou seja, é importante para uma eficiente análise e resolução de falhas.

3.3.4.6 Executar o Roteiro de Testes Integrados

São realizados os testes integrados definidos no roteiro a fim de validar se as unidades executam corretamente quando colocadas juntas.

Os Testes de Integração devem ser executados várias vezes até que todo o sistema tenha sido integrado com sucesso. Ao final desta atividade, os sistemas ou módulos construídos e alterados devem estar completamente validados, não apresentando erros e inconformidades quanto a sua integração.

Garantir que todo código de um sistema passe por testes integrados reduz a probabilidade da ocorrência de erros de comunicação entre diferentes partes (módulos ou componentes) em ambiente de produção. Os defeitos identificados devem ser corrigidos, sem a necessidade de relato formal de erros encontrados, dando flexibilidade e rapidez ao processo de construção.

3.3.5 Teste

A GMQS (2009) descreve essa fase como testar funcionalmente a solução desenvolvida validando se os requisitos solicitados foram adequadamente

implementados de modo a garantir a qualidade em ambiente de produção. Nesta fase se obtém do demandador, que é o responsável pela homologação, faz a aprovação, ou não, da implantação do Sistema. Serão geradas as evidências de teste e o TRHI – Termo de Responsabilidade de Homologação e Implantação.

Para a realização dos testes é preciso garantir que o ambiente de homologação dispõe da estrutura de hardware e software necessária descrita no desenho físico. Feito isso, é então executado o Roteiro de Teste Funcional, a fim de validar se os sistemas construídos possuem algum erro, adquirindo confiança e reduzindo riscos na implantação do sistema. Ao executar os testes, dentro do próprio roteiro deve ser preenchida a coluna de execução de testes como "sim ok" ou "não ok". O preenchimento destas informações poderá ser anexado como evidência de testes.

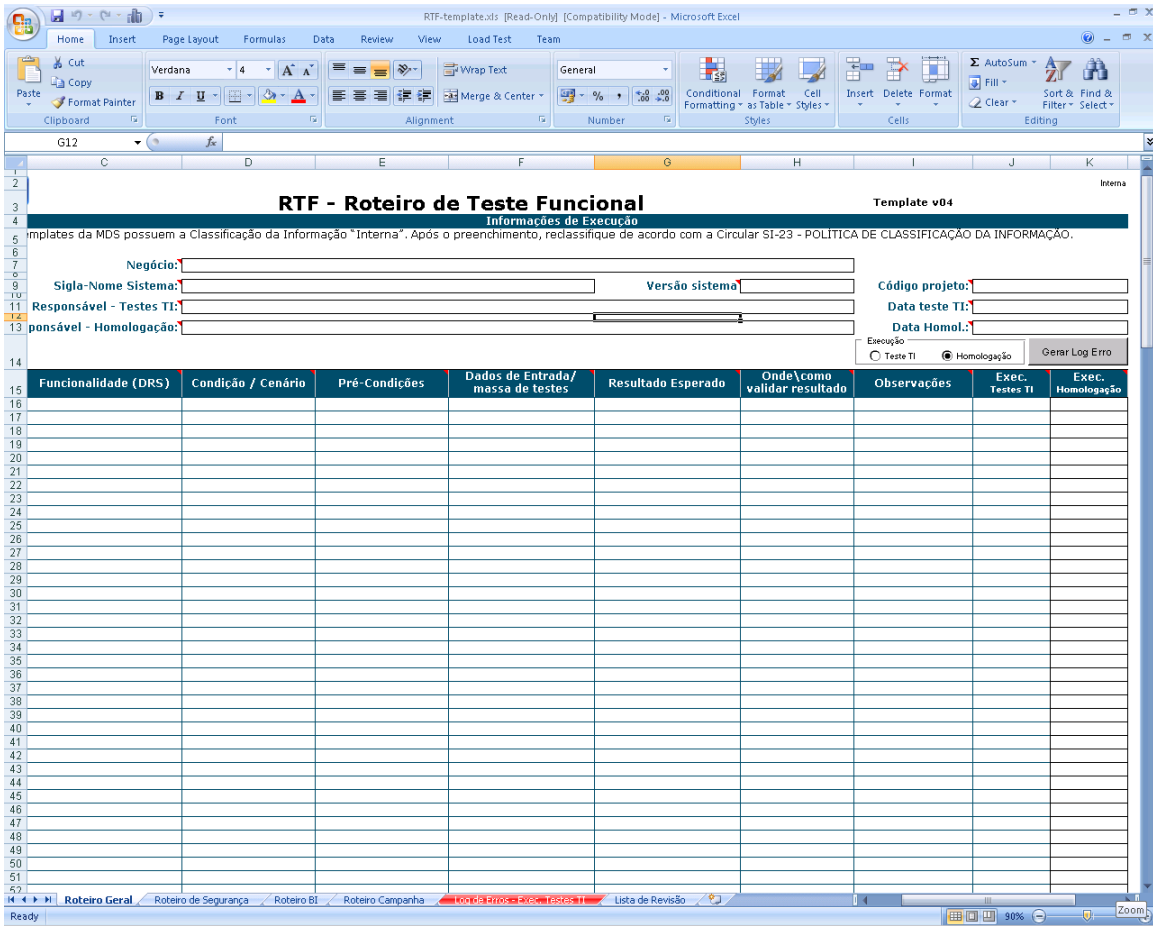


Figura 1 - Template do Roteiro de Teste Funcional.

São executados também testes não funcionais, para garantir que o sistema completo funcione como planejado, atendendo a todos os requisitos não funcionais. Os testes não funcionais verificam confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade do sistema, conforme necessidade do projeto. A performance deve ser testada a fim de certificar que o desempenho, os tempos de resposta e a utilização de recursos por um sistema ou programa atendam aos objetivos de performance definidos no início do projeto. A identificação de gargalos de performance e eliminação destes através de ajustes na aplicação e/ou infraestrutura é conhecida como engenharia de performance e compreende um grande conjunto de práticas que envolvem:

- Monitoração de recursos computacionais e rede;
- Simulação de carga de utilização;
- Medição de tempo de resposta para o usuário;
- Perfil de performance de funções e métodos da aplicação que indiquem alterações estruturais na solução.

A aplicação é então liberada para homologação, que é realizada pelo solicitante. O teste de homologação é a última ação de teste antes da implantação do sistema em produção. O objetivo dos testes de homologação é assegurar que as aplicações estão de acordo com as necessidades do solicitante, integrar corretamente os processos de negócio e garantir a satisfação do cliente, evitando comprometimento com sistemas inadequados. A aplicação é testada a partir do Roteiro de Teste Funcional, que deve ser preenchido pelo homologador, apontando o resultado de cada teste (se está "ok" ou não na coluna "Exec. Homologacao").

Se os resultados obtidos durante os testes forem satisfatórios para o cliente, ele deve assinar o TRHI - Termo de Responsabilidade de Homologação e Implantação. Esse documento visa formalizar o aceite do solicitante de uma mudança em sistemas antes de sua implantação em produção. Mais do que simplesmente atender necessidades legais de controles internos e riscos operacionais, visa trazer maior transparência e controle ao processo de mudança dos sistemas em produção, e assim reduzir a ocorrência de incidentes causados

pela deficiência no planejamento, comunicação ou controle. Esse termo pode ser assinado apenas por colaboradores cadastrados como aprovador e com cargo de gerente ou superior, Gerente da área de negócios e Gerentes de TI. O TRHI é emitido para cada "entrega" de solicitação de serviço, e é válida para liberar a catalogação de todos os pacotes de componentes associados àquela entrega.

3.3.6 Implantação

Para a implantação em produção, a GMQS (2009) determina que o Analista deve planejar a data em que ocorrerá a implantação, o pacote dos entregáveis e solicitar as aprovações das alçadas competentes. A equipe de Suporte é quem deverá analisar os impactos das implantações que ocorrerão no ambiente produtivo.

Ainda nessa fase é elaborado o Documento de Visão. O Documento de Visão, como o próprio nome diz, descreve a visão que todos os envolvidos têm do projeto, em termos das necessidades e características mais importantes. Pelo fato de conter uma descrição dos requisitos centrais pretendidos, o Documento de Visão proporciona a base contratual para requisitos técnicos mais detalhados e permite a captura de todas as perspectivas que o sistema pode abranger. A visão de um projeto pode ser alterada à medida que aumenta a compreensão dos requisitos, da arquitetura, dos planos e da tecnologia.

O Documento de Visão captura objetivos de alto nível de um esforço de modelagem de negócios. Ele serve como base para o processo de aprovação do projeto. Ele comunica os principais questionamentos relacionados ao projeto e funciona como uma referência com base na qual todas as decisões futuras devem ser validadas.

O Documento de Visão de Negócio será lido pelos gerentes, gestores de produto, profissionais envolvidos na modelagem de negócio e desenvolvedores em geral.

A finalidade dessa atividade é descrever a solução de TI para as necessidades identificadas no DEP - Demanda, Estimativa e Planejamento.

Na implantação também é que se altera o repositório de documentos, atualizando-o com todas as alterações realizadas pelo projeto.

Após a implantação ainda existe um processo de acompanhamento “pós-implantação”, onde o Gestor do Projeto realiza os procedimentos finais de controle de qualidade e registra os resultados, para que o projeto possa ser formalmente encerrado pelo solicitante. O acompanhamento dos produtos implantados é feito pelos analistas dos sistemas envolvidos. Quaisquer ocorrências relacionadas aos produtos do projeto devem ser reportadas ao Gestor do Projeto, que concentrará essas informações para registrar o resultado do projeto. Esse processo de acompanhamento da implantação deve ter duração máxima de 30 dias a fim de garantir que o projeto alcançou seus objetivos de qualidade, e é de responsabilidade do Gestor do Projeto, que deve revisar o PPROJ para garantir que o escopo definido foi atingido pelo projeto e registrar o resultado deste, se foi realizado com sucesso ou fracasso, lições aprendidas e outras informações para uso em projetos futuros. O Gerente do Projeto deve também requisitar ao solicitante o encerramento formal do projeto (encerramento da Solicitação de Serviço).

Para realizar o encerramento da Solicitação de Serviço, o solicitante deve garantir que a implantação efetuada atenda aos requisitos definidos no início do projeto, preencher o formulário de feedback e encerrar a solicitação de serviço, através da ferramenta de criação de Solicitações de Serviços para a TI.

Como aprovar a Implantação (TRHI) na SSTI
Quando: Após a homologação **Quem: Aprovador da Implantação**

Solicitação / **Usuários**

- Criar Solicitação
- Alterar Solicitação
- Entregar Serviços (Homologação)
- Aprovar Implantação (Aceite Serviço)
- Implantar Serviços
- Encerramento de Serviços
- Solicitações Pendentes
- Acompanhamento de Solicitações

Aceite de Serviços

TRHI - Termo de Responsabilidade de Homologação e Implantação

Nº Solicitação	Assunto	Cadastro	Estimativa	Aprovação	Entrega
100500	Teste - solicitação exemplo	04/06/2010	07/06/2010	08/06/2010	08/06/2010

Aceite	Não aceitar	Descrição	HH Prevista	HH Realizado	DI Prev Entrega Homologação	DI Real Entrega	DI Aceite	Status
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Entrega 01 - Xxxxxxxxx xxxxxx	120.00 hs	6.00 hs	11/06/2010	08/06/2010		Entregue

Aceite / Aprovação de Implantação -- Caixa de diálogo "Página da Web"

Aprovação / Aceite do serviço

Não Aceitar o Serviço: Implicará o retorno da SS para a área responsável de TI pela demanda, para que seja feita uma nova entrega e a opção de um novo aceite (aprovação)

Aceitar o Serviço: Implicará na aprovação para implantação do serviço em ambiente de produção. Segue termo de para aprovação (TRHI):

Declaro que o pacote de componentes descritos nesta solicitação está autorizado para iniciar o processo de implantação em ambiente produtivo, uma vez que:

- Foi desenvolvido, alterado e/ou adquirido com base nos processos e artefatos previstos na Metodologia de Desenvolvimento de Sistemas vigente.
- Está em conformidade aos requisitos de negócio/técnicos, regras e parâmetros especificados pelo cliente interno solicitante.
- Foi testado pela área de TI de acordo com os roteiros de teste constantes neste projeto.
- Foi homologado/aprovado para implantação pelo cliente interno.

Deseja prosseguir?

O "Aprovador da Implantação" após a homologação, deverá entrar na ferramenta SSTI e registrar a aprovação:

1. Selecionar menu "Solicitação", opção "Aprovar Implantação (Aceite Serviço)
2. Lista de SS que podem ser dado o aceite será disponibilizada
3. Localizar e "SS e entrega" e executar o Aceite (Aceitar ou não a implantação).

TRHI: Termo de Responsabilidade de Homologação e Implantação

Figura 2 - Imagem da janela do sistema de fluxo de aprovação.

3.3.7 Gestão de Mudanças do Projeto

Segundo a GMQS (2009), todo projeto sofre mudanças e as mudanças são desvios em relação ao planejamento inicial do projeto. Elas ocorrem porque projetos são empreendimentos únicos e, portanto, envolvem um certo grau de incerteza. Alguns itens que a GMQS (2009) cita que podem levar a uma mudança são:

- Atividades que são necessárias, mas que não conhecíamos no momento do planejamento;
- Oportunidades de melhoria de um produto;
- Necessidade de esforço extra para que seja atingida uma restrição de data, entre outros.

A solicitação de mudança pode ser realizada por qualquer um que esteja envolvido no projeto.

Ainda de acordo com a GMQS (2009), as melhores práticas de gestão de projetos difundidas no mercado estabelecem que as mudanças no projeto devem ser controladas. O processo de mudança contempla as seguintes etapas: registro, análise de impacto, determinação das alternativas de solução, aprovação ou rejeição e, finalmente, o replanejamento do projeto.

Após a aprovação do planejamento da entrega, qualquer mudança no projeto deve passar pelo processo de Gestão de Mudanças para que seja refletida no planejamento. A alteração no projeto deve ser feita através de um documento de SMP - Solicitação de Mudança de Projeto. É através deste documento que o processo é realizado, desde o registro da mudança até o replanejamento feito no DEP e associado a esta SMP.

Para realizar uma mudança, o Solicitante preencherá os itens do documento de Solicitação de Mudança de Projeto: “Informações do Projeto”, “Identificação do Solicitante da Mudança”, “Características”, “Descrição da Mudança” e “Justificativas para Mudança”. Esse documento é enviado para o Gestor do Projeto, que deverá analisar a solicitação. Feito isso, o Gestor deverá distribuir essa solicitação para os envolvidos do projeto impactado. Os envolvidos, que são os analistas de sistemas, analisam a mudança e determinam o impacto em termos de escopo, prazo, esforço e outras dimensões do projeto como qualidade, recursos humanos, comunicação, riscos etc., sugerindo alternativas de solução. Os analistas impactados então enviam suas análises para o Gestor do Projeto, que consolida as informações, propõe alternativas e faz a sua recomendação, através dos itens “Registro de Impactos” e “Opções para Solução, Pontos de Atenção e Recomendação do Gestor do Projeto”. O aprovador da mudança analisa a solicitação e preenche o item “Aprovação da Mudança”. Em caso de aprovação, o DEP deve ser replanejado, registrando-se o SMP e sua aprovação no item “Controle de Aprovações do Planejamento”, e indicando o SMP como evidência da aprovação.

A GMQS (2009) determina que as entregas devem ser replanejadas, em que todos os envolvidos no projeto devem ser avisados sobre o replanejamento. Para que seja seguido, o novo planejamento é então enviado pelo Gestor do Projeto aos envolvidos no projeto.

3.4 Checklist da Metodologia de Desenvolvimento de Sistemas

O Checklist da Metodologia de Desenvolvimento de Sistemas é um instrumento a ser utilizado como ferramenta de apoio na avaliação da qualidade dos documentos gerados nos projetos que utilizam a Metodologia de Desenvolvimento de Sistemas. Cabe ao avaliador de um documento da Metodologia de Desenvolvimento de Sistemas utilizar o checklist para registrar formalmente as não conformidades identificadas que serão repassadas ao autor. Cada documento gerado durante as fases propostas pela metodologia possui seu próprio checklist.

Os checklists também podem ser utilizados pelo autor como apoio para orientar a correta preparação de um documento. Os checklists de apoio visam principalmente orientar procedimentos para envolvimento de outras áreas e conferir a execução de atividades operacionais durante o processo de desenvolvimento de sistemas.

3.4.1 O preenchimento dos itens do checklist

A GMQS (2009) define que cada item do checklist deve ser respondido com uma das seguintes opções: SIM, Não ou N.A (Não Aplicável):

- SIM: o conteúdo do item do documento atende completamente ao questionamento;
- NÃO: o conteúdo do item do documento não atende completamente ao questionamento. Na coluna “Comentário” pode ser detalhado o que está em desacordo;
- N.A (Não Aplicável) : o questionamento não é aplicável ao contexto do projeto/sistema, podendo referir-se a item opcional do documento. Esta opção de resposta não deve ser utilizada para itens obrigatórios dos documentos da MDS.

O checklist deve ser preenchido ao término da elaboração do documento ou após a entrega do mesmo, para verificar se seu conteúdo segue as diretrizes da Metodologia de Desenvolvimento de Sistemas proposta pela Instituição Financeira.

Na validação de documento recebido de outra área, para verificar a qualidade do conteúdo, é enviado o checklist respondido para o autor como retorno da avaliação, orientando-o sobre os itens a adequar.

.

4. ESTUDO E IDENTIFICAÇÃO DOS PROCESSOS DE GARANTIA DA QUALIDADE NA METODOLOGIA DE DESENVOLVIMENTO DO SISTEMA CRIADO POR UMA INSTITUIÇÃO FINANCEIRA

Existem alguns requisitos que caracterizam um software de qualidade, são eles: Funcionalidade, Confiabilidade, Usabilidade, Eficiência, Manutenibilidade e Portabilidade, conforme citado em 2.5. Com o propósito de alcançar esses requisitos, a Instituição Financeira estudada optou por melhorar seus processos para desenvolvimento de sistemas e pela adoção de uma Metodologia de Desenvolvimento de Sistemas única para todas as empresas do grupo e adequada às necessidades internas.

Essa metodologia teve sua origem interna e foi completamente moldada de acordo com as normas internas da empresa, e com processos envolvendo mais de uma equipe para o desenvolvimento, quando julgado que isso auxiliaria na obtenção de uma maior qualidade.

O estudo que segue sobre a garantia da qualidade de software na Metodologia de Desenvolvimento de Sistemas criada por uma Instituição Financeira foi elaborado a partir da análise da documentação sobre a metodologia, em que foi identificado dentro de cada fase da metodologia características de garantia de qualidade de software nos requisitos necessários dentro da elaboração de cada artefato.

Para identificar características da garantia de qualidade de software dentro da Metodologia de Desenvolvimento de Sistemas estudada foi necessário realizar um estudo sobre como são executadas as fases da metodologia e do que são compostas. Foi identificado que cada fase desta metodologia é composta de atividades que resultam em uma documentação composta de documentos de apoio à implementação da solução, roteiros para a realização de testes funcionais e não

funcionais e documentos de evidência de testes para comprovar o comportamento do sistema.

Para chegar nos resultados do estudo foi necessária a análise de cada template existente para uma determinada fase, analisando o conteúdo de cada um para concluir quais itens de garantia de qualidade de software poderiam ser identificados neles. Todos estes resultados foram representados em uma matriz para que fossem organizados em uma visão simplificada.

4.1 Matriz de Qualidade

Durante o estudo das características dentro desta metodologia, foi necessário analisar os itens que levariam a um produto final de qualidade. Para auxiliar neste trabalho, foi elaborada uma matriz que representa quais são esses aspectos dentro de cada fase da metodologia.

Nesta matriz é possível ver, conforme a Tabela 2, uma coluna que contém todas as fases que a Metodologia de Desenvolvimento de Sistemas possui e uma linha com as especificações de qualidade de software. Para as características que foram identificadas, foi marcado o item referente a elas dentro das especificações de qualidade, e as que não foram reconhecidas permaneceram em branco.

		Especificações de Qualidade					
		Funcionalidade	Confiabilidade	Usabilidade	Eficiência	Manutenibilidade	Portabilidade
Fases da Metodologia	Escopo e Estimativa	✓	✓		✓		✓
	Análise	✓	✓	✓	✓		
	Desenho Físico		✓		✓	✓	✓
	Construção		✓		✓		✓
	Teste	✓	✓	✓	✓	✓	✓
	Implantação	✓	✓	✓	✓	✓	✓
	Gerência de Mudanças	✓	✓	✓	✓	✓	✓

Tabela 2 – Matriz de Qualidade.

Durante pesquisa foram observados os seguintes pontos: requisitos, a implementação da solução, a execução dos testes e como é realizada a implantação do sistema. Segundo PFLEEGER (2004), esses pontos podem ser descritos da seguinte forma:

- Requisitos: São as características do sistema ou a descrição de algo que o sistema é capaz de realizar para atingir seus objetivos;
- Implementação da Solução: É a fase de construção do projeto, em que são escritos os programas que implementarão o sistema. A implementação é feita a partir das informações levantadas nas fases de requisitos;
- Execução de Testes: Durante a execução de testes são efetuados os testes previamente definidos. Esses testes servem para verificar defeitos e falhas no sistema relacionados a codificação e ambiente desenvolvido, avaliar a confiabilidade, disponibilidade, funcionalidade, usabilidade e manutenibilidade do sistema;
- Implantação do Sistema: É o momento em que o sistema é entregue ao usuário para que ele possa avaliar e aprovar a sua implantação em ambiente de produção. Essa entrega deve ser acompanhada de um treinamento para

que o cliente esteja apto a manusear o sistema.

Pelo menos um item da especificação de qualidade foi localizado em cada fase, exceto em alguns casos, conforme citado nos próximos parágrafos. Essas exceções, apesar de terem características de garantia de qualidade de software, não possuem alguns itens, que, de acordo com autores de engenharia de software, são necessários em um processo de desenvolvimento de software.

De acordo com PFLEEGER (2004), em um primeiro momento é necessário estabelecer um comum acordo entre o analista e o solicitante da demanda sobre o que o sistema irá fazer. Dentro da Metodologia de Desenvolvimento estudada, isso ocorre dentro da fase de Escopo e Estimativa, é nela que o solicitante descreve suas necessidades e de acordo com isso o analista propõe alternativas para o solicitante, para que ambas as áreas entrem em acordo sobre a melhor solução.

Nas fases seguintes, de Análise e Desenho Físico, a identificação de requisitos continua sendo trabalhada com a especificação de requisitos que descrevem tecnicamente a solução proposta através de requisitos funcionais e não funcionais. De acordo com PFLEEGER (2004), requisitos funcionais são os que descrevem como o sistema deveria se comportar considerando um certo estímulo, e os não funcionais são a descrição das restrições, como, por exemplo, qual linguagem de programação será utilizada, em que tipo de hardware será implantado e qual a estrutura de dados envolvida.

Um item que não é contemplado na fase de análise, mas deveria, é o fator humano. Durante o desenvolvimento dessas três fases são citados os dados sobre ambiente de implantação, interfaces, funcionalidades, segurança, volume de dados e documentação, mas o fator humano foi esquecido. Segundo PFLEEGER (2004), usuários e fatores humanos também devem ser considerados nos requisitos com o propósito de um melhor entendimento sobre quem utilizará o sistema, os tipos de usuários que utilizarão o sistema, o treinamento que deverá ser desenvolvido e problemas que podem afetar o usuário caso ele utilize o sistema de maneira inadequada.

Na fase de implementação da solução é feita a validação da documentação que é enviada para a equipe de desenvolvimento, porém não há a exigência de uma documentação do código, a preocupação com esse item se resume ao teste unitário. Porém, quando um código não possui documentação interna, pode ocasionar a dificuldade em sua manutenibilidade, pois apesar de sua documentação externa ser completa, uma pessoa que não conhece o sistema e é incumbida de realizar a manutenção nele precisará de tempo até encontrar o ponto exato onde precisa realizar a manutenção. Por isso a fase de construção da metodologia não possui relação com o item de manutenibilidade, pois, apesar de existir toda a documentação externa que auxiliará em uma possível manutenção, o item não contempla uma documentação interna do sistema.

Uma falha que foi possível notar na fase de implantação é que não há nenhuma atividade de elaboração de um manual do sistema para o usuário. Mesmo que tenha sido dado o treinamento, é necessário que o usuário tenha um manual que possa ser utilizado como um guia de consulta, evitando assim ter que recorrer ao analista responsável pelo sistema toda vez que surgir uma dúvida sobre a utilização deste.

4.2 Estudo das fases da Metodologia de Desenvolvimento de Sistemas criada por uma determinada Instituição Financeira

Neste tópico é apresentado como foi feito o desmembramento de cada fase, descrevendo os pontos que identificam, dentro de cada uma, quais itens de qualidade de software poderiam ser associados. O tópico foi subdividido de acordo com as sete fases que compõem a metodologia: Escopo e Estimativa, Análise, Desenho Físico, Construção, Teste, Implantação e Gerência de Mudanças. Em cada fase, foram identificados pontos dentro de suas atividades que se relacionavam com algum item de qualidade.

4.2.1. Escopo e Estimativa

Essa é primeira fase da metodologia. Para ela são realizadas duas atividades: Demanda, Estimativa e Projeto, e a atividade de Direcionamento da Arquitetura do Projeto. Essa fase trata da comunicação inicial entre o solicitante e o analista, para

estabelecer qual o problema e quais as soluções iniciais para ele, tais como em quais tecnologias será desenvolvida a solução e com quais ferramentas, a fim de garantir que a solução estará apta a receber o volume de dados com os quais o solicitante trabalha diariamente.

DEP – Demanda, Estimativa e Projeto

Dentro deste documento, o solicitante do projeto preenche o detalhamento da demanda, ou seja, é onde o solicitante descreve suas necessidades para o projeto e também as regras de negócios necessárias para que essas funções possam ser desenvolvidas. Estes dados são enviados para o analista do projeto, que é quem propõe uma solução tecnológica que atenda aos requisitos descritos. É possível notar neste documento o item de qualidade relativo à funcionalidade, uma vez que o solicitante descreve as funcionalidades que ele precisa encontrar no sistema, e o analista propõe uma solução para que possa implantar essas funções.

DAP – Direcionamento da Arquitetura ao Projeto

Este documento é preenchido pela equipe de arquitetura de sistemas a partir do DEP – Demanda, Estimativa e Projeto, e contém um melhor detalhamento técnico e funcional do projeto. Esse documento é composto por:

- Objetivo do Projeto: é o item que contém a descrição, de um modo geral, dos objetivos do projeto, os problemas que deram origem ao projeto, as necessidades do solicitante e as metas para ele;
- Situação Atual: é o item que contém a descrição do cenário atual, seja um sistema já criado ou um processo manual. Caso seja um sistema já desenvolvido que sofrerá mudanças, são destacados os pontos em que ele será melhorado;
- Solução Proposta: é o item que contém a descrição detalhada da solução dada para o problema que foi exposto pelo solicitante do projeto. Esse item é descrito levando em conta as ferramentas utilizadas e a arquitetura proposta;
- Desenho de Arquitetura: esse item é composto pelas visões funcionais e técnicas da arquitetura para a solução proposta;

- Segurança: é o item que contém as recomendações de uso dos mecanismos corporativos de autenticação;
- Disponibilidade: é o item onde é destacado os casos de contingência e disponibilidade do sistema, como: onde há contingência local, balanceamento, se pode ser parado para backup etc.;
- Escalabilidade: é o item onde é informado se a solução proposta atenderá aos requisitos de escalabilidade, ou seja, se poderá crescer conforme houver aumento da demanda de negócio. Para novas demandas, podem ser realizados testes de carga para verificar se a infraestrutura suportará um possível aumento de acessos;
- Papéis e responsabilidades envolvidos na solução: é o item onde são mostradas quais equipes de suporte estão envolvidas no suporte à solução proposta. A definição da solução proposta está de acordo com as responsabilidades das equipes de suporte, desonerando os analistas e desenvolvedores de encargos de infraestrutura;
- Ferramentas de desenvolvimento e linguagens e Características da solução: são itens onde são descritas as linguagens de programação, as plataformas, tecnologias e bases de dados que suportarão o sistema.

De acordo com todas essas informações fornecidas neste documento, é possível afirmar que o DAP – Direcionamento da Arquitetura ao Projeto -, se encaixa nos itens de especificação de qualidade: Funcionalidade, Confiabilidade, Eficiência e Portabilidade, uma vez que coloca em análise as necessidades do cliente, ao propor uma solução que contenha todas as funções que ele precisa e ao descrever quais tecnologias e recursos serão necessários para que isso possa ser implementado da melhor maneira.

4.2.2 Análise

Nesta fase, a solução proposta já passa a ser moldada através de diagramas e protótipos, e também é elaborado o roteiro de testes que serão executados para verificar se as funcionalidades estão sendo realizadas conforme o esperado. As

atividades desta fase são: Documento de Requisitos de Sistemas, Protótipo, Roteiro de Testes Funcionais e Modelo Lógico de Dados.

DRS – Documento de Requisitos de Sistemas

Caracteriza o item de especificação de qualidade Funcionalidade, pois neste documento todas as funcionalidades do sistema são desenhadas em diagramas e especificadas cada uma delas textualmente.

PROT – Protótipo

O protótipo é um modelo demonstrativo de como será o sistema, a partir da demonstração dele o usuário já consegue perceber os itens de especificação de qualidade Usabilidade e Funcionalidade, pois a sua demonstração já proporciona ao cliente a ideia de como as funcionalidades desejadas serão implantadas, e ele consegue também perceber a dificuldade, ou não, de manusear o sistema.

RTF – Roteiro de Testes Funcionais

Esse roteiro é voltado à descrição dos casos de testes para as funcionalidades do sistema. Ele é composto por roteiro geral, de segurança, de campanha e log de erros. O roteiro geral descreve as funcionalidades que serão testadas, as pré-condições, dados que serão testados, resultados esperados e resultados obtidos. O roteiro de segurança verifica se requisitos de segurança, como, por exemplo, acesso ao sistema e dados restritos, são ou não satisfatórios. O roteiro de campanha é o roteiro em que é avaliado o desempenho e a confiabilidade do sistema, de acordo com dados esperados e dados obtidos. O log de erros contém observações sobre falhas que ocorram em qualquer dos testes executados.

MLD – Modelo Lógico de Dados

O Modelo Lógico de Dados é desenhado a partir de conceitos de Modelagem Entidade Relacionamento, como chave primária e normalização de tabelas. Esses itens estão relacionados à confiabilidade sistêmica e à eficiência que esse modelo terá, pois o modo como esses itens são desenhados pode interferir no desempenho de tabelas e na consistência dos dados contidos nelas.

4.2.3 Desenho Físico

Nesta fase foi possível verificar uma maior preocupação em relação ao desempenho, manutenção e implantação, pois são desenvolvidas atividades como a documentação mais detalhada da solução, com um desenho físico especificado com maiores detalhes através de diagramas que representam todo o fluxo de informações e a descrição destas, a modelagem da base de dados do sistema e roteiro de testes relativos à integração do sistema.

DFIS – Desenho Físico

O desenho físico é utilizado para a implementação do código, por isso deve ser esclarecedor e detalhado. Para que isso seja possível, ele é representado por diagramas e com uma especificação textual. A manutenibilidade do sistema depende diretamente dessa documentação, já que ela deve ser consultada se houver a necessidade de futuras alterações sistêmicas.

RTI – Roteiro de Testes Integrados

Esse roteiro de testes contém a descrição dos casos de testes que devem garantir que o sistema foi integrado com outros sistemas corretamente, sem apresentar problemas. O sistema desenvolvido pode ser integrado com sistemas desenvolvidos em outras tecnologias, o que leva a uma verificação, nesse item, da portabilidade do sistema.

MFD – Modelo Físico de Dados

Esse modelo é desenhado considerando os requisitos de desempenho e eficiência, pois é elaborado através de uma análise da normalização de tabelas, definição de views e índices, e criação de store procedures.

MAL – Mapa de Acesso Lógico

É através deste mapeamento que a performance do sistema é garantido. Ele é que mapeia as informações relativas à volumetria, acessos ao sistema e, quais recursos são mais acessados. Ele mapeia informações também como a do tipo de

processamento realizado, os tipos de acessos que serão permitidos : leitura, inserção, exclusão ou atualização.

4.2.4 Construção

Essa fase de construção, do modo como foi elaborada para a Metodologia de Desenvolvimento de Sistemas da Instituição Financeira, se encaixa nos itens de garantia de software - Confiabilidade, Eficiência e Portabilidade - , já que as atividades envolvidas nessa fase estão relacionadas às características departamentais do sistema. Suas atividades consistem na preparação do ambiente de homologação e implementação do código para a solução desenhada.

Na fase de construção, antes de seu início, caso seja executada por uma equipe terceirizada de fábrica de software, há um responsável técnico pela fábrica que deve analisar toda documentação elaborada até essa fase para avaliar se a documentação é suficiente para a implementação pela fábrica.

Em (GMQS, 2009), é aconselhado que os testes unitários façam parte do ciclo de desenvolvimento do sistema, seja esse feito internamente ou externamente, para que possa auxiliar o desenvolvedor a detectar problemas ao longo da construção ou manutenção do sistema, reduzindo assim erros em ambiente de produção.

Nesta fase é preparado também o ambiente de integração dentro do ambiente de homologação do sistema, de acordo com as especificações da infraestrutura para software e hardware. Feito isso, o Roteiro de Testes Integrados é executado neste ambiente.

4.2.5 Teste

Nessa fase são realizados testes funcionais e não funcionais. É a partir daí que é possível avaliar o todo, verificando se o sistema desenvolvido atende a todas as especificações de qualidade de software: Funcionalidade, Confiabilidade, Usabilidade, Eficiência, Manutenibilidade e Portabilidade.

Para a validação da qualidade nas funcionalidades do sistema é executado o Roteiro de Testes Funcionais, e para os requisitos não funcionais a verificação da qualidade é feita a partir de:

- Monitoração de recursos de rede;
- Simulação com uma carga semelhante à que será utilizada;
- Medição de tempo de resposta;
- Identificação de alterações na estrutura da solução.

Durante esta fase de testes devem ser geradas evidências para anexar a toda documentação como prova de que o sistema funcionou corretamente, de acordo com o solicitado. Se estiver tudo conforme os requisitos durante os testes em ambiente de homologação, o solicitante deve assinar um Termo de Responsabilidade de Homologação e Implantação para que o aceite possa ser formalizado.

4.2.4 Implantação

A fase de implantação é dividida em dois pontos: a implantação e o pós-implantação. Para a implantação é desenvolvido um documento que apresenta uma visão geral do projeto a fim de avaliar a solução da equipe de tecnologia para as necessidades identificadas no DEP – Demanda, Estimativa e Planejamento. Além desta avaliação funcional há uma pós-implantação, em que é avaliado se o projeto como um todo está atendendo aos itens de Confiabilidade, Usabilidade, Eficiência, Manutenibilidade e Portabilidade. Esse período de avaliação de pós-implantação é realizado pelos analistas envolvidos no projeto, e se ocorrer algum problema, o mesmo é reportado para o gestor do projeto, que é quem avaliará as informações para uma tomada de decisão sobre o que fazer.

4.2.5 Gestão de Mudanças

Quando ocorre algum desvio em relação ao planejamento inicial do projeto, como a necessidade de acrescentar uma nova funcionalidade ou realizar a melhoria do produto, é necessária a realização de uma mudança. Para isso deve ser preenchida a SMP – Solicitação de Mudança de Projeto. Neste documento é descrito quem solicitou a mudança e a justificativa para isso, além de conter também os impactos e as soluções propostas.

A gestão de mudanças contemplará todos os itens de qualidade, já que uma mudança envolve a análise de impacto, alternativas de solução e, por fim, um replanejamento do projeto.

5. CONCLUSÃO

A pesquisa sobre o estudo da garantia da qualidade de software dentro da MDS – Metodologia de Desenvolvimento de Sistemas – foi todo baseado nas informações extraídas da documentação e de templates da metodologia. Com o estudo da garantia de qualidade de software utilizando essa metodologia, foi possível analisar como os itens relacionados à garantia de qualidade de software (Funcionalidade, Confiabilidade, Usabilidade, Eficiência, Manutenibilidade e Portabilidade) são aplicados durante cada fase dela.

É possível identificar todos esses itens relacionados à garantia de qualidade dentro da metodologia de modo geral, ainda que nem todas as fases contemplem todos os itens, o que leva à conclusão de que a metodologia, como foi construída, atende aos requisitos necessários para que o software seja considerado um produto de qualidade. Porém, é possível que apenas a adoção de uma metodologia não seja o suficiente para garantir que um software será de qualidade, o fator humano também deve ser considerado.

Quando uma equipe de Gerência de Qualidade não possui pessoas bem treinadas dentro dos processos de qualidade da empresa, ou um número suficiente de pessoas para avaliar todo o volume de documentos gerados durante o desenvolvimento de um projeto, podem ocorrer atrasos nos projetos ou falhas nos documentos podem passar despercebidas. Além disso, o treinamento contínuo sobre a metodologia e todas as ferramentas que estão envolvidas com elas, para os analistas, gestores e todos os demais envolvidos no projeto, deve ser eficiente para garantir que eles estarão aptos a gerar os artefatos necessários.

O uso de uma metodologia para garantir a qualidade de um software é uma opção válida, porém a empresa precisa se estruturar adequadamente para isso, para garantir um número suficiente de funcionários para exercer suas funções e proporcionando a eles um programa de treinamento contínuo e que alcance a todos, e não apenas a um número restrito deles.

REFERÊNCIAS

PAULK, Mark C., et al. **The Capability Maturity Model: Guidelines for Improving the Software Process**. Pearson Education. 20ª ed. 2004.

WEINBERG, Gerald M. **Software com Qualidade: Pensando e idealizando sistemas**. Makron Books. 1ª ed. 1994.

JUNIOR, Isnard Marshall, et al. **Gestão da Qualidade**. Editora FGV. 9ª ed. 2008.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 9126-1: Engenharia de Software – Qualidade de produto – Parte 1: Modelo de qualidade**. Rio de Janeiro: ABNT, 2003.

JURAN, Joseph M.. **Quality control handbook**. McGraw-Hill. 3ªed 1974.

ISHIKAWA, Kaoru. **Controle de qualidade total: à maneira japonesa**. Editora Campus. 1ª ed. 1993.

CARPINETTI, Luiz Cesar Ribeiro, et al. **Gestão da Qualidade ISO 9001:2008 Princípios e Requisitos**. Editora Atlas. 3ªed 2010.

JURAN, Joseph M.. **A Qualidade desde o Projeto**. Cengage Learning, 1992.

GARVIN, David. **What Does “Product Quality Really Mean?”**. Harvard business Review, Fall 1984.

SWEBOK 2004. **Guide to the Software Engineering Body of Knowledge**. 2004. Disponível em <http://www.swebok.org/htmlformat.html> (último acesso 07/2011)

HUMPHREY, W. S., **Managing the software Process**, Reading, Mass., Addison – Wesley, 1989.

AVISON, D.E., FITZGERALD, G. **Information Systems Development Methodologies, Techniques and Tools**. 2 ed. London: McGraw-Hill, 1997.

YOURDON, EDWARD. **Modern Structered Analysis**. Prentice-Hall, 1991.

PFLEEGER, SHARI L. **Engenharia de software Teoria e Prática**. Pearson Prentice-Hall, 2ªEd 2004.

GANE, C., SARSON, T. **Structured Systems Analysis: Tools and Techniques**. New Jersey: Prentice-Hall, 1979.

DAVID, MARCIO FRAYZE. Artigo sobre Orientação à Objetos. 2007.

Disponível em <http://www.hardware.com.br/artigos/programacao-orientada-objetos> (último acesso 09/2011)

PRESSMAN, ROGER S., **Engenharia de Software**, São Paulo, Ed. McGrawHill, 6ª Ed, 2006.

MECENAS, Ivan, et al, **Qualidade em Software: Uma Metodologia para a Homologação de Sistemas**, Rio de Janeiro, Ed. Alta Books, 1ª Ed, 2005.

BANCO CENTRAL, **Resolução 2.554**, 1998, Disponível em <https://www3.bcb.gov.br/normativo/detalharNormativo.do?N=098186548&method=detalharNormativo> (último acesso 10/2011)

CROSBY, Philip B., **Qualidade é investimento**. Rio de Janeiro, José Olympio Editora, 1985.

LONGO, Rose Mary Juliano, **Artigo: Gestão da Qualidade: Evolução Histórica, Conceitos Básicos e Aplicação na Educação**, 1996. Disponível em http://www.ipea.gov.br/pub/td/td_397.pdf (último acesso 10/2011)

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO 9001: Sistemas de Gestão da Qualidade – Requisitos**. Rio de Janeiro: ABNT, 2000.

IBM, **Rational Unified Process: Best Practices for Software development Teams**, 1998. Disponível em http://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf (último acesso 11/2011).

ARQUITETURA, Comitê . **Apresentação: Metodologia de Desenvolvimento de Sistemas**. 2009. Disponível em <https://digital.itau/Bem-vindo/areas.asp> (último acesso 07/2011)

GMQS, Gerência de Metodologia e Qualidade de Sistemas. **Base de conhecimento Wiki da Metodologia de Desenvolvimento – MDS**. 2009. Disponível em <http://sscorp1.itau/sites/sass/Wiki/PaginasWiki/MDS-MetodologiadDesenvolvimento de Sistemas.aspx> (último acesso 07/2011)