UNIVERSIDADE ESTÁCIO DE SÁ

GIULLIA EDWIGES PIRES BOENTE

A Teoria dos Conjuntos *Fuzzy* como Ferramenta para Avaliação da Qualidade de *Software* de Automação Industrial

Rio de Janeiro
2011

GIULLIA EDWIGES PIRES BOENTE

A Teoria dos Conjuntos *Fuzzy* como Ferramenta para avaliação da Qualidade de *Software* de Automação Industrial.

Projeto de Pesquisa apresentado à Disciplina de Projeto Final I do Curso de Bacharel em Sistemas de Informação da Universidade Estácio de Sá, como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação.

Rio de Janeiro
2011

SUMÁRIO

1. **PROBLEMÁTICA**

1.1. CONTEXTO

Na área de produção de *software*, há uma urgente necessidade de um maior foco sobre a questão de qualidade. Segundo Mecenas e Oliveira (2005), a produção de *software* deixou de ser, há algum tempo, uma atividade baseada apenas na intuição ou na experiência dos desenvolvedores. O processo de desenvolvimento de *software* tem sido objeto de inúmeros estudos, há mais de três décadas, numa tentativa de derivar modelos que possibilitem o gerenciamento das fases de produção e assegurem que os produtos tenham a qualidade desejada pelos consumidores.

Nesse contexto, existem inúmeras definições a serem consideradas. Segundo a NBR ISO 8402 (2008), "qualidade é a totalidade das características de uma entidade que lhe confere a capacidade de satisfazer as necessidades explícitas e implícitas, onde, entidade é o produto do qual estamos nos referindo, que pode ser um bem ou prestação de serviço". As necessidades explícitas são as próprias condições e objetivos propostos pelo produtor. As necessidades implícitas incluem a diferença entre o usuário e a evolução do tempo, as implicações éticas, as questões de segurança e outras visões subjetivas.

Pressman (2006) propõe que o uso de metodologias para desenvolvimento de *software* seja o primeiro passo para obtenção da qualidade de processos e produtos de *software*.

Mesmo estando no século XXI e existindo inúmeras técnicas de engenharia de software, ainda existem projetos de *software* que são gerados com pouquíssima qualidade ou até com nenhuma. Dados da *Standish Group* (1995 apud SOARES, 2004, p. 2), usando como base para aproximadamente 8.400 projetos, mostram que apenas 16,2% dos projetos foram entregues respeitando os prazos e os custos e com todas as funcionalidades

especificadas; Aproximadamente 31% dos projetos foram cancelados antes de sua finalização efetiva e 52,7% foram entregues, mas com prazos maiores, custos maiores ou com menos funcionalidade do que o especificado no início do projeto. Dentre os projetos que não foram finalizados de acordo com os prazos e custos especificados, a média de atrasos foi de 22,2%, e a média de custo foi de 18,9% a mais do que o previsto. Considerando todos os projetos que foram entregues além do prazo e com custo maior, na média, apenas 61% das funcionalidades originais foram incluídas. Mesmo os projetos cuja entrega foi feita respeitando os limites de prazo e custo, possuiram qualidade suspeita, uma vez que provavelmente foram feitos com muita pressão sobre os desenvolvedores, o que pode quadruplicar o número de erros de *software*, segundo a mesma pesquisa.

No entanto, Soares (2004), propõe que ao se aplicar as normas de engenharia de *software*, os resultados iniciais em termos de qualidade, confiança, datas de entrega e custo são promissores.

Mello Filho (2009) afirma que “o grande mérito do relatório da *Standish Group* foi mostrar a generalização do problema e expondo a primitiva prática de projeto aplicado na área”. Tomando como base a realização de uma pesquisa com 365 organizações de desenvolvimento de *software* e abrangendo 3682 sistemas ou projetos, foi demonstrado pelo projeto que 222% indicava o atraso médio na entrega de um projeto de *software* e em média 189% dos projetos apresentavam seus custos além do estimado no início do projeto. Em média, somente 61% do que era combinado realmente era entregue (MELLO FILHO, 2009).

Qualidade é o sucesso para o negócio de *software*, como em qualquer outro. Uma questão chave é verificar se realmente os gerentes de projeto estão satisfeitos com os

aspectos da qualidade de *software* produzidos. Na literatura pesquisada não se tem um modelo que possa realizar tal avaliação.

Em Fundações Públicas Estaduais, em particular na FAETEC - Fundação de Apoio à Escola Técnica existe uma necessidade, por parte da diretoria de informática, DINFO, em saber se os gerentes de projetos estão satisfeitos com a qualidade dos softwares que são produzidos por suas equipes.

Considerando que avaliação da qualidade de *software* e a satisfação dos gerentes de projetos são aspectos de representação imprecisos, composto, em sua maioria, por conceitos subjetivos e de avaliação não trivial, utilizaremos a teoria dos conjuntos *fuzzy* como diretriz base no auxílio da análise dos dados.

1.2. FORMULAÇÃO DO PROBLEMA

Como avaliar a satisfação dos gerentes de projetos quanto à qualidade de *software*?

1.3. OBJETIVOS DA PESQUISA

1.3.1. OBJETIVO GERAL

Avaliar a satisfação dos gerentes de projetos quanto à qualidade de *software*.

1.3.2. OBJETIVOS ESPECÍFICOS

Entre os objetivos específicos, podem ser listados:

- Identificar qualidade;
- Discutir a qualidade dos processos de *software*;
- Discutir a qualidade dos produtos de *software*;
- Discutir os modelos de índice de satisfação;
- Discutir a teoria dos conjuntos *fuzzy*.

1.4. HIPÓTESE

Para efeito desta pesquisa, utilizando-se a teoria dos conjuntos *fuzzy* que constitui-se em variáveis lingüísticas, supõe-se que é possível mensurar a satisfação dos gerentes de projetos a partir de indicadores de satisfação quanto a qualidade de *software* produzidos.

1.5. DELIMITAÇÕES

A pesquisa está direcionada à mensuração da satisfação dos gerentes de projetos quanto à qualidade de *software*. Segundo Ireland (1991 apud VARGAS, 2002, p. 70) "o conceito moderno de qualidade tem foco na satisfação do cliente e na conformidade do projeto com as necessidades desse cliente, e não com padrões previamente criados pela empresa ou pelo time do projeto". Não existem parâmetros fixados que mensurem a satisfação do gerente de projeto (PMI, 2005, p. 111). No entanto, como o gerente de projeto também é considerado um cliente interno, acredita-se que estudos sobre a satisfação de clientes venham contribuir para identificar variáveis para realizar tal mensuração. Assim, procurar-se-á identificar as possíveis variáveis de mensuração da satisfação dos gerentes de projetos quanto à qualidade do *software* produzido por uma Fundação Pública Estadual.

1.6. JUSTIFICATIVA

Em relação à academia, o estudo contribui para a área de administração, marketing e informática, por abordar um tema que se situa na intersecção das áreas de gestão, mercado e tecnologia da informação.

É importante para a equipe de desenvolvimento de *software* da Diretoria de Informática (DINFO) da Fundação de Apoio à Escola Técnica do Estado do Rio de Janeiro (FAETEC), visto que a diretoria geral da DINFO, através dessa pesquisa, poderá identificar se existe satisfação dos gerentes de projetos, quanto à qualidade de *software*.

Apresenta uma relevância pessoal visto que, como engenheiro de *software*, irei beneficiar-me ao identificar as variáveis que venham medir a satisfação de gestores de projetos, quanto à qualidade de *software*.

1.7. DEFINIÇÃO DE TERMOS

Desenvolvimento de sistemas - atividade de implantação de sistemas por meio de linguagens de programação (PRESSMAN, 2006).

Garantia da qualidade - visa garantir que os processos e produtos de *software* estejam em conformidade com os requisitos especificados e aos planos estabelecidos (ISO/IEC 9126-1:2001, 2009).

Gestão da qualidade - atividades coordenadas para dirigir e controlar uma organização, no que diz respeito à qualidade (ISO/IEC 9126-2:2001, 2009).

Processo de *software* – refere-se às atividades que definem certo projeto de *software* (ISO/IEC 9126-3:2001, 2009).

Produção de *software* - envolve, num ciclo de vida de *software*, as atividades de análise, projeto e implementação de *software* (ISO/IEC 9126-3:2001, 2009).

Produto de *software* – resultado consistente proveniente da produção de *software* (ISO/IEC 9126-4:2001, 2009).

Qualidade de *software* - habitual prática do processo de produção de *software* segundo os princípios da engenharia de *software* (ISO/IEC 9126-1:2001, 2009).

Métrica de *software* - representa a relação estabelecida entre medidas de alguma propriedade do *software* ou da sua especificação (ISO/IEC 9126-4:2001, 2009).

2. REFERENCIAL TEÓRICO

O objetivo deste capítulo é fazer uma revisão teórica a respeito da qualidade e satisfação dos gerentes de projetos, tendo como foco principal a qualidade de *software* e os modelos nacionais de índice de satisfação do cliente existentes.

2.1. QUALIDADE

Segundo a atual norma brasileira que trata sobre o assunto, NBR ISO 8402, qualidade é a totalidade das características de uma entidade que lhe confere a capacidade de satisfazer as necessidades explícitas e implícitas.

Pela interpretação da norma, entidade refere-se ao produto ou prestação de serviço. As necessidades explícitas são as próprias condições e objetivos propostos pelo contratante/contratado. As necessidades implícitas incluem as diferenças entre os diversos usuários, a evolução no tempo, as implicações éticas, as questões de segurança e outras visões subjetivas.

Ao perguntar-se para qualquer pessoa o que pensa sobre qualidade, ou como diferencia a qualidade entre um e outro produto, recebemos uma enormidade de respostas que representam a idéia sobre a sensibilidade do indivíduo acerca daquilo que a mesma cria sobre o produto ofertado ou avaliado (SIMÃO, 2003, p. 28).

Rezende (1999, p. 97) define qualidade como a:

> [...] conformidade com os requisitos, ou adaptabilidade ao uso, adequação ao cliente e/ou usuário; atendimento perfeito de forma confiável (sem defeitos), acessível (baixo custo), segurança e no tempo certo às necessidades do cliente; é a ausência de desperdício, é “atitude”.

Rodrigues (2006, p. 11) afirma que a “qualidade é o que o cliente/usuário, percebe ou entende por valor, diante do seu socialmente aprendido, do mercado ou sociedade e das tecnologias disponíveis”.

A figura 1, em conformidade com a ABNT – Associação Brasileira de Normas Técnicas mostra o modelo de gestão da qualidade adotado pela ISO 9000:2000, onde a partir das necessidades e expectativas de clientes, tem-se a agregação de valores aos produtos e prestações de serviços, visando atingir a satisfação deles.

Quando os benefícios pesam significativamente mais que os custos, valor alto é percebido e os consumidores ficam satisfeitos (CHURCHILL, 2000, p. 151). A satisfação do consumidor e o valor recebido por ele pode influenciar em decisões de compras futuras.

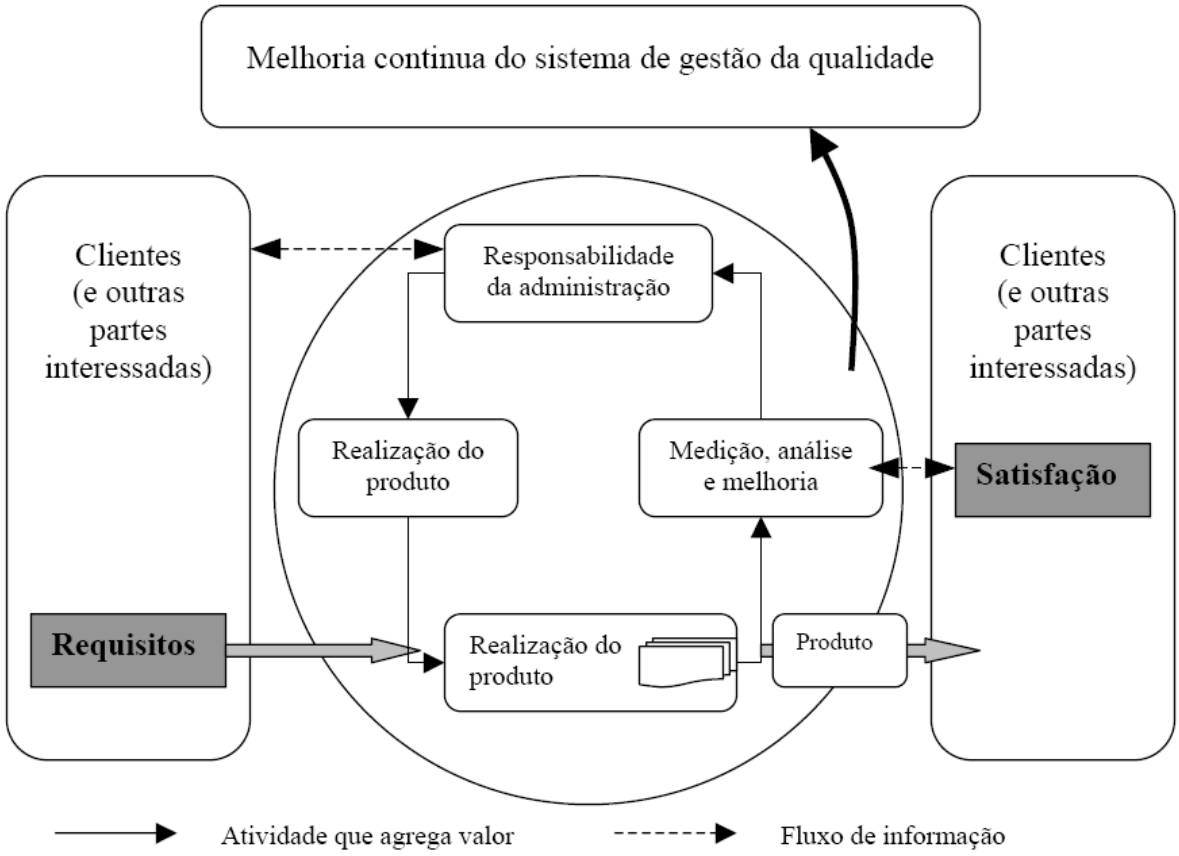

Figura 1. Modelo do Sistema de Gestão da Qualidade da ISO 9000:2000
Fonte: González, 2005, p. 20.

A satisfação pode ser definida como a avaliação pós-consumo de que uma alternativa escolhida pelo consumidor ao menos atende ou excede as expectativas do mesmo (ENGEL et al., 2000 apud IZARD, 2007, p. 34).

Para Slack (1997 apud LOPES, SILVA LEITE e SILVA LEITE, 2007, p. 5), qualidade é “a consistente conformidade com as expectativas do cliente”. Este conceito adota a definição da qualidade baseada no usuário, isto é, na sua percepção de valor.

Qualidade de *software* não pode ser avaliada isoladamente. No desenvolvimento de *software*, um método pobre ou a ausência de uma metodologia pode ser a causa da baixa qualidade. A avaliação da qualidade está diretamente relacionada com a qualidade de processos e metodologias utilizadas no desenvolvimento do *software* (BOENTE, OLIVEIRA e ALVES, 2008, p. 2).

No contexto da qualidade do desempenho de um *software*, Mecenas e Oliveira (2005, p. 38), afirmam que “[...] constitui um mito e um desafio nos meios tecnológicos”. Apesar de o *software* ser um produto ligado à alta tecnologia, como qualquer outro produto, tem a origem das suas métricas de qualidade baseadas nas práticas voltadas para produção de manufaturado (CÔRTEZ e CHIOSSI, 2001 apud MARQUES e SILVA, 2008, p. 2). Do mesmo modo, este produto objetiva com a formulação e a implantação de modelos de qualidade obter, principalmente, a satisfação das necessidades do usuário final e o aprimoramento do processo de produção (MARQUES e SILVA, 2008, p. 3).

Dromey (2008) propõe que alguns desenvolvedores procurem fazer uma abordagem da qualidade de *software* como uma habitual prática do processo de produção de *software* segundo os princípios da engenharia de *software*.

Pressman (2006, p. 724) define qualidade de *software* como:

> A conformidade a requisitos funcionais e de desempenho explicitamente declarados, a padrões de desenvolvimento claramente documentados e a características implícitas que são esperadas de todo o *software* profissionalmente desenvolvido.

Na figura 2, observam-se os requisitos adotados para a qualidade. A análise destes requisitos segundo esse critério é muito útil para qualquer empresa de desenvolvimento de *software* que trabalhe nesse mercado globalizado e altamente competitivo.

Rezende (1999, p. 97) afirma que "um *software* ou sistema de informação tem qualidade quando está adequado à empresa, ao cliente e/ou usuário e atende a padrões de qualidade predefinidos".

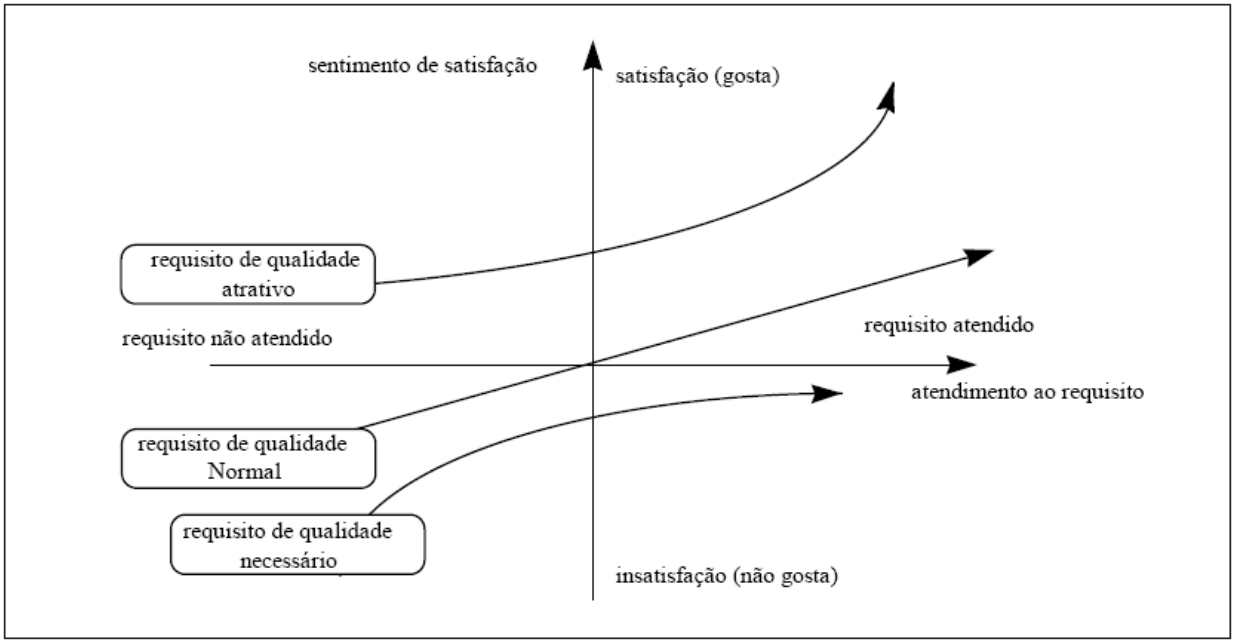

Figura 2. Requisitos da Qualidade
Fonte: Tsukumo et al., 1997, p. 175.

No contexto de desenvolvimento de *software*, Rocha (1994) diz que qualidade pode ser entendida como um conjunto de características a serem satisfeitas em um determinado grau, de modo que o produto de *software* atenda às necessidades explícitas e implícitas de seus usuários. Contudo, não se consegue obter qualidade do produto de forma espontânea. A qualidade deve ser construída ao longo do processo do desenvolvimento do *software* e também após a entrega do mesmo.

A gestão da qualidade faz parte das atividades que envolvem o gerenciamento e controle de projetos. O controle da qualidade do projeto inclui os processos requeridos para garantir que o projeto de *software* irá satisfazer ou até superar as expectativas de seu contratante (BOENTE, 2003, p. 66).

Concomitantemente ao desenvolvimento de novos produtos e serviços, as organizações precisam de agilidade nos seus processos operacionais, mudanças em tecnologias de produto, atualizações nos seus sistemas de informação e redefinição das formas de interação com fornecedores e clientes. Poucas organizações podem escapar da introdução de projetos em suas atividades diárias. Este fenômeno tem resultado em um maior interesse na aplicação da gestão de projetos como uma metodologia formal de gestão da qualidade (MORRISON e BROWN, 2004 apud PAPADIUK e SANTOS, 2008, p. 3).

Gestão da qualidade são atividades coordenadas para dirigir e controlar uma organização, no que diz respeito à qualidade (CASTRO, 2003, p. 4).

De acordo com Gil (1999, p. 140):

> [...] é pressuposto básico para o processo de melhoria continuada em informática consoante os vetores a seguir expostos:
> a) Exercida quanto à capacidade, conhecimento e atuação dos *Stakeholders* envolvidos no processo;
> b) Existência de programas e incentivo à gestão da qualidade em informática praticada por executivos de informática da empresa.

Do ponto de vista do gerenciamento, Nogueira (2006), propõe que os objetivos gerenciais básicos - eficiência e eficácia - se apresentem intrinsecamente relacionados à gestão da qualidade.

Boente (2003) propõe que o controle da qualidade de certo projeto deva ser direcionado tanto para o gerenciamento do projeto em si, quanto para o produto a ser gerado pelo projeto.

Conforme é proposto por Cunha, Cunha e Dahab (2001):

> O desenvolvimento e a maturação da gestão da qualidade fizeram este movimento de gestão extravasar do seu domínio industrial inicial, levando-o a adquirir proeminência em todos os setores de atividade, incluindo os setores públicos e privado, industrial e de serviços. Em simultâneo, a qualidade adquiriu o estatuto de campo teórico acessível para a comunidade acadêmica, o que gerou esforços significativos para a expansão e refinamento das suas bases teóricas.

O gerenciamento da qualidade do projeto inclui os processos necessários para garantir que o projeto irá satisfazer as necessidades para as quais ele foi empreendido (PMI - *Project Management Institute*, 2002, p. 95).

Figura 3. Visão Geral do Gerenciamento da Qualidade do Projeto
Fonte: PMI, 2002, p. 96.

A gestão da qualidade, de acordo com Rodrigues (2006, p. 20), "é integrada através de ações estratégicas, ações comportamentais, ações operacionais e ações estruturais". Essas ações estratégicas envolvem o desdobramento eficaz e focado das estratégias

componentes estratégicos, objetivos e metas. As ações comportamentais envolvem comprometimento, capacitação e integração. As ações operacionais envolvem ferramentas, análise estatística e programas operacionais. As ações estruturais envolvem a reestruturação interna e otimização da cadeia de suprimentos (RODRIGUES, 2006, p. 20-21).

Pressman (2006, p.724) afirma que a garantia da qualidade de *software* é uma "atividade de guarda-chuva" que é aplicada ao longo do processo de engenharia de *software*.

A INB, Indústrias Nucleares do Brasil, (2008) propõe que a garantia da qualidade seja a:

> [...] área de importância fundamental para todo o processo produtivo [...] realiza, rotineiramente, auditorias para verificação do atendimento aos rigorosos padrões de qualidade exigidos aos seus produtos. Estabelecido de maneira a atender principalmente as normas da [...] Associação Brasileira de Normas Técnicas - NBR/ISO 9001.

Portanto, produzir *software* de qualidade é fazer verificações constantes em padrões de qualidade exigidos por cada produto de *software* a ser desenvolvido, garantindo assim, sua efetiva qualidade.

O PMI (2002, p. 95) determina que "o objetivo da garantia da qualidade seja avaliar periodicamente o desempenho geral do projeto visando assegurar a satisfação dos padrões de qualidade relevantes".

O processo de garantia da qualidade da norma ISO/IEC 12207 é usado para garantir que os processos e produtos de *software* estejam em conformidade com todos os requisitos que foram especificados e aos planos que foram estabelecidos (ROCHA, MALDONADO e WEBER, 2001, p. 64).

Pressman (2006) propõe que o processo de garantia da qualidade deva estar coordenado com os processos de verificação, validação, teste, revisão conjunta e auditoria.

Rocha, Maldonado e Weber (2001, p. 71) afirmam que o processo de verificação consiste em duas em:

❶ Verificação do contrato – observa se o fornecedor apresenta capacidade para satisfazer os requisitos, se esses são coerentes e abrangentes às necessidades do usuário, se existem procedimentos adequados de acordo com os critérios de aceitação do produto ou prestação de serviço, para manipular as mudanças dos requisitos etc.

❷ Verificação do processo: observa se o planejamento do projeto e a atribuição de tempo estão adequados, se o que foi determinado no projeto está sendo seguido e está de acordo com o contrato, se a equipe de desenvolvimento possui treinamento desejado etc.

❸ Verificação dos requisitos: observa se os requisitos do sistema e do *software* são coerentes, são factíveis e testáveis, se ambos os requisitos se refletem com precisão.

❹ Verificação do projeto: observa se o projeto está correto e coerente com os requisitos, se ele implementa apropriadamente as seqüências de eventos, entradas, saídas, requisitos de segurança com métodos rigorosos etc.

❺ Verificação do código: observa se o código está de acordo com os requisitos e padrões, se é testável e correto, se implementa requisitos críticos e de segurança com métodos rigorosos etc.

❻ Verificação da integração: observa se os componentes e unidades de código, bem como os itens de *hardware* e *software* foram integrados completa e corretamente de acordo com um plano de integração etc.

➐ Verificação da documentação: observa se a documentação está adequada, completa, coerente, se está sendo desenvolvida no prazo e se o gerenciamento de configuração dos documentos segue os procedimentos especificados.

No que se refere ao processo de validação, Rocha, Maldonado e Weber (2001) afirmam que ele está relacionado à sua implementação e abordar aspectos relativos ao plano de validação e ao seu cumprimento, ao grau de independência organizacional desejado, na preparação dos requisitos de teste, casos de teste e especificações de teste para analisar os resultados.

Quanto ao teste, este é visto no contexto da norma como uma atividade de verificação e validação e consiste na análise dinâmica do *software* (Rocha, Maldonado e Weber, 2001, p. 73).

Existem inúmeros tipos de testes de *software* a serem realizados para complementar os processos de verificação e validação. Cada um deles adequado a uma situação particular do *software* analisado e do fluxo de informação do teste realizado. A figura 4 mostra que à medida que os resultados de testes são reunidos e avaliados, uma indicação quantitativa da qualidade e da confiabilidade do *software* começa a parecer claramente.

Na revisão do produto de *software* artefatos são revisados ao longo do processo de desenvolvimento, garantindo assim, que a equipe de desenvolvimento esteja utilizando documentos pelo menos com a qualidade mínima especificada (Pressman, 2006, p. 874).

Fagan (1976 apud ROCHA, MALDONADO e WEBER, 2001, p. 85) afirma que “inicialmente, revisões de *software* foram aplicadas especificamente a código-fonte”. A partir de então, Rocha, Maldonado e Weber (2001) citam que “diferentes abordagens e técnicas vem sendo empregadas não somente ao código-fonte, mas a todos os artefatos gerados ao longo do processo de desenvolvimento”.

Seguindo os padrões e técnicas de engenharia de *software*, hoje, se tem equipes de desenvolvimento de *software* revisando constantemente todo o processo de desenvolvimento primando assim, pela garantia da qualidade do produto de *software* a ser gerado.

Segundo Gil (1998, p. 53) auditoria "é a função administrativa de revisão das funções planejadas, execução e controle, e como tal é operacionalizada". Portanto, auditoria vem agregar valor a atividade de revisão que é exercida sobre o produto de *software* que está sendo desenvolvido.

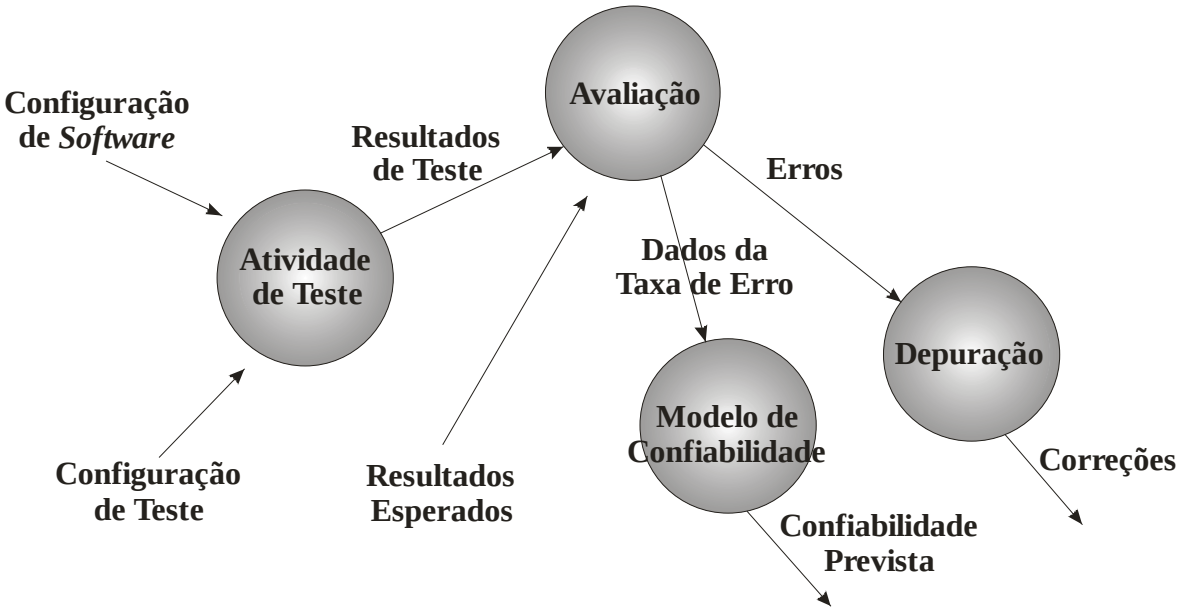

Figura 4. Fluxo de Informações de Teste
Fonte: Pressman, 2006, p. 796.

Tem-se como objetivo garantir que as atividades definidas pelo processo e padrões sejam seguidas e que os produtos de trabalho satisfaçam aos requisitos definidos para o projeto. As atividades do grupo de qualidade devem ser relatadas às gerências da organização (SOARES, 2003, p. 55).

Na garantia da qualidade se procura avaliar, de tempo em tempo, o desempenho global do projeto buscando sempre assegurar a satisfação dos padrões relevantes de qualidade do mesmo (BOENTE, 2003, p. 66).

No aspecto da garantia da qualidade, Square (2000) propõe que tal ação dependa exclusivamente da gerencia de projetos em exercer a qualidade dos processos e projetos de *software* antes, durante e depois de seu efetivo desenvolvimento. Para isso é necessário garantir o processo e o produto de *software*.

2.2. QUALIDADE DOS PROCESSOS DE *SOFTWARE*

Segundo Weber e Rocha (1999), o CMM (*Capability Maturity Model*) e o SPICE (*Software Process Improvement and Capability dEtermination*) são considerados os processos de *software* com base na Norma Internacional ISO/IEC 12207:1995 e as normas internacionais da série ISO 9000.

De acordo com Mecenas e Oliveira (2005, p. 136) o CMM é "[...] um modelo de maturidade de capacidade, criado pelo SEI (*Software Engineering Institute*) para avaliar e melhorar a capacitação das empresas que produzem *software*".

Além desses dois processos chave, existe o moderno processo denominado CMMI e outros métodos e abordagens para o processo de *software*, todos com vantagens e desvantagens em relação aos outros.

Na figura 5 são observados os níveis de maturidade do SW-CMM que, segundo Rezende (1999, p. 147), "[...] está organizado em cinco níveis distintos, cada um com suas características próprias que determina qual é a especificação do processo".

Mecenas e Oliveira (2005) definem os cinco níveis, conforme descrição a seguir:

Nível-1 (Inicial) – o processo de desenvolvimento é desorganizado e pessoal, sendo considerado assim, um processo caótico. Neste nível, poucos processos são definidos e o sucesso depende de esforços individuais e heróicos.

Nível-2 (Repetitivo) – Os processos básicos de gerenciamento de projeto estão estabelecidos e permitem acompanhar custos, cronograma e funcionalidades, sendo considerado assim, um processo disciplinado. Neste nível é possível repetir o sucesso de um processo utilizado anteriormente em outros processos similares.

Nível-3 (Definido) – Tanto as atividades de gerenciamento quanto a engenharia do processo de desenvolvimento de *software* estão documentadas e integradas aos padrões de desenvolvimento da empresa, caracterizando assim um processo padronizado e consistente. Todos os projetos, neste nível, utilizam uma versão aprovada e adaptada do processo padrão de desenvolvimento de *software* da empresa.

Figura 5. Níveis de Maturidade do SW-CMM
Fonte: Soares, 2003, p. 44.

Nível-4 (Gerenciado) – São coletadas medidas detalhas de qualidade do produto do processo de desenvolvimento de *software*, perfazendo assim um processo plenamente previsível. Neste nível, tanto o produto quanto o processo de desenvolvimento de *software* são entendidos e controlados quantitativamente.

Nível-5 (Otimizado) – Neste nível o melhoramento contínuo do processo é conseguido por meio de *feedback* quantitativo dos processos e pelo uso pioneiro de idéias e tecnologias inovadoras, fazendo assim que o processo tenha melhoria contínua.

Segundo Koscianski e Soares (2007, p. 95):

> Dois dos principais modelos criados pelo SEI (*Software Engineering Institute*) para melhoria de processos são o SW-CMM e o CMMI. Criado no final da década de 1990 apenas para *software*, o SW-CMM obteve grande sucesso ao gerar novos padrões para a engenharia de sistemas. Posteriormente, como uma evolução dos vários CMMs existentes, foi criado o CMMI.

A *Carnegie Mellon Software Engineering Institute* reconhece que o CMMI apresenta "[…] quarto disciplinas: engenharia de sistemas, engenharia de *software*, desenvolvimento e integração de produto e processo e fontes de aquisição". Essas disciplinas podem ser mais bem visualizadas por meio da figura 6.

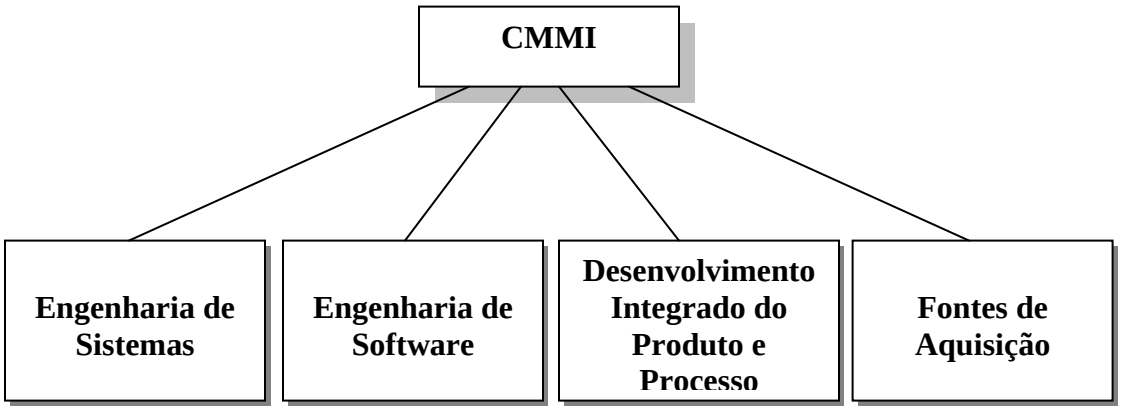

Figura 6. Disciplinas do CMMI
Fonte: Koscianski e Soares, 2007, p. 103.

A disciplina de engenharia de sistemas trás um enfoque interdisciplinar que objetiva a obtenção do sistema bem-sucedido, com apresentação ou não do produto de *software* gerado. A engenharia de *software* faz uma abordagem sistemática, disciplinada e qualificável para o desenvolvimento, operação e manutenção do *software*. O desenvolvimento integrado do produto e do processo é uma abordagem sistemática que utiliza a colaboração dos *stakeholders* para melhor satisfazer as expectativas e requisitos do

cliente. A disciplina CMMI de fontes de aquisição atua na aquisição de produtos que tenham participação efetivas fornecedores (KOSCIANSKI e SOARES, 2007, p. 104).

A abordagem para medir e melhorar o processo de *software* proposto por Rocha, Maldonado e Weber (2001) baseia-se nas etapas de seleção e definição das métricas apropriadas para a realização das medições, com base nos objetivos do projeto; na realização de medições como parte integrante do processo de desenvolvimento de *software*; análise dos resultados do uso do processo em projetos e a realização de estudos empíricos envolvendo medições de processos de *software*.

De acordo com Tsukumo et al. (1997, p. 176):

> A qualidade de *software* é largamente determinada pela qualidade dos processos utilizados para o desenvolvimento. De modo, a melhoria da qualidade de *software* é obtida pela melhoria da qualidade dos processos. Esta visão orientou a elaboração de modelos de definição, avaliação e melhoria de processos de *software*.

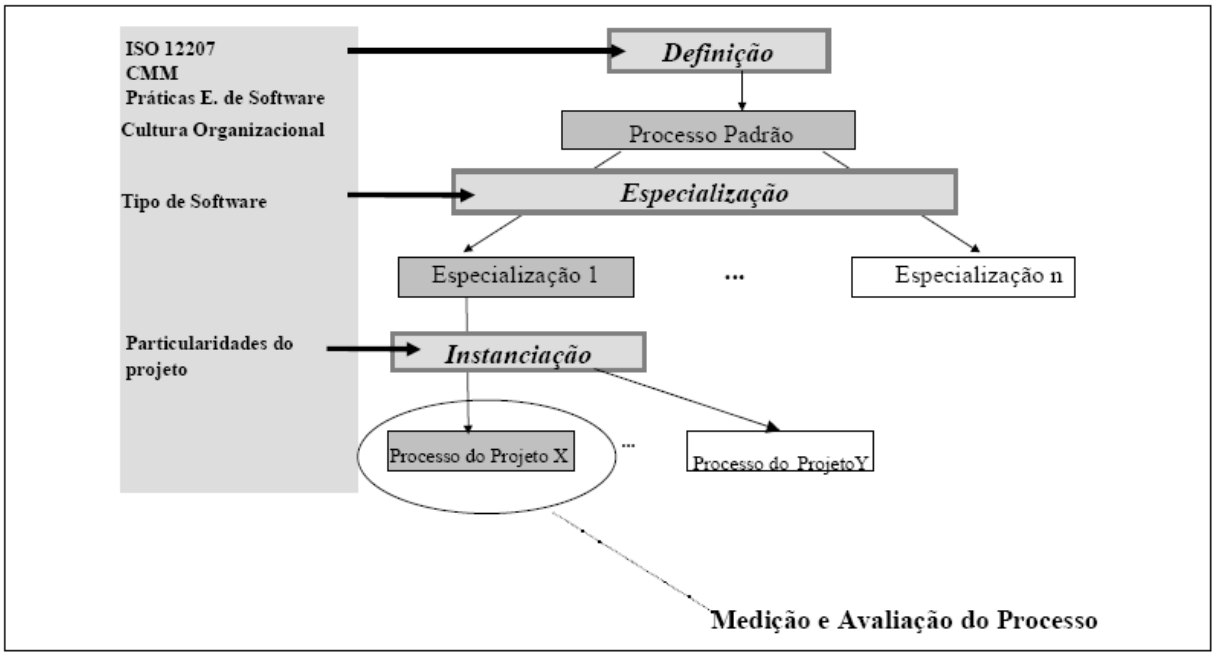

Figura 7. Modelo para Definição do Processo de *Software*
Fonte: Gomes, Oliveira e Rocha, 2001, p. 94.

A garantia do processo de *software* pode ser aferida através de sua medição, bastando definir de forma clara e precisa, a fim de evitar problemas na sua interpretação e, além disso, deve ser executado pela equipe de desenvolvimento para que os valores coletados sejam válidos para sua avaliação (GOMES, OLIVEIRA e ROCHA, 2001, p. 94).

A atividade de garantia do processo, de acordo com Rocha, Maldonado e Weber (2001, p.66) implica em:

> Garantir que os processos do ciclo de vida do *software* (fornecimento, desenvolvimento, operação, manutenção e processo de apoio) utilizados no projeto estejam de acordo com o contrato e com os planos; garantir as práticas de engenharia de *software*, o ambiente de desenvolvimento, o ambiente de teste e as bibliotecas estejam de acordo com o contrato, com as negociações e com os planos; garantir no caso de subcontratação, que os requisitos aplicáveis sejam passados ao subcontratado e que seus produtos de *software* satisfaçam os requisitos do contrato original; garantir que o adquirente e que outras partes envolvidas tenham o apoio e a cooperação necessários; garantir que as medições do produto e do processo de *software* estejam de acordo com os padrões e procedimentos estabelecidos; garantir que a equipe tenha a qualificação e o conhecimento necessários para o projeto e receba todo o treinamento necessário.

Assim, garantindo os processos (de ciclo de vida, subcontratação, medições do produto e do processo de *software*, qualificação da equipe etc.), o projeto é garantido em sua íntegra.

2.3. QUALIDADE DOS PRODUTOS DE *SOFTWARE*

Qualidade do produto é o conjunto das propriedades que determinam sua habilidade em satisfazer as necessidades para as quais ele foi criado (JURAN, 1991, p. 15).

Nogueira (2006) afirma que a “qualidade em produtos ou prestação de serviços tem sido um pré-requisito para a empresa vender e lucrar mais”. Assim, as organizações conseguem vender mais e, ocasionalmente, lucrar mais, quando geram produtos de *software* de qualidade no mercado.

De acordo com Weber e Rocha (1999), as organizações internacionais de normalização ISO/IEC vêm trabalhando em conjuntamente em um modelo que permita avaliar a qualidade dos produtos de *software*.

Rocha, Maldonado e Weber (2001, p. 116-118) afirmam que:

> O modelo de qualidade, definido na ISO/IEC 9126-1 e utilizado como referência para o processo de avaliação da qualidade de produtos de *software*, está subdividido em modelo de qualidade para características externas e internas, que classifica os atributos de qualidade de *software* em funcionalidade, confiabilidade, usabilidade, eficiência manutenibilidade e portabilidade e, modelo de qualidade para qualidade em uso, que classifica os atributos em efetividade, produtividade, segurança e satisfação.

O processo de avaliação é definido pelas seguintes normas internacionais:

a) Características da qualidade e métrica

   a1) ISO/IEC 9126-1: Modelo de Qualidade;

   a2) ISO/IEC 9126-2: Métricas Externas;

   a3) ISO/IEC 9126-3: Métricas Internas;

   a4) ISO/IEC 9126-4: Métricas da Qualidade em Uso;

b) Avaliação dos produtos de *software*;

   b1) ISO/IEC14598-1: Visão Geral;

   b2) ISO/IEC 14598-2: Planejamento e Gerenciamento;

   b3) ISO/IEC14598-3: Processo para Equipe de Desenvolvimento;

   b4) ISO/IEC 14598-4: Processo para Adquirentes;

   b5) ISO/IEC14598-5: Processo para Avaliadores;

   b6) ISO/IEC 14598-6: Documentação de Módulos de Avaliação;

c) Requisitos da qualidade e testes em pacotes de *software*;

   c1) ISO/IEC 12119: Pacotes de *Software* – requisitos da qualidade e testes;

A figura 8, obtida da ISO/IEC 9126-1, mostra a relação existente entre os documentos da série 9126 e 14598, deixando clara a abrangência da norma 14598-1 sobre

todo o processo de avaliação, bem como a necessidade de uso da norma ISO/IEC 9126-1 como referência na aplicação das métricas. As empresas estão preocupadas em saber o que significa cada uma dessas normas para melhor utilizá-las ao confeccionar produtos de *software*.

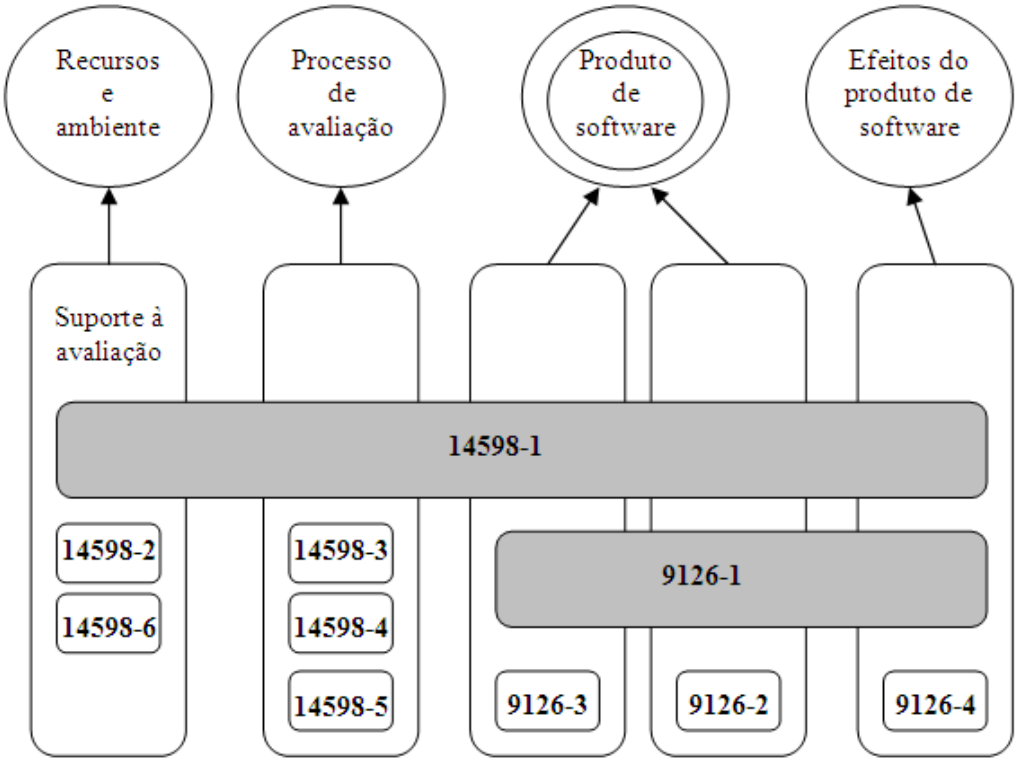

Figura 8. Custo de Mudança no Projeto no Modelo em Cascata
Fonte: Rocha, Maldonado e Weber, 2001, p. 116.

No modelo de qualidade de *software* em uso, satisfação Rocha, Maldonado e Weber (2001) propõe que os atributos sejam classificados em quatro características: efetividade, produtividade, segurança e satisfação.

Entende-se por efetividade a capacidade do produto de *software* permitir que seus usuários atinjam metas especificadas com acurácia e completitude, em um contexto de uso especificado. A produtividade é a capacidade do produto de *software* permitir que seus usuários empreguem quantidade apropriada de recursos em relação à eficácia obtida, em um contexto de uso específico. Segurança é a capacidade do produto de *software* de

apresentar níveis aceitáveis de riscos de danos a pessoas, negócios, *software*, propriedade ou ambiente, em um contexto de uso especificado. Satisfação entende-se pela capacidade do produto de *software* satisfazer seus usuários, em um contexto específico.

Tabela 1. Conhecimento de Normas da Qualidade de Produtos

Categorias	NBR 13596 (ISO/IEC 9126)		ISO/IEC 14598		ISO/IEC 12119	
	Nº	%	Nº	%	Nº	%
Conhece e usa sistematicamente	16	3,9	5	1,2	10	2,4
Conhece e começa a usar	31	7,5	14	3,4	22	5,4
Conhece, mas não usa	224	54,4	228	55,6	230	56,0
Não conhece	141	34,2	163	39,8	149	36,3
Base	412	100	410	100	411	100

Fonte: Ministério da Ciência e Tecnologia, 2008.

Segundo o Ministério da Ciência e Tecnologia (2008), em pesquisa realizada em organizações cerca de 65% das empresas conheciam as normas ISO/IEC 2196 ou ISO/IEC 12119, e um pouco menos (60%), a ISO/IEC 14598 (vide tabela 1 e gráfico 1).

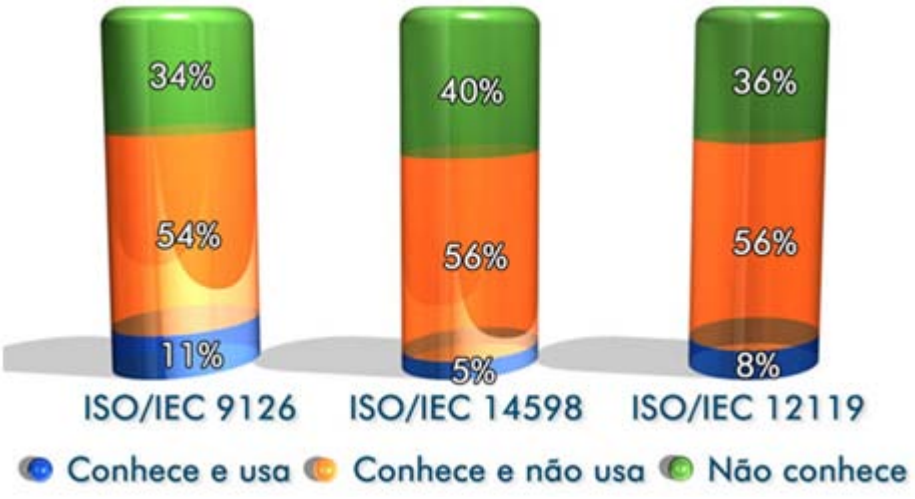

Gráfico 1. Conhecimento de Normas da Qualidade de Produtos
Fonte: Ministério da Ciência e Tecnologia, 2008.

O processo de avaliação dos produtos de *software* encontra-se definida na série de normas ISO/IEC 14598, é proposto por Rocha, Maldonado e Weber (2001) que seja utilizada em conjunto com a série ISO/IEC 9126.

De acordo com Tsukumo (1997, p. 16):

> A qualidade de um produto de *software* é resultante das atividades realizada dos processos de desenvolvimento do mesmo. Avaliar a qualidade de produtos de *software* é verificar, através de técnicas e atividades operacionais o quanto os requisitos são atendidos. Tais requisitos, de uma maneira geral, são a expressão das necessidades, explicitados em termos de quantitativos ou qualitativos, e tem por objetivo definir as características de um *software* [...].

Pode-se afirmar que a garantia da qualidade do *software* está diretamente relacionada com a garantia do processo e com a garantia do produto de *software* a ser confeccionado e, esta garantia envolve aplicação de métodos técnicos, realização de revisões técnicas formais, atividades de teste de software, aplicação de padrões, controle de mudanças, métrica de *software* e manutenibilidade do produto (BOENTE, OLIVEIRA e ALVES, 2008, p.8).

A atividade de garantia do produto segundo Rocha, Maldonado e Weber (2001, p. 65) implica em:

> Garantir que todos os planos exigidos pelo contrato estejam de acordo com o mesmo, sejam documentados, estejam mutuamente consistentes e sejam executados quando exigidos; garantir que os produtos de *software* e sua documentação estejam de acordo com o contrato e os planos; garantir que os produtos de *software* satisfaçam totalmente os requisitos contratuais e sejam aceitáveis pelo adquirente.

A garantia efetiva do produto garante a qualidade do *software* ou sistema de informação produzido.

Os sistemas baseados em computador, de acordo com Carosia (2004), estão sendo usados nas mais diversas áreas e na maioria das situações não admitem erros e, se algum sistema de uso global deixar de funcionar, aproximadamente 40% da população mundial

sofrerá as conseqüências do problema. Assim, torna-se primordial a garantia da qualidade no processo de produção de *software*.

De acordo com Rezende (1999, p. 98), produtividade de *software*:

> [...] é a relação entre os resultados obtidos e os recursos disponíveis consumidos; produzir cada vez mais e melhor, com máxima satisfação das necessidades dos clientes; vem do latim *productivus*, que significa fértil, rendoso, proveitoso, profícuo; é diferente de produção que é simplesmente quantidade produzida, sem valor e uso.

Um *software* tem produtividade, conforme propõe Weber e Rocha (1999), quando seu produto foi disponibilizado com qualidade no tempo pré-estabelecido no escopo do projeto. Assim, pode-se considerar que qualidade e produtividade são conceitos amplos, pois representam uma filosofia de gestão, que visa conduzir as organizações a uma postura de melhoria de seus processos, por meio do compromisso de seus dirigentes e empregados e a partir de métricas de *software* eficientes.

A métrica de *software* representa a relação estabelecida entre medidas de alguma propriedade do *software* ou da sua especificação (MAFFEO, 1992, p. 391).

Na maioria dos empreendimentos técnicos, as medições e as métricas ajudam-nos a entender o processo técnico usado para desenvolver um produto, como também o próprio produto (PRESSMAN, 2006, p. 56).

As políticas de qualidade incorporadas pela Ciência da Administração serviram de base para o desenvolvimento de métricas de controle dos procedimentos da engenharia de *software*, bem como as determinações previstas em normas, entre elas a ISO 12.207 (STRAFACCI JÚNIOR, 2002, p. 29).

De acordo com o IEEE (1998) “a mensuração é realizada através do estabelecimento de um programa de métricas”. O padrão IEEE 1061, classifica as métricas de *software* em duas categorias: de produto e de processo.

As métricas de produto são usadas para medir as características da documentação ou do código, enquanto que as métricas de processo são utilizadas para medir características dos métodos, técnicas e ferramentas envolvidos no projeto de desenvolvimento de *software* (FIGUEIRA, BECKER e RUIZ, 2007, p. 2).

Pressman (2006) afirma que o *software* pode ser medido de diversas formas: indicar a qualidade do produto; avaliar a produtividade das pessoas que produzem o produto; formar uma linha básica para estimativas; ajudar a justificar os pedidos de novas ferramentas ou treinamento adicional.

A medição tem seu papel muito importante dentro da engenharia de *software*, especialmente na gerência de projetos de *software*, seja qual for à metodologia de desenvolvimento a ser utilizada.

Com base nessa afirmativa Maffeo (1992) mostra que existem várias metodologias de métricas de *software* como: métricas orientadas a seres humanos, métricas orientadas a função, métricas orientadas ao tamanho, métricas de produtividade, métricas de qualidade, métricas técnicas etc.

A medição é analisada pelos gerentes de projetos de *software* e coletada pelos engenheiros de *software*. Ela se refere à mensuração dos indicadores quantitativos do tamanho e complexidade de um sistema. Estes indicadores são utilizados para correlatar contra os desempenhos observados no passado a fim de derivar previsões futuras. A medição é aplicada em diferentes ambientes, conforme mostra a figura 9.

Pode-se identificar como estão sendo utilizados os recursos disponíveis; verifica-se quanto mede e como está a qualidade dos produtos de *software*; identifica-se como estão sendo recebidos e percebidos os trabalhos e os produtos pelos clientes; verifica-se como

estão sendo realizados os processos (trabalhos de desenvolvimento de *software*); identifica-se como está sendo feito o gerenciamento do contexto de TI.

Figura 9. Ambientes para Aplicação de Métricas.

Um programa de medição envolve a geração de um grande volume de dados, que somente pode ser manipulado e analisado de forma eficaz, se devidamente automatizado (FRANCA, STAA e LUCENA, 1998, p. 72).

As normas ISO/IEC 9126-2:2001, 9126-3:2001 e 9126-4:2001 expressam com clareza tudo que precisa-se saber sobre métricas de *software*, para atendimento de certo programa de medição.

A métrica de *software* então tem como princípios especificar as funções de coleta de dados de avaliação e desempenho, atribuir essas responsabilidades a toda a equipe envolvida no projeto, reunir dados de desempenho pertencentes à complementação do *software*, analisar os históricos dos projetos anteriores para determinar o efeito desses fatores e utilizar esses efeitos para pesar as previsões futuras.

Na área da engenharia de *software* a medição de *software* é campo de pesquisa relevante. Pfleeger (1997 apud FRANCA, STAA e LUCENA, 1998, p. 74) afirma que:

> Em qualquer campo científico, as medições fornecem descrições quantitativas dos processos e dos produtos, possibilitando a compreensão de comportamentos e de resultados. Este aumento de entendimento permite a melhor seleção de técnicas e ferramentas para controlar e melhorar os processos, produtos e recursos. Como a Engenharia envolve a análise de medições, a Engenharia de *Software* somente será uma verdadeira engenharia, quando estiver sedimentada numa sólida fundação de teorias de medição.

As métricas de *software*, de acordo com Pressman (2006), são evidentes por si mesmas, pois elas nos capacitam a quantificar e, por conseguinte, a administrar mais efetivamente.

Medidas oferecem visibilidade de como os processos, produtos, recursos, métodos, e tecnologias de desenvolvimento de *software* se relacionam entre si. Medidas podem nos ajudar a responder perguntas sobre a efetividade de técnicas ou ferramentas, sobre a produtividade de atividades de desenvolvimento e a qualidade dos produtos (SANTOS FILHO, 2002, p. 10).

Métricas de *software* são utilizadas e o histórico de aferições passadas é usado como base a partir da qual estimativas são feitas (PRESSMAN, 2006, p. 57).

O modelo da qualidade de *software* descrito na ISO/IEC 9126, de acordo com Mecenas e Oliveira (2005, p. 40), compreende a funcionalidade, confiabilidade, usabilidade, eficácia, manutenibilidade e portabilidade.

Pressman (2006, p. 754) afirma que a Força Aérea dos Estados Unidos da América, baseado nos conceitos propostos pelo IEEE 982.1, usa informações obtidas do projeto arquitetural e de dados para derivar um índice de qualidade da estrutura de projeto (DSQI = *Design Structure Quality Index*), variando entre 0 e 1, perfazendo com que sejam gerados os seguintes valores:

*S*1 = número total de módulos na arquitetura de programa;

*S*2 = número total de módulos cuja função correta dependa da fonte de entrada de dados ou que produza dados a ser usado em outro lugar;

*S*3 = número de módulos cuja função correta dependa do processamento anterior;

*S*4 = número de itens de banco de dados diversos;

*S*5 = número de itens de bancos de dados únicos;

*S*6 = número de segmentos de bancos de dados;

*S*7 = número de módulos com uma única entrada e saída.

Os valores acima são determinados para um programa de computador e os valores intermediários abaixo podem ser computados (PRESSMAN, 2006, p. 755):

*D*1 = estrutura do programa (projeto estruturado usando métodos distintos - *D*1 = 1; caso contrário, *D*1 = 0);

*D*2 = interdependência modular: *D*2 = 1 - (*S*2/*S*1);

*D*3 = módulos não dependentes de processamento anterior: *D*3 = 1 - (*S*3/*S*1);

*D*4 = tamanho do banco de dados: *D*4 = 1 - (*S*5/*S*4);

*D*5 = compartimentalização do banco de dados: *D*5 = 1 - (*S*6/*S*4);

*D*6 = característica de entrada / saída modular: *D*6 = 1 - (*S*7/*S*1);

Com base nesses valores intermediários determinados o índice da qualidade da estrutura de projeto é computado através da expressão DSQI = $\sum wiDi$**,** onde *i* = 1 a 6, *wi* é o peso relativo da importância de cada um dos valores intermediários e $\sum wi$ = 1. No entanto, se todos os *Di* tiverem um peso igual, então *wi* = 0,167 (PRESSMAN, 2006, p. 756).

O IEEE 982.1 sugeri um índice de maturidade de *software* (SMI - *Software Maturity Index*) que forneça a indicação da estabilidade de um *software* onde as seguintes informações são determinadas, de acordo com Pressman (2006):

M_T = número de módulos da versão atual;

F_c = número de módulos da versão atual que foram mudados;

F_a = número de módulos da versão atual que foram adicionados;

F_d = número de módulos da versão anterior que foram suprimidos da liberação atual;

Assim, a expressão SMI = $\frac{[M_T - (F_a + F_c + F_d)]}{M_T}$, representa o índice de maturidade de *software* que é computado.

De acordo com Simão e Belchior (2003) existem seis características de qualidade para componentes de *software*: funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade, conforme mostra a tabela 2. Estas, por sua vez, podem ser divididas da seguinte forma:

❶ Funcionalidade: Adequação, Acurácia, Autocontido, Coesão Funcional, Inteoperabilidade, Segurança de Acesso e Conformidade com a Funcionalidade;

❷ Confiabilidade: Maturidade, Tolerância a Falhas, Recuperabilidade, Avaliabilidade e Conformidade com a Confiabilidade;

❸ Usabilidade: Acessibilidade, Legilibilidade, Inteligibilidade, Facilidade de Uso, Apreensibilidade, Operacionalidade, Atratividade e Conformidade com a Usabilidade;

❹ Eficiência: Comportamento em Relação ao Tempo, Comportamento em Relação aos Recursos, Comportamento em Relação ao Estado, Escalabilidade, Nível de Granularidade e Conformidade com a Eficiência;

❺ Manutenibilidade: Analisabilidade, Implementabilidade, Modificabilidade, Estabilidade, Testabilidade e Conformidade com a Manutenibilidade;

❻ Portabilidade: Adaptabilidade, Capacidade de ser Instalado, Coexistência, Substituibilidade e Conformidade de com a Portabilidade.

Tabela 2. Características da Qualidade para Componentes de *Software*

Características de Qualidade de Software
Funcionalidade
Confiabilidade
Usabilidade
Eficiência
Manutenibilidade
Portabilidade

Fonte: Adaptado de Simão e Belchior, 2003.

Para efeito dessa pesquisa as características da qualidade de *software* apontadas por Simão e Belchior (2003), serão utilizadas como variáveis na construção do questionário I que avaliará a qualidade de *software*, conforme mostra o anexo I.

2.4. INDICADORES DE SATISFAÇÃO

Conceitualmente, a satisfação dos gerentes de projetos apresenta os mesmos preceitos da satisfação dos consumidores (clientes) de certos produtos ou prestação de serviços visto que, de certa forma, os gerentes de projetos também podem ser considerados clientes internos.

Izard (2007, p. 34) afirma que "a satisfação de um cliente é considerada uma estimativa que acontece a partir da escolha de uma circunstância específica de compra". Então, as pessoas entram num processo de compra com certas expectativas sobre certo produto ou prestação de serviços.

Segundo Kotler (2000, p. 58):

> Satisfação do cliente consiste na sensação de prazer percebido de um produto em relação às expectativas do comprador. Se o desempenho alcançar as expectativas, o cliente ficará satisfeito. Se o desempenho for além das expectativas, o cliente ficará altamente satisfeito ou encantado.

Lee e Kim (1999 apud ROSES, 2007, p. 4) afirmam que a satisfação do cliente deve ser vista tanto sob a perspectiva dos negócios bem como a perspectiva dos usuários.

Castelli (1999, p. 87) afirma que “a meta de toda empresa é satisfazer as necessidades das pessoas com as quais tem compromisso, através da oferta de bens e serviços com a qualidade que elas desejam”.

De acordo com Izard (2007, p. 35):

> A criação de valor para o consumidor gera fidelidade e essa, por sua vez, gera crescimento, lucros e mais valor. Com isso as organizações que conseguem desenvolver programas de retenção do consumidor, como adotar uma política de descontos satisfatória nos preços de seus produtos, apresenta grande chance de ter o cliente comprando continuamente.

É importante para as organizações perceber o valor que o consumidor atribui aos seus produtos e serviços e fazer disto uma busca constante, pois atualmente não se vende produto, vende-se valor (FERREIRA e SGANZERLLA, 2000 apud IZARD, 2003, p. 36).

A satisfação do cliente está associada ao comportamento dele em relação à empresa no que se refere à freqüência com que ele compra. Um cliente que compra com certa regularidade um determinado produto ou marca é considerado um cliente fiel e pode estar proporcionando bastante lucro às empresas (PINHEIRO, 2003, p. 2).

Ferreira e Sganzella (2000, apud IZARD, 2007, p. 36) constatam que o cliente é:

> Influenciado pelo modo como recebe o serviço e como vivencia a relação. Se uma reclamação for resolvida com resultados satisfatórios para ele, a empresa terá boa qualidade em sua avaliação, mas se o consumidor ficar menos satisfeito, ou a obtenção do resultado tiver sido complicado ou demorado, ele irá simplesmente avaliar toda a empresa, seus produtos e serviços como ineficiente e, fatalmente, se sentirá lesado.

Izard (2007, p. 36) afirma que “O valor percebido é a avaliação geral feita pelo consumidor da utilidade de um produto baseado em percepções do que é recebido e do que é dado em troca à empresa”.

A satisfação do usuário é uma avaliação afetiva que um usuário tem em relação a sua experiência (ROSES, 2007, p. 4)

Em linhas gerais, o valor como um conceito muito mais amplo, é percebido pelo cliente por meio de sua experiência de vida e da expectativa que ele acaba criando sobre determinado produto ou prestação de serviço.

Existem alguns modelos de índices nacionais de satisfação dos clientes de determinados produtos ou prestação de serviços. De acordo com Carlos (2004, p. 13) a Suécia foi a pioneira a apresentar um índice de satisfação do consumidor em 1989, denominado de SCSI – *Swedish Customer Satisfaction Index*, sendo Fornell, seu principal idealizador. Esse modelo avalia a qualidade dos produtos ou prestação de serviços através dos antecedentes a satisfação: desempenho percebido pelo cliente com o produto ou prestação de serviço e a expectativa do cliente. Os conseqüentes são as reclamações dos clientes e a fidelização. Na figura 10, pode-se observar a estrutura do modelo Sueco.

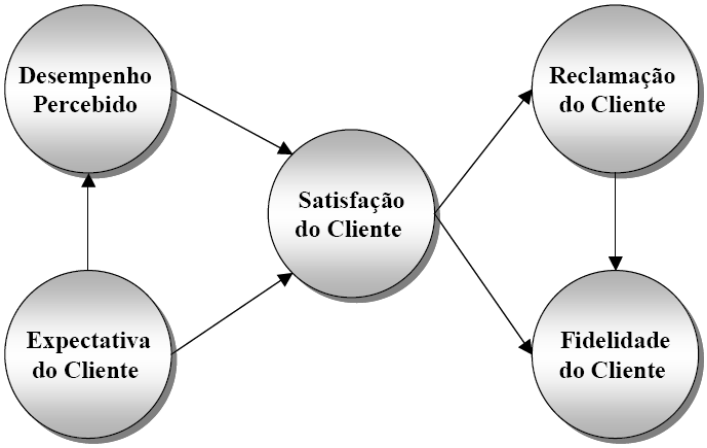

Figura 10. Modelo do índice Sueco de Satisfação.
Fonte: Carlos, 2004, p. 13.

O desempenho percebido é comparado com o valor percebido ou nível percebido de qualidade que o cliente recebe ao adquirir um produto ou serviço relativo ao preço pago. A qualidade pelo valor é um determinador comum utilizado pelos consumidores ao comprar marcas ou categorias similares ou substitutas (González, 2005, p. 28).

Souza (2004, p. 24) afirma que a partir do modelo Sueco, em 1994, surge nos Estados Unidos da América, o Índice Americano de Satisfação do Consumidor – *American Customer Satisfaction Index* (ACSI), que fora desenvolvido em um trabalho conjunto com o Centro Nacional de Pesquisa de Qualidade da Universidade de Michigan – *National Economic Research Associates (NERA) at the University of Michigan Business School*, e a Associação Americana para Controle de Qualidade – *American Society for Quality* (ASQ).

Tanto o SCSI como o ACSI adotam um modelo econométrico de multi-equações para produzir índices ao nível da empresa (PINHEIRO, 2003, p. 16).

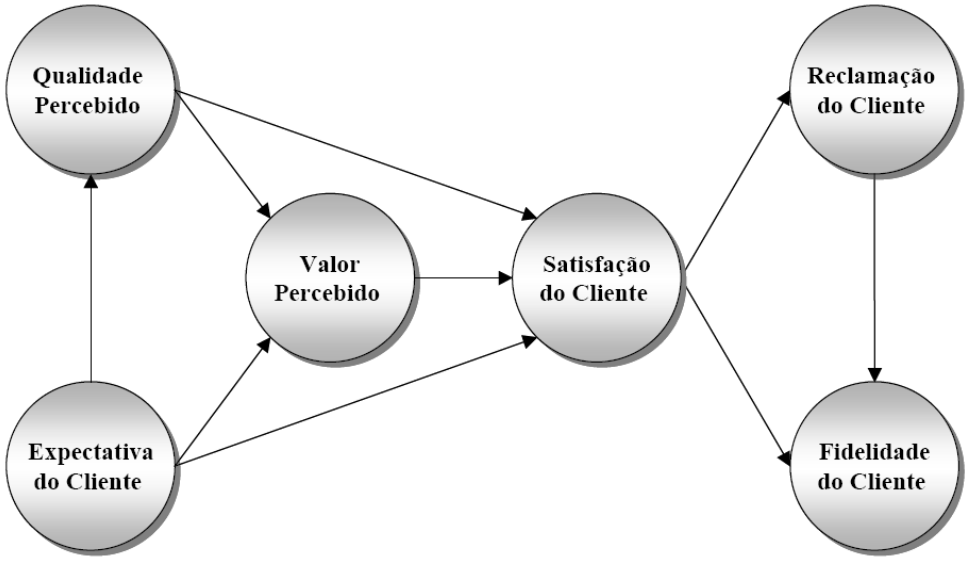

Figura 11. Modelo do índice Norte-Americano de Satisfação.
Fonte: Souza, 2004, p. 25.

Johnson et al. (2001, apud CARLOS, 2004, p.14) afirma que o ACSI prediz que na proporção em que o valor percebido e a qualidade percebida aumentam a satisfação do cliente deveria aumentar.

Com apenas um construto a mais, o modelo americano passou a ser adotado, não somente nos Estados Unidos da América. Souza (2004) afirma que "a adoção do modelo americano tomou uma esfera mundial, sendo inclusive utilizada na Suécia, por um grande número de instituições, substituindo assim o SCSI".

Segundo Pinheiro (2003, p. 20):

> Em 1996, a Comissão Européia, a pedido da EOQ – *European Organizational for Quality*, encomendou ao MFQ – *Movement Français pour la Qualité*, um estudo de viabilidade para desenvolver um índice nacional e um índice Europeu de satisfação do cliente, tomando como base a experiência já conseguida ao nível de diferente países. Este estudo defende entusiasticamente o cálculo de um tal índice Europeu de satisfação do cliente e recomenda a adoção da metodologia Sueca/Americana como ponto de partida do ECSI.

O modelo Europeu inclui a modalidade imagem da empresa como um antecedente da satisfação, influenciando diretamente na expectativa do consumidor, satisfação e fidelidade, e excluindo o construto reclamações dos consumidores como uma conseqüência da satisfação (Souza, 2004, p. 25).

No ECSI, o construto qualidade percebida, conforme afirma González (2005, p. 33) é dividido em qualidade percebida do produto que é a avaliação da experiência recente do consumo de produtos e em qualidade percebida do serviço que é a experiência do consumo recente do serviço associado ao produto.

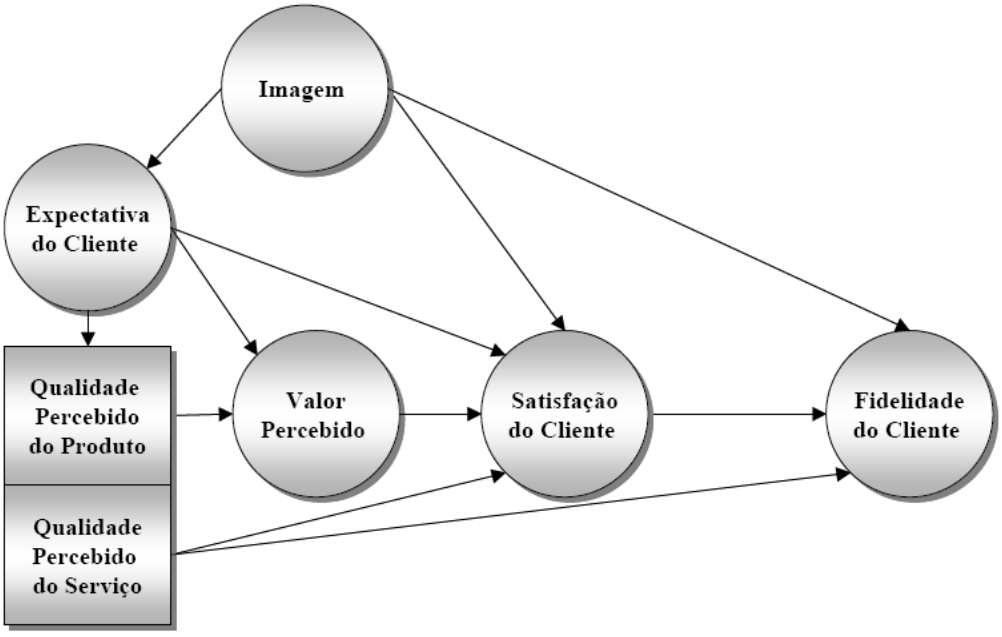

Figura 12. Modelo do índice Europeu de Satisfação.
Fonte: Pinheiro, 2003, p. 21.

Souza (2004, p. 26) afirma que em 1996, a Noruega também constituiu um índice nacional de satisfação denominado Barômetro Norueguês de Satisfação do Consumidor – *Norwgian Customer Satisfaction Barometer* (NCSB), o qual foi aplicado a aproximadamente 43 empresas e 12 indústrias.

Souza (2004) explica que o NCSB é considerado um modelo bastante similar ao ACSI, com exceção da inserção do construto imagem e de seus relacionamentos com a satisfação e fidelidade do cliente.

De acordo com Carlos (2004, p. 15):

> Com base na experiência e aplicações de modelos existentes, o pesquisador e professor da Universidade de Michigan, M. D. Johnson (2001) propôs um novo modelo, rotulando 'novo' modelo Norueguês. O NCSB tem uma série de modificações em relação aos modelos anteriores, pois incorpora novos construtos e eliminam outros, contribuindo e aprimorando os índices nacionais de satisfação dos clientes.

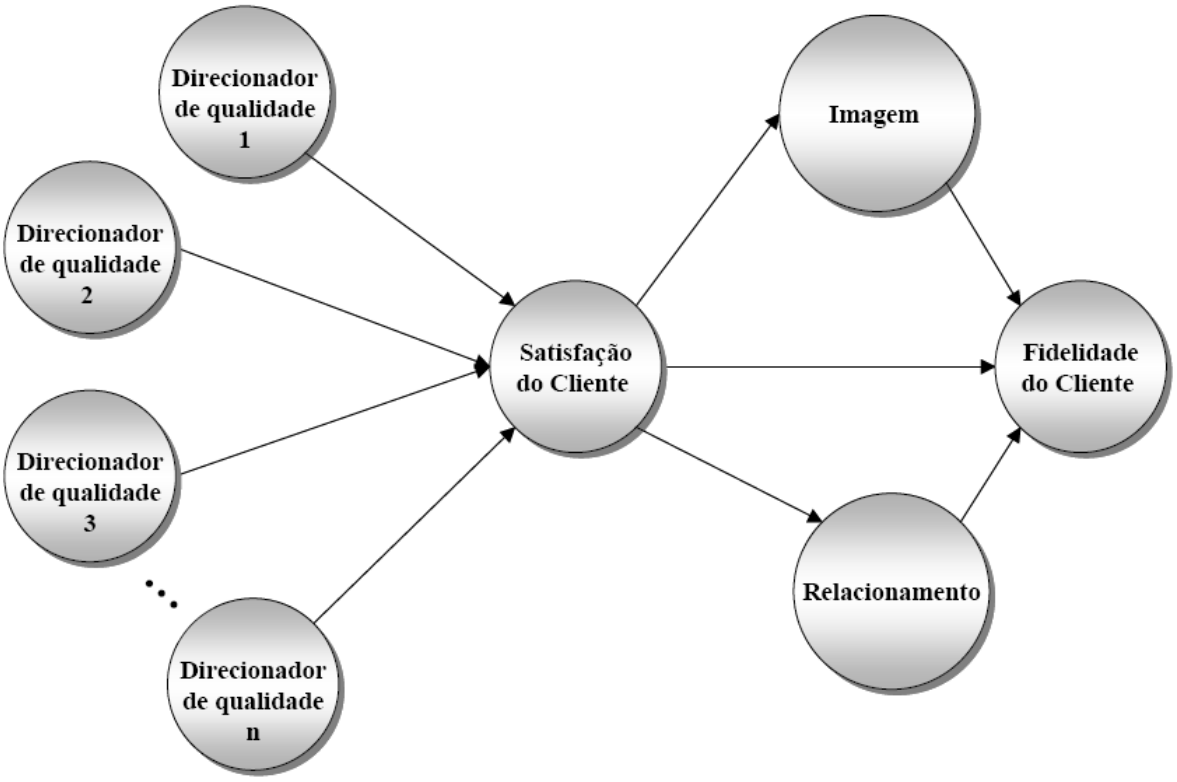

Figura 13. Modelo do índice Norueguês (original) de Satisfação.
Fonte: González, 2005, p. 32.

Na figura 14 se observa os construtos: Direcionadores de Qualidade, Índice de Preço, Gerenciamento de Reclamações, Satisfação do Cliente, Imagem da Empresa, Compromisso Afetivo, Compromisso Calculado e Fidelidade do Cliente (GONZÁLEZ, 2005, p. 34).

Pinheiro (2003, p. 24) cita que as variáveis de compromisso são modeladas como mediadoras dos efeitos da satisfação sobre a fidelidade.

Segundo Mayer et al. (2002, apud GONZÁLEZ, 2005, p. 35) o construto compromisso é multidimensional, cujo antecedente e conseqüente varia por dimensões. Os autores ainda apresentam três formas de compromisso organizacional: compromisso afetivo, compromisso permanente e compromisso normativo, onde os mesmos se relacionam.

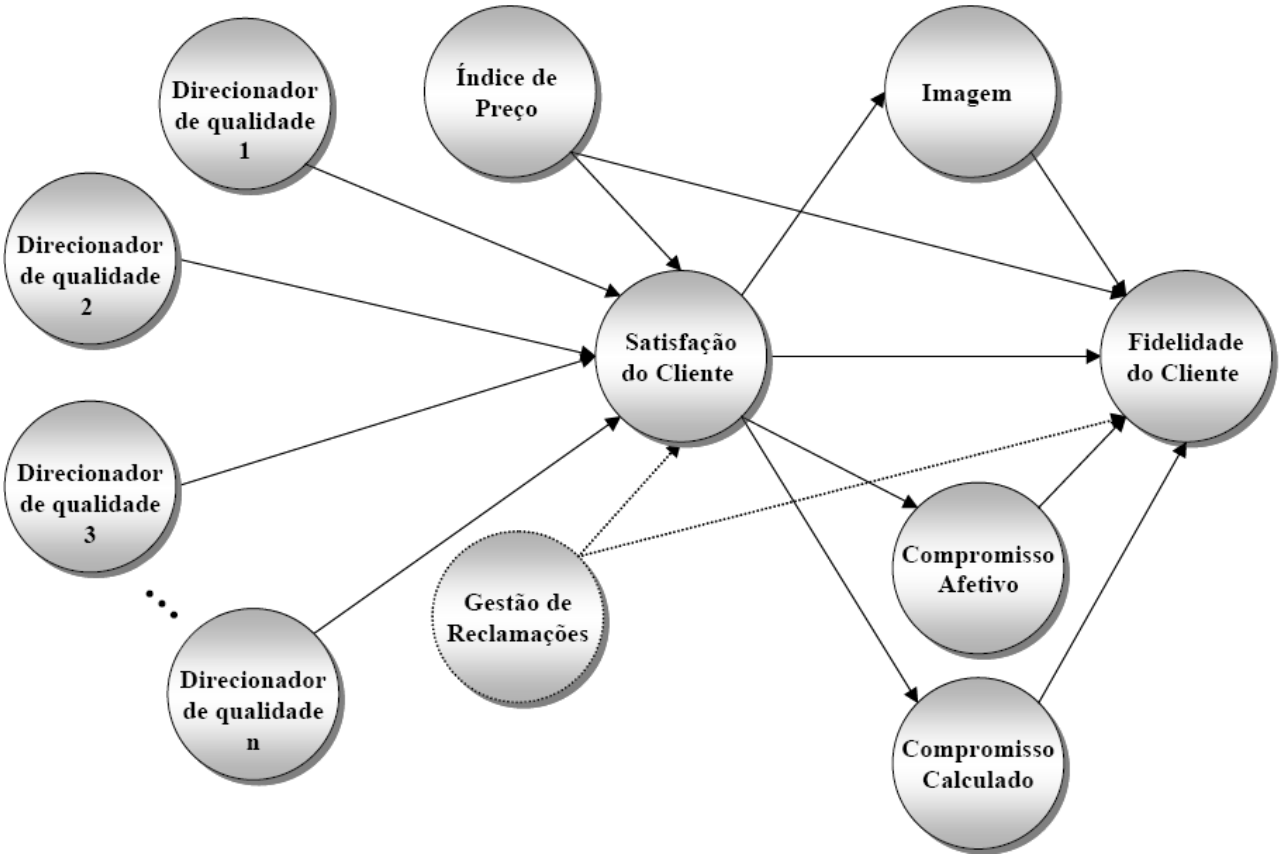

Figura 14. Modelo do índice Norueguês (2001) de Satisfação.
Fonte: González, 2005, p. 36.

González (2005) mostra através da tabela 3 um comparativo dos modelos apresentados anteriormente quanto ao número de construtos que cada um apresenta. Nesta tabela pode-se observar um resumo dos construtos considerados em cada modelo de índice de satisfação apresentado.

Consegue-se observar que o modelo Norueguês apresenta o maior número de construtos que os demais modelos apresentados. Embora esse modelo seja aparentemente o mais completo, com o maior número de construtos, nem sempre é o escolhido para ser usado como referência em pesquisas de índice de satisfação do cliente.

Tabela 3. Construtos Apresentados nos Modelos do Índice de Satisfação do Cliente

CONSTRUTOS		SCSI	ACSI	ECSI	NCSB
Antecedentes	Desempenho Percebido	X			
	Qualidade Percebida		X	X	
	Direcionadores de Qualidade				X
	Expectativas do Cliente	X	X	X	
	Valor Percebido		X	X	X
	Índice de Preço				X
Conseqüentes	Reclamações	X	X		X
	Imagem da Empresa			X	X
	Compromisso Afetivo				X
	Compromisso Calculado				X
	Fidelidade do Cliente	X	X	X	X

Fonte: González, 2005, p. 37.

Para efeito dessa pesquisa o modelo do índice americano de satisfação do consumidor apontado por Souza (2004), será utilizado como base para a construção do questionário II que avaliará a satisfação dos gerentes de projetos em relação à qualidade de *software* produzidos, conforme mostra o anexo II, com exceção do construto fidelidade do cliente, que consideramos inadequado para esta pesquisa.

2.5. TEORIA DOS CONJUNTOS *FUZZY*

Segundo Consenza et al. (2006, p. 2):

> Em 1965, o Professor Lotfi Zadeh formalizou o que, anos depois vinha a ser uma das maiores revoluções no setor matemático: a Lógica *Fuzzy* ou Lógica Nebulosa ou Lógica difusa. Esta teoria trata dos conjuntos não totalmente verdadeiros nem tampouco dos totalmente falsos. Em outras palavras, a lógica *fuzzy* deve ser vista como uma teoria matemática formal para a representação de incertezas.

O advento da teoria *fuzzy* foi causado pela necessidade de um método capaz de expressar de uma maneira sistemática quantidades imprecisas, vagas e mal-definidas (IZARD, 2007, p. 41).

De acordo com Guimarães (2008, p. 24):

> O princípio fundamental da lógica *fuzzy* é o princípio da dualidade, que estabelece que dois eventos opostos possam coexistir. Isto é, um elemento pode pertencer, a certo grau, a um conjunto e, em outro grau, a outro conjunto. Em lógica *fuzzy* os paradoxos podem ser reduzidos a 'meias verdades' ou 'meias mentiras', como se queira a partir de uma lógica multi-valorada. O verdadeiro (1) e o falso (0) são substituídos por graus de pertinência que podem assumir qualquer valor entre 0 e 1. O valor 0,5 descreve um paradoxo.

Os conjuntos *fuzzy* podem então ser vistos como uma generalização da noção de conjunto na qual a função de pertinência pode assumir valores no intervalo [0,1] (FARIA, et al., 2008, p. 5).

Prucole (2006, p. 10) explica que um conjunto *fuzzy* A do universo de discurso _ é definido por uma função de pertinência μ A : __[0,1] . Essa função associa a cada elemento x de _ o grau (x) A μ , com o qual x pertence a A . A função de pertinência (x) A μ indica o grau de compatibilidade entre x e o conceito expresso por A :

- (x) = 1 A μ indica que x é completamente compatível com A ;
- (x) = 0 A μ indica que x é completamente incompatível com A ;
- 0 < (x) < 1 A μ indica que x é parcialmente compatível com A , com grau (x) A μ .

Um conjunto A da teoria dos conjuntos clássica pode ser visto como um conjunto *fuzzy* específico, denominado usualmente de "crisp", para o qual A μ : __{0,1} , ou seja, a pertinência é do tipo "tudo ou nada", "sim ou não", e não gradual como para os conjuntos *fuzzy* (PRUCOLE, 2006, p. 11).

Consenza et al. (2006, p.2) afirmam que a lógica *fuzzy* tem sido fundamental para a consecução de projetos de sistemas especialistas e um importante suporte para tomadas de decisão, em vários segmentos do conhecimento humano.

A teoria dos conjuntos *Fuzzy*, de acordo com Sousa (2007, p. 33), "é em grande parte uma extensão da teoria dos conjuntos tradicionais.". Com o desenvolvimento da lógica *fuzzy* tornou-se possível a aplicação do conceito de verdade parcial, podendo-se apresentar qualquer valor entre os extremos, somente verdadeiro ou somente falso (SOUSA, 2007).

Guimarães (2008, p. 29) explica que "de modo geral, pode-se dizer que, em um problema concreto, muitos números que lá aparecem são idealizações de informações imprecisas envolvendo valores numéricos, como são os casos de frases como em torno de".

Ao medir-se a altura de certa pessoa, por exemplo, o que se obtém é um valor numérico agregado de imprecisões, as quais poderiam ter sido causadas pelos instrumentos de medição utilizados.

Geralmente, opta-se por um valor preciso, um número real, que expresse tal medida. Porém, seria mais prudente que a altura é em torno de "a". Matematicamente, indica-se a expressão *em torno de "a"* por um conjunto *fuzzy* A, cujo domínio é o conjunto de números reais.

Quando necessário, deve ser feito o processo de desfuzzyficação, que pode ser definido como uma função que associa a cada conjunto *fuzzy* um elemento (do conjunto abrupto subjacente) que o represente (IZARD, 2007, p. 44).

Segundo Izard (2007, p. 43):

> A fuzzificação é o processo de transformação da entrada em graus de pertinência produzindo uma interpretação ou adjetivação da entrada, os quais caracterizam o estado do sistema (variáveis de estado), e os normaliza em um universo de discurso padronizado. Estes valores são então fuzzificados, com a transformação da entrada ruim, péssimo, importante etc., em conjuntos nebulosos (triangulares, gaussianas, trapezoidais etc.) para que possam se tornar instâncias de variáveis lingüísticas. A fuzzificação também representa que há atribuição de valores lingüísticos, descrições vagas ou qualitativas (por exemplo: a percepção sobre o estado de uma variável), definidas por funções de pertinências às variáveis de entrada.

A desfuzzyficação não é exatamente o processo inverso da fuzzificação, conforme pode-se constatar a partir de sua definição.

Dado um problema de classificação com n_c classes, onde ω_1, ω_2, ..., ωn_c representam as classes do problema. Prucole (2006, p. 11) afirma que a solução do problema de classificação de dados através de métodos baseados na teoria dos conjuntos *fuzzy* representa cada classe do problema por um conjunto *fuzzy*:

$$\omega_1 = \left\{\left(x, \mu_{\omega_1}(x)\right), x \in \Omega\right\}$$

$$\omega_2 = \left\{\left(x, \mu_{\omega_2}(x)\right), x \in \Omega\right\}$$

$$\vdots$$

$$\omega_{n_C} = \left\{\left(x, \mu_{\omega_{nC}}(x)\right), x \in \Omega\right\}$$

A abordagem do problema de classificação *fuzzy* consiste em calcular a função de pertinência $\mu_{\omega 1}(x)$, $1 \leq j \leq n_{c1}$ partindo da definição de uma discretização (partição) *fuzzy* sobre o universo de cada atributo (PRUCOLE, 2006).

Por meio da representação matemática de um conjunto ordenado de conceitos da linguagem natural através de conjuntos *fuzzy*, Prucole (2006) afirma que a discretização *fuzzy* $\{A_1, A_2, ..., A_n\}$ do universo Ω tal que $\forall x \in \Omega$, $\exists A_i$, $\mu A_i (x) \neq 0$, pode gerar o gráfico 2, que apresenta um exemplo de discretização *fuzzy* para o caso de uma variável dividida em cinco conjuntos *fuzzy*.

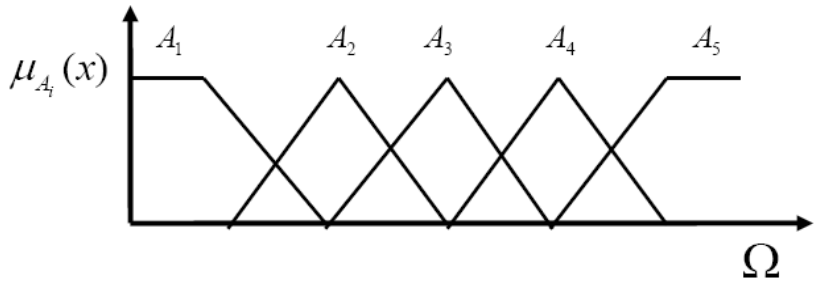

Gráfico 2. Discretização *fuzzy* para uma variável dividida em cinco conjuntos *fuzzy*.
Fonte: Prucole (2006, p. 12).

A representação da informação em diversos níveis de generalização é permitida por meio de diferentes discretizações do universo. Assim, quanto maio o número de conjuntos *fuzzy*, maior será a precisão encontrada.