

# GERÊNCIA DE PROJETOS DE SOFTWARE: MEDIDAS DE QUALIDADE PARA AVALIAÇÃO DE SOFTWARE

Kedna Vieira Martins<sup>1</sup>  
Robson Seleme<sup>2</sup>

## RESUMO

Este trabalho visa apresentar um estudo sobre como medir a qualidade de um software. Além disso, irá relatar métricas para se garantir a qualidade; assinalar algumas normas; estabelecer padrões que devem ser seguidos para alcançar a qualidade; e registrar a diferença entre termos relacionados a qualidade. Software de qualidade é fácil de usar, funciona corretamente, é de fácil manutenção e mantém a integridade dos dados em falhas do ambiente ou outras fora do seu controle. No entanto, em sua grande maioria, os softwares requerem conhecimentos técnicos especiais na sua utilização, são difíceis de alterar para modificar função existente ou implementar novos processos ou facilidades e, para desespero de seus usuários, apresentam falhas sem aviso prévio e não preservam a integridade dos dados. Para ajudar nessa questão a International Organization for Standardization – ISO e a International Electrotechnical Commission – IEC, que são organismos normalizadores com importância internacionalmente reconhecida no setor de software, se uniram para editar normas internacionais conjuntas.

Palavras-chave: Qualidade. Software. Usuários. Normas

## 1.Introdução

Analisando as diferentes referências relativas a gerenciamento de projetos de software, verifica-se que há diferentes visões sobre como estes projetos devem ser gerenciados e estas são centradas em alguns modelos. Assim não basta apenas avaliar as visões de diferentes autores sobre o assunto, mas também os diferentes modelos propostos pelas principais instituições que propõem modelos na área, que são: o PMI – Project Management Institute, o SEI – Software Engineering Institute e ISO – International Standards Organization, além de um modelo comercial amplamente difundido, o RUP – IBM Rational Unified Process.

Ao se deter sobre os diferentes modelos, verifica-se que o gerenciamento de projetos constitui-se em uma tarefa de fundamental importância no processo de gerenciamento de software. O gerenciamento de projeto, no entanto, não é visto como uma etapa clássica do processo de desenvolvimento, uma vez que ele acompanha a todas as etapas tradicionais: Concepção, análise, projeto, desenvolvimento, testes e manutenção.

Segundo a ABNT, na norma técnica NBR 10006, projeto é “processo único, consistindo de um grupo de atividades coordenadas e controladas com datas para início e término, empreendido para alcance de um objetivo conforme requisitos específicos, incluindo

---

<sup>1</sup> Kedna Vieira Martins, Graduada em Sistemas de Informação pela Universidade Estadual de Goiás e Pós-graduada em Gestão de Projetos com ênfase no PMI pela Faculdade Internacional de Curitiba.

<sup>2</sup> Professor Robson Seleme, Engenheiro Civil, Mestre e Doutor em Engenharia de Produção (Universidade Federal de Santa Catarina UFSC/SC)

limitações de tempo, custo e recursos”. De acordo com o Project Management Institute (PMBOK, 2004), projeto é “um empreendimento temporário, planejado, executado e controlado, com objetivo de criar um produto ou serviço único.”

Segundo Pressman (1995), para que um projeto de software seja bem sucedido, é necessário que alguns parâmetros sejam corretamente analisados, como por exemplo, o escopo do software, os riscos envolvidos, os recursos necessários, as tarefas a serem realizadas, os indicadores a serem acompanhados, os esforços e custos aplicados e o sistema a ser seguido. A análise de todos estes parâmetros seria a função típica do gerenciamento de projetos a qual, em geral, se inicia antes do trabalho técnico e prossegue à medida que a entrega do software vai se concretizando.

Os objetivos deste trabalho serão o de estabelecer parâmetros de qualidade de software, buscando relatar métricas que venham a garantir a qualidade; assinalar as normas relacionadas à qualidade; estabelecer padrões que devem ser seguidos para alcançar a qualidade; registrar diferenças entre termos relacionados à qualidade, pois o número de empresas que estão buscando melhorias em seus processos de desenvolvimento de software é crescente. A melhoria não consiste apenas no software em si, mas envolve o relacionamento com o cliente também, por meio de planejamento e redução do número de defeitos nos produtos vendidos. Devido a essa demanda devemos atender aos requisitos da qualidade de um software que fará com que o mesmo atenda as exigências do cliente. Existem hoje no mercado vários softwares que ajudam a garantir essa tão sonhada qualidade.

## **2.Desenvolvimento**

A Engenharia de Software é uma disciplina que aplica os princípios de engenharia com o objetivo de produzir software de alta qualidade a baixo custo. Mas, o que vem a ser qualidade? Qualidade é o grau no qual um conjunto de características inerentes satisfaz aos requisitos.

Do ponto de vista dos fornecedores de software, qualidade não é mais um fator de vantagem no mercado, mas uma condição necessária e pode-se dizer indispensável para que seja possível competir com sucesso.

Para Sanders (1994) “um produto de software apresenta qualidade dependendo do grau de satisfação das necessidades dos clientes sob todos os aspectos do produto”.

As empresas que desenvolvem software passaram a valorizar não somente o seu produto, mas todas as fases que envolvem a sua elaboração e posterior entrega. Assim, a preocupação passou a ser com seu capital intelectual e o comprometimento com a qualidade, que é fator de competitividade e se mostra cada vez mais presente no plano estratégico das organizações. Este novo cenário sócio-econômico mundial, voltado à qualidade, tem forçado as organizações a reverem sua postura frente ao consumidor, ao empregado e à sociedade em geral.

A qualidade tem sido a responsável pelo aumento no grau de competitividade de inúmeras empresas. No entanto, apesar das vantagens associadas à qualidade, existe um grande caminho a se percorrer, no sentido de conscientizar as organizações sobre estas vantagens e sobre as ações que devem ser executadas para alcançar a qualidade. (Storch, 2000)

Para conquistar clientes, diante da competição atual, é preciso encantá-los e seduzí-los. Qualidade intrínseca, preço acessível, flexibilidade, baixo custo de manutenção, valor de revenda, prazo de entrega, rapidez, variedade de opções, cordialidade no atendimento, boas condições de pagamento, imagem no mercado, segurança pessoal e ambiental são algumas entre as muitas possibilidades exploradas como vantagens competitivas.

Diante do mencionado acima, pode-se concluir que a qualidade apresenta-se então como uma necessidade e não como um mero modismo. Qualidade pode ser tudo aquilo que alguém faz ao longo de um processo para garantir que um cliente, interno ou externo à organização, obtenha exatamente aquilo que deseja.

A qualidade de software não deve ser encarada como apenas um diferencial de mercado para que uma dada empresa de software possa conseguir vender e lucrar, mas sim como um pré-requisito que a empresa deve conquistar para conseguir colocar o produto no Mercado Global (Sanders, 1994).

Segundo a norma ISO/IEC 9126 (2000), “qualidade é a totalidade das características de uma entidade que lhe confere a capacidade de satisfazer as necessidades explícitas e implícitas”. Esta definição formal merece alguns complementos, principalmente para definir o que são entidades, as necessidades explícitas e as necessidades implícitas. A entidade é o produto do qual se está falando, que pode ser um bem ou um serviço. As necessidades explícitas são as próprias condições e objetivos propostos pelo produtor. As necessidades implícitas incluem as diferenças entre os usuários, a evolução no tempo, as implicações éticas, as questões de segurança e outras visões subjetivas.

Os fatos explícitos são externados pelo cliente. É o cliente quem define a qualidade, como qualidade dos projetos/processos, qualidade do produto final, manutenções corretivas, adaptativas e introdução de melhorias no produto. Os fatores implícitos dizem respeito aos fatores da qualidade do software, que são percebidos pelos desenvolvedores, que atendem aos fatores explícitos e principalmente à produção econômica do software.

Só que para alcançar o “topo” da qualidade deve-se seguir algumas métricas que são necessárias para analisar qualidade e produtividade do processo de desenvolvimento e manutenção bem como do produto de software construído; e qualificar a performance técnica dos produtos do ponto de vista do desenvolvedor. Ainda na busca da qualidade existem alguns softwares que ajudam nessa empreitada e também algumas normas e padrões que são impostas onde permitirão a construção de um software que garanta a qualidade ao seu usuário final.

Embora a atual norma ISO/IEC 9126 (NBR 13596) enumere as características e subcaracterísticas um software, ela ainda não define como dar uma nota a um software em cada um destes itens. Se você não está familiarizado com o processo de avaliação de software, pode ter dificuldades em tentar utilizar a norma. Se você pretende avaliar um software segundo esta norma, deve tentar atribuir valores (como se fossem notas ou conceitos) a cada uma das subcaracterísticas.

Segundo Moreira (2008) algumas características podem ser realmente medidas, como o tempo de execução de um programa, número de linhas de código, número de erros encontrados em uma sessão de teste ou o tempo médio entre falhas. Nestes casos, é possível utilizar uma técnica, uma ferramenta ou um software para realizar medições. Em

outros casos, a característica é tão subjetiva que não existe nenhuma forma óbvia de realizar a medição.

- Importância de controlar os gastos com software.
- Analisar a performance dos resultados obtidos com o seu desenvolvimento e manutenção.
- Permitir uma padronização.
- Mas como medir o processo de software?
- Como medir, por exemplo, o tamanho das aplicações e a partir daí estimar prazos e custos realistas?
- Como medir a produtividade dos programadores e a partir daí avaliar o progresso no projeto e estimar os recursos necessários?
- Como medir a qualidade do software enquanto este está a ser desenvolvido e partir daí determinar se são necessárias ações corretivas?

Medidas são necessárias para:

- Analisar qualidade e produtividade do processo de desenvolvimento e manutenção bem como do produto de software construído.
- Qualificar a performance técnica dos produtos do ponto de vista do desenvolvedor.
- Medidas funcionais são necessárias para qualificar a performance dos produtos pela perspectiva do usuário.
- Devem ser independentes das decisões do desenvolvimento técnico e implementação.
- Utilizadas para comparar a produtividade de diferentes técnicas e tecnologias.

Possibilita realizar uma das atividades mais fundamentais do processo de gerenciamento de projetos que é o planejamento. A partir deste, identifica-se a quantidade de esforço, o custo e as atividades que serão necessárias para a realização do projeto.

Há bem pouco tempo, ainda segundo Jussara, a única base para a realização de estimativas era a experiência da equipe técnica envolvida no projeto. Que poderia ocasionar:

- Levava a atividades atropeladas ou não realizadas.
- Produtos com deficiência funcional.
- Custo de realização além do previsto.
- Atraso na entrega do produto.

O mercado está cheio de "ferramentas de produtividade". Com certeza o aumento de produtividade mais representativo será obtido quando conseguirmos estabelecer uma sistemática de métricas significativa para os resultados do desenvolvimento de software e efetivamente usá-la.

A medição é algo comum no mundo da engenharia. Mas para engenharia de software está longe se ter uma medição padrão amplamente aceita e com resultados sem nenhum fator subjetivo. Tem-se dificuldade em concordar sobre o que medir e como avaliar o resultado das medições obtidas.

## 2.1 Razões para se medir o software

Medições e métricas auxiliam a entender o processo usado para se desenvolver um produto. O processo é medido a fim de melhorá-lo e o produto é medido para aumentar sua qualidade.

Alguns questionamentos sempre vêm à tona quando é feita uma tentativa de se medir alguma coisa que não tenha sido medida no passado. Isso é o que acontece com a engenharia de software (o processo) e com o software (o produto).

As medidas são uma forma clara de avaliação de produtividade no desenvolvimento de software. Através da obtenção de medidas relativas à produtividade e à qualidade, é possível que metas de melhorias no processo de desenvolvimento sejam estabelecidas como forma de incrementar estes dois fatores.

Métricas de software referem-se a uma ampla variedade de medidas de software. No contexto de gerenciamento de projeto de software, estamos preocupados com métricas de produtividade e de qualidade - medidas do resultado do desenvolvimento de software como uma função do esforço aplicado e medidas da "adequação ao uso" do resultado que é produzido.

Algumas das razões para se medir um software são mencionadas pelo professor Forchesatto (2001), tais como:

- Indicar a qualidade do produto.
- Avaliar a produtividade dos que desenvolvem o produto.
- Determinar os benefícios derivados de novos métodos e ferramentas de engenharia de software.
- Formar uma base para as estimativas. Ajudar na justificativa de aquisição de novas ferramentas ou de treinamentos adicionais. As métricas de software, do ponto de vista de medição, podem ser divididas em duas categorias: medidas diretas e indiretas.

<b>MEDIDAS DIRETAS</b>	<b>MEDIDAS INDIRETAS</b>
- Custo; - Esforço; - Linhas de Código; - Velocidade de Execução; - Memória; - N° de Erros.	- Funcionalidade; - Qualidade; - Complexidade; - Eficiência; - Confiabilidade; - Manutenibilidade.

*Fonte: Gomes (2000)*

As medições de software podem ser organizadas em outras classes, as quais serão definidas a seguir conforme Gomes (2000):

- **Métricas da produtividade**, baseadas na saída do processo de desenvolvimento do software com o objetivo de avaliar o próprio processo;
- **Métricas da qualidade**, que permitem indicar o nível de resposta do software às exigências explícitas e implícitas do cliente;
- **Métricas técnicas**, nas quais encaixam-se aspectos como funcionalidade, modularidade, manutenibilidade, etc.

Sob outra ótica, é possível definir uma nova classificação das medições:

- **Métricas orientadas ao tamanho**, baseadas nas medições diretas da Engenharia de Software;
- **Métricas orientadas à função**, que oferecem medidas indiretas;
- **Métricas orientadas às pessoas**, as quais dão indicações sobre a forma como as pessoas desenvolvem os programas de computador.

As métricas de software se dividem, sob o ponto de vista de aplicação, em duas categorias: métricas de produtividade e de qualidade. As métricas de produtividade se concentram na saída do processo de engenharia de software e métricas de qualidade indicam o quanto o software atende aos requisitos definidos pelo usuário.

O principal problema com que se defronta a engenharia de software é a dificuldade de se medir a qualidade de software. A NBR 13596 fornece um modelo de propósito geral o qual define seis amplas categorias de características de qualidade de software que são, por sua vez, subdivididas em subcaracterísticas: funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade.

Os desenvolvedores fazem os softwares, mas os clientes são os que vão usá-los. Por isso a necessidade urgente de sistematizar formas de evitar os custos elevadíssimos resultantes dos defeitos de software e dos erros não intencionais dos usuários. E isso só será possível se forem priorizadas e atendidas pelo menos quatro características de qualidade de softwares: usabilidade, confiabilidade, funcionalidade e manutenibilidade. (Gomes, 2000)

A ISO/IEC 9126 (NBR 13596) fornece um modelo de propósito geral o qual define seis amplas categorias de características de qualidade de software que são, por sua vez, subdivididas em subcaracterísticas:

CARACTERÍSTICAS	SUBCARACTERÍSTICAS	SIGNIFICADO
<b>Funcionalidade</b> O conjunto de funções satisfazem as necessidades explícitas e implícitas para a finalidade a que se destina o produto?	Adequação	Propõe-se a fazer o que é apropriado?
	Acurácia	Gera resultados corretos ou conforme acordados?
	Interoperabilidade	É capaz de interagir com os sistemas especificados?
	Segurança de acesso	Evita acesso não autorizado, acidental ou deliberado a programas e dados?
	Conformidade	Está de acordo com normas e convenções previstas em leis e descrições similares?

<b>Confiabilidade</b> O desempenho se mantém ao longo do tempo e em condições estabelecidas?	Maturidade	Com que frequência apresenta falhas?
	Tolerância a falhas	Ocorrendo falhas como ele reage?
	Recuperabilidade	É capaz de recuperar dados após uma falha?
<b>Usabilidade</b> É fácil utilizar o software?	Inteligibilidade	É fácil entender os conceitos utilizados?
	Apreensibilidade	É fácil aprender a usar?
	Operacionalidade	É fácil de operar e controlar a operação?
<b>Eficiência</b> Os recursos e os tempos utilizados são compatíveis com o nível de desempenho requerido para o produto?	Comportamento em relação ao tempo	Qual é o tempo de resposta e de processamento?
	Comportamento em relação aos recursos	Quanto recurso utiliza?
<b>Manutenibilidade</b> Há facilidade para correções, atualizações e alterações?	Analisabilidade	É fácil encontrar uma falha quando ocorre?
	Modificabilidade	É fácil modificar e remover defeitos?
	Estabilidade	Há grandes riscos de <i>bugs</i> quando se faz alterações?
	Testabilidade	É fácil testar quando se faz alterações?
<b>Portabilidade</b> É possível utilizar o produto em diversas plataformas com pequeno esforço de adaptação?	Adaptabilidade	É fácil adaptar a outros ambientes sem aplicar outras ações ou meios além dos fornecidos para esta finalidade no software considerado?
	Capacidade para ser instalado	É fácil instalar em outros ambientes?
	Capacidade para substituir	É fácil substituir por outro software?
	Conformidade	Está de acordo com padrões ou convenções de portabilidade?

*Fonte: próprio autor*

O modelo proposto pela ISO/IEC 9126 (NBR 13596) tem por objetivo servir de referência básica na avaliação de produto de software. Além de ter força de norma internacional, ela cobre os aspectos mais importantes para qualquer produto de software.

Um dos grandes erros geralmente cometidos por pessoas e empresas é confundir os conceitos e aplicação dos termos Controle da Qualidade e Garantia da Qualidade. Embora usados de maneira errônea em muitos lugares, ambos os termos têm propósitos totalmente diferentes.

Segundo a NBR ISO 9000 (2005), "qualidade é o grau no qual um conjunto de características inerentes satisfaz aos requisitos". Ou seja, pode-se afirmar que se algum produto ou serviço atende aos requisitos especificados, este mesmo produto ou serviço possui a qualidade desejada.

A qualidade pode ser medida através do grau de satisfação em que as pessoas avaliam determinado produto ou serviço. No entanto, esse produto ou serviço pode ter qualidade para algumas pessoas e para outras nem tanto, ou seja, a qualidade é algo subjetivo.

A qualidade, seja ela usada no contexto de software ou de produtos e serviços, hoje não mais é uma obrigação e um diferencial das empresas. A mesma se tornou um padrão

em qualquer ramo de atividade e indústria sendo assim necessária para garantir a satisfação do cliente. Qualidade hoje em dia, não é apenas um diferencial de mercado para as empresas conseguirem vender e lucrar mais, é um pré-requisito que se deve conquistar para conseguir colocar o produto ou serviço no Mercado Global.

Diante dessa complexidade na definição da palavra qualidade, Pressman (1995) sugere que a qualidade de software seja implementada e não somente uma idéia ou desejo que uma organização venha a ter. Para tanto faz as seguintes colocações sobre qualidade de software (PRESSMAN, 1995, p. 193):

- "Definir explicitamente o termo qualidade de software, quando o mesmo é dito".
- "Criar um conjunto de atividades que irão ajudar a garantir que cada produto de trabalho da engenharia de software exiba alta qualidade".
- "Realizar atividades de segurança da qualidade em cada projeto de software".
- "Usar métricas para desenvolver estratégias para a melhoria de processo de software e, como consequência, a qualidade no produto final".

Sendo assim, a busca constante pela qualidade não se faz apenas no começo do projeto ou no seu final realizando testes, mas sim e um processo que visa abranger toda a engenharia de software bem como a colaboração de todos os membros do time do projeto.

O quadro abaixo mostra a diferença entre estas duas atividades – Garantia da Qualidade e Controle da Qualidade:

<b>Quality Assurance</b>	<b>Quality Control</b>
<ol style="list-style-type: none"><li>1. Garantia da qualidade garante que o processo é definido e apropriado.</li><li>2. Metodologia e padrões de desenvolvimento são exemplos de garantia da qualidade.</li><li>3. Garantia da qualidade é orientada a processo.</li><li>4. Garantia da qualidade é orientada a prevenção.</li><li>5. Foco em monitoração e melhoria de processo.</li><li>6. As atividades são focadas no inicio das fases no ciclo de vida de desenvolvimento de software.</li><li>7. Garantia da qualidade garante que você está fazendo certo as coisas e da maneira correta.</li></ol>	<ol style="list-style-type: none"><li>1. As atividades de controle da qualidade focam na descoberta de defeitos em i específicos.</li><li>2. Um exemplo de controle da qualidade poderia ser: "Os requisitos definidos são os requisitos certos?".</li><li>3. Controle da qualidade é orientado a produto.</li><li>4. Controle da qualidade é orientado a detecção.</li><li>5. Inspeções e garantia de que o produto de trabalho atenda aos requisitos especificados.</li><li>6. As atividades são focadas no final das fases no ciclo de vida de desenvolvimento de software.</li><li>7. Controle da qualidade garante que os resultados do seu trabalho são os esperados conforme requisitos.</li></ol>

Fonte: Pressman (1995)



Com isso, pode-se afirmar que o teste de software é uma das atividades de controle da qualidade, ou seja, o teste de software é orientado a produto e está dentro do domínio do controle da qualidade.

Entidades de normalização de mais de noventa países fazem parte do grupo *International Organization for Standardization* (ISO – Organização Internacional de Normalização), entre os quais o Brasil, através da Associação Brasileira de Normas Técnicas (ABNT) (Antonioni, 1995).

Para realizar o acompanhamento da qualidade são estabelecidos procedimentos, parâmetros e medidas para verificar se um produto está dentro de limites aceitáveis. No caso do produto de software, essas medidas são difíceis de serem definidas, pois ele possui características que o diferenciam dos outros produtos. Não existem medidas que possam afirmar que um produto de software está totalmente correto ou não. No máximo pode-se dizer que ele funcionou em todas as circunstâncias em que foi testado (Antonioni, 1995).

Surgiu então, a necessidade de criar normas para estabelecer uma forma de avaliação da qualidade de produtos de software a partir de características que devem ser observadas para que ele seja considerado um produto de qualidade.

As principais normas nacionais e internacionais existentes na área de software são:

- ISO 9126 – Características da qualidade de produtos de software.
- NBR 13596 – Versão brasileira da ISO 9126.
- ISO 14598 – Guias para a avaliação de produtos de software, baseados na utilização prática da Norma ISO 9126.
- ISO 12119 – Características de qualidade de pacotes de software.
- IEEE P1061 – *Standard for Software Quality Metrics Methodology* (Normas para Métricas de Qualidade de Software).
- ISO 12207 – *Software Life Cycle Process* (Norma para a qualidade do processo de desenvolvimento de software).
- NBR ISO 9001 – Sistemas de qualidade – modelo para garantia de qualidade em projeto, desenvolvimento, instalação e assistência técnica (processo).
- NBR ISO 9002 – Modelo para garantia da qualidade em produção, instalação e serviços.
- NBR ISO 9003 – Gestão de qualidade e garantia da qualidade. Aplicação da norma ISO 9000 para o processo de desenvolvimento de software.
- NBR ISO 10011 – Auditoria de Sistemas de Qualidade (processo).

O Subcomitê de Software (SC7) do Comitê Técnico Conjunto (JTC1) da ISO e IEC vem trabalhando na elaboração de normas e relatórios técnicos que permitam especificar e avaliar a qualidade dos produtos de software. Essas normas e relatórios são definidos pelos seguintes documentos:

- Qualidade de produto de software:
  - ISO/IEC 9126-1: Modelo de Qualidade;
  - ISO/IEC 9126-2: Métricas Externas;
  - ISO/IEC 9126-3: Métricas Internas;
  - ISO/IEC 9126-4: Métricas da Qualidade em Uso.
- Avaliação de produtos de software:
  - ISO/IEC 14598-1: Visão Geral;
  - ISO/IEC 14598-2: Planejamento e Gestão;
  - ISO/IEC 14598-3: Processo para Desenvolvedores;
  - ISO/IEC 14598-4: Processo para Adquirentes;
  - ISO/IEC 14598-5: Processo para Avaliadores;
  - ISO/IEC 14598-6: Documentação de Módulos de Avaliação.
- Teste e requisitos de qualidade em pacotes de software:
  - NRO ISO/IEC 12119 – Pacotes de Software – Teste e requisitos de qualidade.

A qualidade do produto depende da qualidade do processo. Para que se possa verificar a qualidade do produto, é necessário que se tenha um modelo para que se faça a comparação. O produto ter qualidade, não quer dizer que seja perfeito, mas que estejam dentro das especificações e atendam ao usuário. Para alcançar a qualidade do produto, é necessário dividir os processos para conquistar o produto desejado. Isto é, verificar e corrigir cada etapa do processo. Através de dados quantitativos que descrevam a realidade do processo. A garantia da qualidade é obtida através das atividades de verificação e validação ao desenvolvimento de software. O processo de garantia relaciona a qualidade do produto e do processo.

Antes de liberar o produto para o usuário é necessário realizar as revisões. A revisão proporciona a possibilidade de se corrigir os erros antes de disponibilizar o produto ao usuário. As normas permitem avaliar e especificar a qualidade dos produtos de software.

Segundo Gomes (2000) a avaliação da qualidade de software é feita com um dos seguintes objetivos:

- para aprimorar o processo de desenvolvimento e conseqüentemente melhorar a qualidade do produto resultante. Neste caso ela é feita pelas empresas que desenvolvem o software;
- para avaliar a qualidade do produto visando emitir documento oficial sobre a qualidade de um software e sua conformidade em relação a uma norma ou padrão. Essas avaliações utilizam como referência normas internacionais e são feitas por organismo de certificação;

- para aquisição de software objetivando escolher o produto mais adequado dentre um conjunto de softwares selecionados. Este processo é feito por pessoas ou organizações que adquirem software.

Desejando selecionar o software de melhor qualidade e mais adequado às suas necessidades, as organizações podem adotar um, ou a combinação, dos seguintes procedimentos: exigir que as empresas apresentem certificado emitido por organismo oficial de certificação; avaliar o software *in home* utilizando equipe multidisciplinar com especialistas da área de tecnologia e especialistas da área que se utilizará do software; contratar empresa para efetuar em seu nome a avaliação dos produtos ofertados segundo os requisitos internacionais de qualidade e demais requisitos do contratante.

## **Conclusão**

A qualidade de software pode ser medida através de suas características externas e internas. Os fatores internos de qualidade não são facilmente percebidos pelo usuário, pois são inerentes à concepção do sistema. Os fatores internos são meios para atingir os fatores externos de qualidade. Considera-se fatores externos aqueles que podem ser facilmente percebidos pelo usuário, tais como facilidade de uso, eficiência, portabilidade e robustez.

É muito importante considerar que não apenas o produto final é passível de avaliação qualitativa. A qualidade do processo de desenvolvimento é diretamente ligada à qualidade do produto. A avaliação deve começar já nas fases iniciais do desenvolvimento do software, pois isso significa economia, uma vez que erros identificados em fases tardias requerem muito esforço e implicam em altos custos na sua resolução.

Ainda é elevado o número de empresas brasileiras de software que não adotam técnicas para melhoria da qualidade de seus produtos, mas não há como negar que as empresas desenvolvedoras de software de qualidade são mais competitivas, o que é muito importante para sua sobrevivência em um mercado cada vez mais globalizado. A abertura de mercados externos para os softwares brasileiros exigirá uma nova postura frente à qualidade, beneficiando o mercado interno que acabará por exigir softwares melhores a menor preço.

É preciso que os esforços em melhoria da qualidade de software tenham seu foco não no produto apenas (fazer software melhor), mas principalmente no processo (fazer melhor o software) e no cliente (fazer software mais fácil de usar). Para que a evolução dos modelos possa reduzir a quantidade de falhas no desenvolvimento de software é importante o investimento em métodos de gerenciamento mais amplos e na profissionalização das organizações.

Devido ao aumento da competitividade e a preocupação em oferecer ao mercado softwares que atendam as expectativas de seus clientes, os desenvolvedores de software tem buscado aplicar os conceitos de qualidade em seus produtos. Dado os fatos mencionados nota-se que nos dias de hoje a qualidade dos softwares desenvolvidos tem aumentado significativamente. Algumas certificações existentes no mercado têm dado uma atenção especial nos processos de qualidade e mobilizado as organizações a adotarem este tipo de processo em todos seus produtos. A automatização das etapas de testes de software também tem estado presente cada vez mais nas organizações, auxiliando o processo final de testes. É esperado que as organizações comecem a incluir em seu processo de

desenvolvimento, o processo de qualidade de software, não apenas no momento que o produto foi finalizado ou desenvolvido, mas desde o início de sua concepção.

## Referências

ABNT – ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 13596**: Tecnologia de informação – Avaliação de produto de software – Características de qualidade e diretrizes para o seu uso. Rio de Janeiro, 1996.

\_\_\_\_\_. **NBR ISO 9000**: Sistemas de Gestão de Qualidade: Fundamentos e Vocabulário. Rio de Janeiro, 2005.

\_\_\_\_\_. **NBR ISO 9126**: Características da qualidade de produtos de software. Rio de Janeiro, 2003.

ANTONIONI, J. A. Rosa. **Qualidade em software**: manual de aplicação da ISO-9000. São Paulo: Makron Books, 1995.

FORCHESATTO, André Luis. **Introdução à Engenharia de Software**. Santa Catarina, 2001. Disponível em: <<http://pt.scribd.com/doc/55494552/12/Capitulo-2-%E2%80%93-Projeto-de-software-14>>. Acesso em: 16 de maio de 2011.

GOMES, Nelma da Silva. **Qualidade de software**: Uma necessidade. Brasília, 2000. Disponível em: <[http://www.fazenda.gov.br/ucp/pnafe/cst/arquivos/Qualidade\\_de\\_Soft.pdf](http://www.fazenda.gov.br/ucp/pnafe/cst/arquivos/Qualidade_de_Soft.pdf)>. Acesso em: 13 de dezembro de 2010.

ISO/IEC 9126. **Tecnologia de informação**. Modelo de qualidade para produto de software. Rio de Janeiro, 2000. Disponível em: <[http://www.abnt.org.br/index\\_body.htm](http://www.abnt.org.br/index_body.htm)>. Acesso em: 07 de fevereiro de 2011.

MOREIRA, Jussara. **Métricas de Software**. Pernambuco, 2008. Disponível em: <<http://www.facape.br/jussaramoreira/index.php>>. Acesso em: 16 de maio de 2011.

PRESSMAN, Roger S. **Engenharia de Software**. Tradução José Carlos Barbosa dos Santos. São Paulo: Makron Books, 1995.

PROJECT Management Institute. **Um Guia do Conjunto de Conhecimentos em Gerenciamentos de Projetos**: Guia PMBOK. Terceira Edição. Local Pennsylvania: Four Campus Boulevard, 2004.

SANDERS, Joc. **Qualidade de software**. 1994. Disponível em: <<http://www.geekbrasil.com.br>>. Acesso em: 20 de dezembro de 2010.

STORCH, Mirian Mirdes. **Proposta de avaliação da qualidade de produtos de software utilizando a norma ISO/IEC 9126**. 2000. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau, 2000.