

SUMÁRIO

Normas de Qualidade de Software	2
OBJETIVOS:	2
Garantia de Qualidade de Software (Software Quality Assurance – SGA)	4
Abordagens Formais Para a Garantia de Qualidade de Software (SGA).....	5
Princípios e Modelos da Norma ISO 9000	8
NBR ISO 9126.....	9
NBR ISO 12119.....	22
REFLEXÃO:	37
ATIVIDADE (EM GRUPO)	38
REFERÊNCIAS	39

Normas de Qualidade de Software

Na aula anterior foram vistos os conceitos de qualidade de software e pode-se perceber que os mesmos são extremamente importante, visto que, com a grande demanda de software, os clientes encontram-se cada vez mais exigentes e o mercado cada dia mais competitivo.

Apesar da necessidade da qualidade nos softwares, garantir isso nem sempre é uma tarefa fácil para as empresas desenvolvedoras de software, dentro desse contexto, é preciso analisar a atividade SQA – garantia de qualidade de software e as normas de qualidade de software ISO/NBR 9126 e ISO/NBR 12119.

OBJETIVOS:

- Descrever os passos necessários para realizar a garantia de qualidade de software estatística.

- Discorrer sobre a diferença entre confiabilidade de software e segurança de software.
- Abordar os princípios da ISO 9000 e suas vertentes.
- Discorrer sobre os modelos ISO 9000 para sistemas de garantia da qualidade.

Garantia de Qualidade de Software (Software Quality Assurance – SGA)

Segundo Pressman (2002), Software Quality Assurance é um padrão sistemático e planejado de ações que são exigidas para garantir a qualidade de software. Essas ações englobam:

- **Aplicação de Métodos Técnicos:** ajudam o analista a conseguir uma especificação de elevada qualidade e o projetista a desenvolver um projeto de elevada qualidade.
- **Realização de Revisões Técnicas Formais:** para avaliar a qualidade da especificação e do projeto.
- **Atividades de Teste de Software:** para ajudar a garantir uma detecção de erros efetiva.

- **Aplicação de Padrões e Procedimentos Formais** no processo de engenharia de software.
- **Processo de Controle de Mudanças:** atividade que faz parte do gerenciamento de configuração de software.
- **Mecanismos de Medição:** para ser possível rastrear a qualidade de software
- **Anotação e Manutenção de Registros:** procedimentos para a coleta e disseminação de informações de garantia de qualidade de software.

Abordagens Formais Para a Garantia de Qualidade de Software (SGA)

As principais abordagens são comentadas a seguir:

- **Prova de Corretitude** — se o modelo dos requisitos (especificação) e a linguagem de programação podem ser representadas de uma maneira rigorosa, deveria ser possível aplicar provas matemáticas de corretitude para demonstrar que o programa atende exatamente suas especificações.

- **Garantia Estatística de Qualidade** — implica coletar e categorizar informações sobre os defeitos do software; tentar descobrir a causa de cada defeito; isolar 20% das causas (princípio de Pareto); corrigir os problemas que causaram os defeitos
- **Processo Sala Limpa (Cleanroom)** — combina a prova de corretude e a Garantia Estatística de Qualidade para melhorar a qualidade do produto software

A necessidade de qualidade de software é reconhecida por praticamente todos os gerentes e profissionais da área, porém muito poucos estão interessados em estabelecer funções de Garantia de Qualidade de Software Formais. Algumas razões para essa aparente contradição:

1. os gerentes relutam em incorrer em custos extras logo de cara
2. os profissionais acham que estão fazendo absolutamente tudo o que precisa ser feito
3. ninguém sabe onde colocar essa função organizacionalmente
4. todos querem evitar a burocracia que, segundo entendem, a Garantia de Qualidade de Software introduzirá no processo de engenharia de software.

Alguns aspectos positivos da garantia de qualidade de software que podemos mencionar são:

- o software terá menos defeitos latentes resultando em redução do esforço e do tempo gasto durante as atividades de teste e manutenção
- a maior confiabilidade resultará em maior satisfação do cliente
- os custos de manutenção podem ser reduzidos
- o custo do ciclo de vida global do software é reduzido

Pressman (2004), destaca alguns passos necessários para realizar a SQA estatística e criar um processo adaptativo de engenharia de software no qual são feitas modificações para aprimorar os elementos do processo que promovem erro:

- Coletar e categorizar os defeitos de software encontrados.

- Rastrear o defeito até sua causa subjacente.
- Considerar que 20% do código têm 80% dos defeitos.
- Corrigir os problemas que causaram os defeitos.

Princípios e Modelos da Norma ISO 9000

Conforme já comentamos anteriormente, para se alcançar os objetivos de produtividade e qualidade é necessário que o software seja eficiente, definido, gerenciado, medido e controlado, o que nem sempre é uma tarefa fácil. Para auxiliar nessas tarefas existem norma e modelos padrões que possuem como principal objetivo a obtenção da qualidade. O conjunto de Normas da ISO 9000 é um exemplo disso. A seguir comentaremos sobre duas normas de qualidade, sendo uma voltada para produto de software (você se lembra o que é um produto de software, não é?) e outra voltada para a avaliação de pacotes de software, ou software de prateleiras, como alguns preferem chamar.

NBR ISO 9126

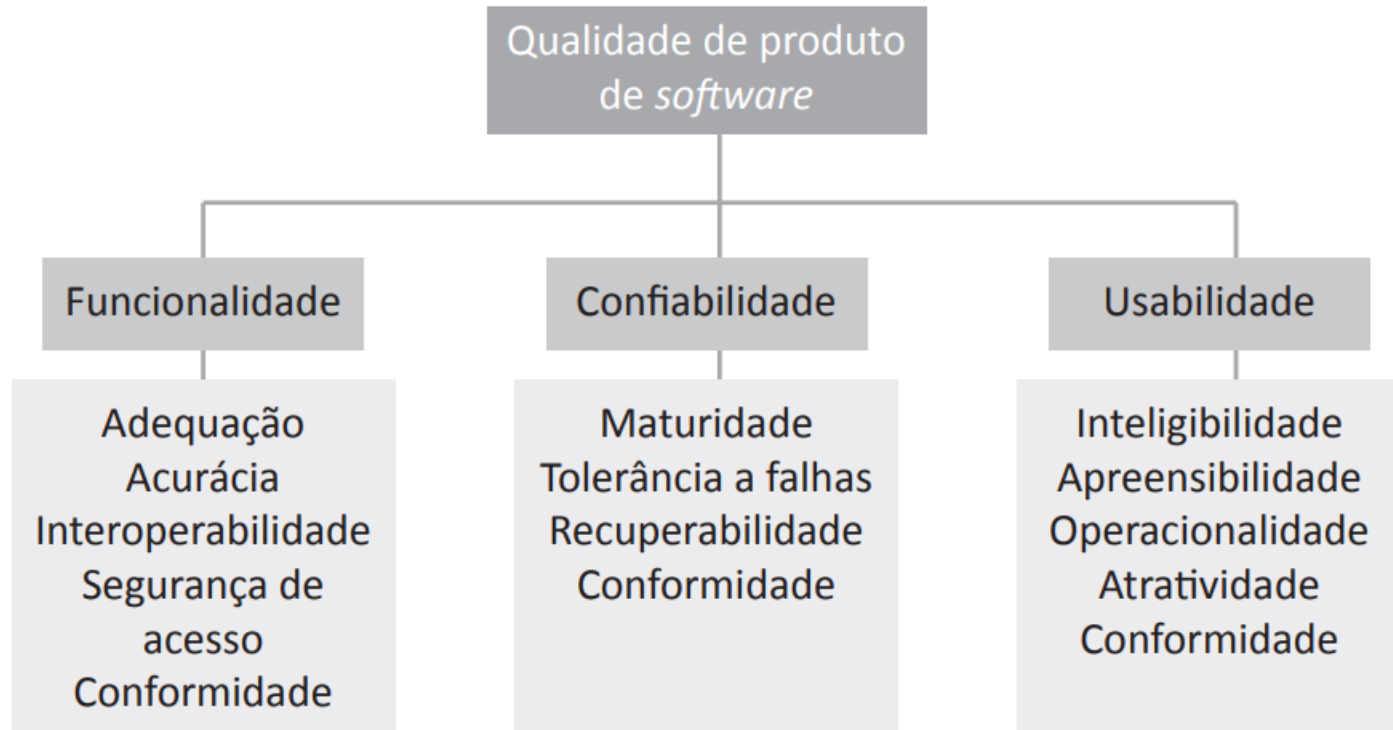
Como já comentamos anteriormente, a qualidade é um conceito muito importante para o software. Durante muito tempo, foram propostos modelos de qualidade, mas a confiabilidade era o único meio de se avaliar a qualidade de um software. Surgiu então a necessidade de um modelo padronizado. Por essa razão, o comitê técnico da ISO (International Organization for Standardization) e IEC (International Electrotechnical Commission), iniciaram em 1985 o desenvolvimento da Norma Internacional ISO/IEC 9126, que estabelece características baseadas no conceito de qualidade para um produto de software.

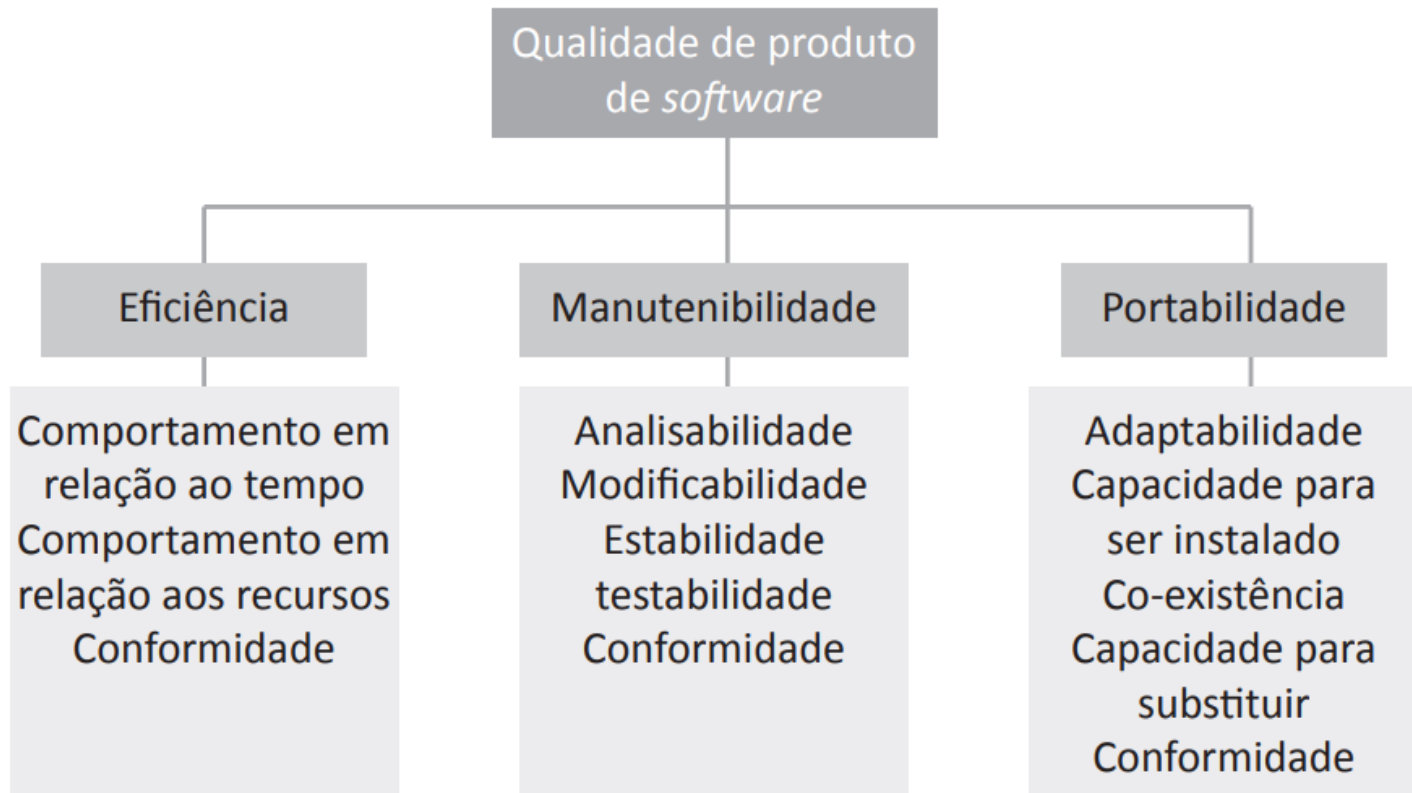
A Norma ISO/IEC 9126:1991 ou NBR 13596:1996 apresenta a padronização mundial do Software como Produto considerado como um “Software de Qualidade”.

Esta norma fornece um modelo de propósito geral o qual define 6 categorias de características de qualidade de software que são, por sua vez, divididas em

subcaracterísticas. As subcaracterísticas podem ser avaliadas por um conjunto de métricas.

A ISO/IEC 9126 e foi publicada em 1991, sendo uma das mais antigas da área de qualidade de software e já possui sua tradução para o Brasil, publicada em agosto de 1996 como NBR 13596. Estas normas listam o conjunto de características que devem ser verificadas em um software para que ele seja considerado um “software de qualidade”. São seis grandes grupos de características, cada um dividido em algumas subcaracterísticas, os quais são descritos abaixo:





Segundo a ISO/IEC 9126-1, as definições das características e subcaracterísticas de qualidade interna e externa são:

Funcionalidade: capacidade do produto de software de prover funções que atendam às necessidades explícitas e implícitas, quando o software estiver sendo utilizado sob condições específicas.

- **Adequação:** capacidade do produto de software de prover um conjunto apropriado de funções para tarefas e objetivos do usuário especificados.
- **Acurácia:** capacidade do produto de software de prover, com o grau de precisão necessário, resultados ou efeitos corretos ou conforme acordados.
- **Interoperabilidade:** capacidade do produto de software de interagir com um ou mais sistemas especificados.

- **Segurança de Acesso:** capacidade do produto de software de proteger informações e dados, de forma que pessoas ou sistemas não autorizados não possam lê-los nem modificá-los e que não seja negado o acesso às pessoas ou sistemas autorizados.
- **Conformidade:** capacidade do produto de software de estar de acordo com normas, convenções ou regulamentações previstas em leis e prescrições similares relacionadas à funcionalidade.

Confiabilidade: capacidade do produto de software de manter um nível de desempenho especificado, quando usado em condições específicas.

- **Maturidade:** capacidade do produto de software de evitar falhas decorrentes de defeitos no software.

- **Tolerância a Falhas:** capacidade do produto de manter um nível de desempenho especificado em casos de defeitos no software ou de violação de sua interface especificada.
- **Recuperabilidade:** capacidade do produto de software de restabelecer seu nível de desempenho especificado e recuperar os dados diretamente afetados no caso de uma falha.
- **Conformidade:** capacidade do produto de software de estar de acordo com normas, convenções ou regulamentações relacionadas à confiabilidade.

Usabilidade: capacidade do produto de software de ser compreendido, aprendido, operado e atraente ao usuário, quando usado sob condições específicas.

- **Inteligibilidade:** capacidade do produto de software de possibilitar ao usuário compreender se o software é apropriado e como ele pode ser usado para tarefas e condições de uso específicas.
- **Apreensibilidade:** capacidade do produto de software de possibilitar ao usuário aprender sua aplicação.
- **Operacionalidade:** capacidade do produto de software de possibilitar ao usuário operá-lo e controlá-lo.
- **Atratividade:** capacidade do produto de software de ser atraente ao usuário. [
- **Conformidade:** capacidade do produto de software de estar de acordo com normas, convenções, guias de estilo ou regulamentações relacionadas à usabilidade.

Eficiência: capacidade do produto de software de apresentar desempenho apropriado, relativo à quantidade de recursos usados, sob condições específicas.

- **Comportamento em relação ao tempo:** capacidade do produto de software de fornecer tempos de resposta e de processamento, além de taxas de transferência, apropriados, quando o software executa suas funções, sob condições estabelecidas.
- **Comportamento em relação aos recursos:** capacidade do produto de software usar tipos e quantidades apropriados de recursos, quando o software executa suas funções, sob condições estabelecidas.
- **Conformidade:** capacidade do produto de software de estar de acordo com normas e convenções relacionadas à eficiência.

Manutenibilidade: capacidade do produto de software ser modificado. As modificações podem incluir correções, melhorias ou adaptações do software devido a mudanças no ambiente e nos seus requisitos ou especificações funcionais.

- **Analisabilidade:** capacidade do produto de software de permitir o diagnóstico de deficiências ou causas de falhas no software, ou a identificação de partes a serem modificadas.
- **Modificabilidade:** capacidade do produto de software que uma modificação específica seja implementada.
- **Estabilidade:** capacidade do produto de software de evitar efeitos inesperados decorrentes de modificações no software.
- **Testabilidade:** capacidade do produto de software de permitir que o software, quando modificado, seja validado

- **Conformidade:** capacidade do produto de software de estar de acordo com normas ou convenções relacionadas à manutenibilidade.
- **Portabilidade:** capacidade do produto de software de ser transferido de um ambiente para outro.
- **Adaptabilidade:** capacidade do produto de software de ser adaptado para diferentes ambientes especificados, sem necessidade de aplicação de outras ações ou meios além daqueles fornecidos para essa finalidade pelo software considerado.
- **Capacidade de ser instalado:** capacidade do produto de software ser instalado em um ambiente especificado.
- **Coexistência:** capacidade do produto de software de coexistir com outros produtos de software independentes, em um ambiente comum, compartilhando recursos comuns.

- **Capacidade para substituir:** capacidade do produto de software de ser usado em substituição a outro produto de software especificado, com o mesmo propósito e no mesmo ambiente.
- **Conformidade:** capacidade do produto de software de estar de acordo com normas ou convenções relacionadas à portabilidade.

Na norma cada característica é refinada em um conjunto de subcaracterísticas e cada subcaracterística é avaliada por um conjunto de métricas. A norma não fornece métricas e nem métodos para medição, pontuação ou julgamento.

- Cada organização pode estabelecer seus próprios modelos de processo de avaliação e métodos para a criação e validação de métricas relacionadas com as características.

- Também é necessário estabelecer níveis de pontuação e critérios específicos para a organização ou para a aplicação.

NBR ISO 12119

A Norma NBR 12119 foi publicada em 1996 com o objetivo de fornecer subsídios para a avaliação de pacotes de software.

O escopo da norma NBR 12119 refere-se a pacotes de software, na forma oferecida no mercado, e não aos processos de desenvolvimento e fornecimento de software.

O que é considerado um pacote de software:

Pacote de Software: trata-se de um produto de software que envolve um conjunto completo e documentado de programas fornecidos a diversos usuários para uma aplicação ou função genérica.

Também conhecido como “software de prateleira”.

Um Pacote de Software envolve todos os componentes do produto disponíveis aos usuários, tais como documentação, manual de instruções e guia para instalação.

Exemplos:

- Processadores de Texto
- Planilhas Eletrônicas
- Gerenciadores de Banco de Dados
- Software Gráficos
- Programas para Funções Técnicas ou Científicas
- Programas Utilitários

Os usuários dessa norma são normalmente fornecedores, órgãos de certificação, laboratórios de teste, auditores e compradores/usuário.

A norma de avaliação de pacote de software NBR ISO 12119 faz a avaliação do pacote de software como um todo, ou seja, deve ser avaliado a parte externa da caixa do pacote de software e também a parte interna que nesse caso seria o software em si. A avaliação externa é também de supra importância, uma vez que na caixa do software

deve conter todas as informações que o cliente necessita saber antes de executar a compra do software.

A norma é dividida em duas partes:

- **Requisitos de Qualidade**, onde apresenta a descrição do produto, manual do usuário e programas e dados (que é o software propriamente dito) e,
- **Instruções para teste**, onde apresenta as principais atividades de teste a serem realizadas, como deve ser feito o registro destes testes e como deve ser elaborado o relatório final dos testes.

A seguir, apresentamos detalhadamente cada uma das etapas da Norma NBR ISO 12119.

Requisitos de qualidade:

- **Descrição do Produto:** Documento exposto as propriedades de um pacote de software, com o principal objetivo de auxiliar os potenciais compradores na avaliação da adequação do produto antes de sua aquisição. A descrição do produto fornece informações sobre a documentação do usuário, programas e, se existirem, sobre os dados.

A descrição do produto inclui as principais propriedades do pacote e é um documento disponível ao usuário, independente da aquisição do produto.

A descrição do produto é dividida em: Requisitos gerais, identificações e indicações, e declarações sobre as características de funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade.

É importante ressaltar aqui que a avaliação sobre essas características é feita com base no conteúdo externo da caixa do pacote do software, quando o cliente ainda não comprou o software.

Vamos detalhar com mais cuidado cada uma das partes da Descrição do produto:

Requisitos gerais

A descrição deve ser inteligível, completa, bem organizada e bem apresentada para auxiliar os compradores em potencial na avaliação da adequação do produto às suas necessidades, antes de comprá-lo. Deve ser livre de inconsistências internas e é interessante que cada termo tenha um único significado.

Identificações e Indicações

O documento de descrição de produto deve possuir uma única identificação, essa indicação do produto deve ter no mínimo o nome do produto e uma versão ou data. A identificação do fornecedor deve conter o nome e o endereço de, no mínimo, um fornecedor.

As tarefas que podem ser realizadas utilizando o produto devem ser identificadas. A descrição do produto pode fazer referência aos documentos de requisitos com os quais o produto está em conformidade. Nesse caso as edições relevantes devem ser identificadas.

Os requisitos de hardware e software para colocar o produto em uso devem ser especificados, incluindo nomes de fabricantes e identificação do tipo de todos os componentes. Se a descrição do produto faz referências a interfaces com outros produtos, as interfaces ou produtos devem ser identificados.

Todo os itens a serem entregues (componentes físico do produto) devem ser identificados, incluindo todos os documentos impressos e todos os meios de armazenamento de dados. Deve ser declarado se a instalação do produto pode ou não ser conduzida pelo usuário.

Deve ser declarado se o suporte para operação do produto é oferecido ou não. Deve ser declarado se a manutenção é oferecida ou não. Em caso afirmativo deve ser declarado especificamente o que é incluído

Declarações sobre Funcionalidade

A descrição do produto deve fornecer uma visão geral das funções disponíveis para o usuário do produto, os dados necessários e as facilidades oferecidas. Nem toda função disponível para o usuário necessita ser mencionada, e nem todos os detalhes de como uma função é chamada necessitam ser descritos.

Declarações sobre Confiabilidade

A descrição do produto deve incluir informações sobre procedimentos para preservação de dados. Uma declaração do tipo: “é possível fazer backup através de funções do sistema operacional” é suficiente na descrição do produto.

Declarações sobre Confiabilidade

Convém que propriedades adicionais do produto sejam descritas para assegurar sua capacidade funcional. Exemplo: verificar se a entrada é aceitável; proteger contra consequências danosas decorrentes de erro de usuário; recuperar erro.

Declarações sobre Usabilidade

Deve ser especificado o tipo de interface com o usuário, como por exemplo, linha de comando, menu, janelas, teclas de função e função de auxílio. Deve ser descrito o conhecimento específico requerido para a aplicação do produto.

Declarações sobre Usabilidade

Se a proteção técnica contra infrações a direitos autorais pode dificultar a usabilidade, então essa proteção deve ser declarada. Exemplos: proteção técnica contra cópias,

datas programadas de expiração de uso, lembretes interativos para pagamento por cópia.

Declarações sobre Usabilidade

A descrição do produto deve incluir dados sobre a eficiência de uso e satisfação de usuário.

Declarações sobre Eficiência

Na descrição do produto podem ser incluídos dados sobre o comportamento do produto em relação ao tempo, tais como tempo de resposta para uma dada função sob condições estabelecidas. Por exemplo, a configuração do sistema.

Declarações sobre Manutenibilidade

A descrição do produto pode conter declarações sobre a manutenibilidade do produto.

Declarações sobre Portabilidade

A descrição do produto pode conter declarações sobre a portabilidade do produto.

- **Manual do usuário:** é um conjunto completo de documentos, disponível na forma impressa ou não, que é fornecido para a utilização de um produto, sendo também uma parte integrante do produto. Deve incluir todos os dados necessários para a instalação, para o uso da aplicação e para a manutenção do software produto. Deve-se observar o manual do usuário com relação à completitude, correção, consistência, inteligibilidade, apresentação e organização:
- **Completitude:** O manual deve conter todas as informações necessárias para o uso do produto, tais como estabelecer todas as funções do pacote e procedimentos de instalação.
- **Correção:** A informação apresentada no manual deve estar correta e sem ambiguidade.

- **Consistência:** Deve haver plena coerência entre a documentação no manual e a descrição do produto. Cada termo deve ter um único significado.
- **Inteligibilidade:** A documentação deve ser compreensível pela classe de usuários que desenvolve atividades com o produto, utilizando termos apropriados, exibições gráficas e explicações detalhadas.
- **Apresentação e Organização:** O manual deve ser apresentado de uma forma que facilite uma visão geral de índices e tabelas de conteúdo. Se o documento não está na forma impressa, deve haver indicação de como efetuar a impressão.

É importante ressaltar que as características de Funcionalidade, Confiabilidade e Usabilidade são destacadas e devem ser verificadas através do uso do produto. Não há requisitos específicos para os aspectos de Eficiência, Manutenibilidade e Portabilidade, porém se algum desses requisitos estiver declarado na documentação do pacote, eles devem estar em conformidade.

- **Programas e Dados:** são os requisitos de programas e dados que devem estar descritos, caso existam, para o funcionamento do produto. Os requisitos de qualidade para Programas e Dados utilizam as mesmas definições das características de qualidade da norma ISO/IEC 9126.
- **Funcionalidade:** Devem ser verificados os procedimentos para instalação do produto; a presença de todas as funções mencionadas; a execução correta dessas funções; a ausência de contradições entre a descrição do produto e a documentação do usuário.
- **Confiabilidade:** O usuário deve manter o controle do produto, sem corromper ou perder dados, mesmo que a capacidade declarada seja explorada até os limites ou fora deles, se uma entrada incorreta for efetuada, ou ainda se instruções explícitas na documentação forem violadas.

- **Usabilidade:** A comunicação entre o programa e o usuário deve ser de fácil entendimento, através das entradas de dados, mensagens e apresentação dos resultados, utilizando um vocabulário apropriado, representações gráficas e funções de auxílio (help), entre outras. O programa também deve proporcionar apresentação e organização que facilitem uma visão geral das informações, além de procedimentos operacionais que o auxiliem, por exemplo, a reversão de uma função executada.

A segunda parte da Norma, a parte que se refere às Instruções para teste. Como já comentado no início desse tópico, mas como a estrutura dessa norma é bastante extensa, vale a pena lembrar que a parte da Norma de Instruções para testes é dividida em: pré-requisitos de teste, atividades de teste, registro de teste e relatório de teste:

Os pré-requisitos de teste são:

- Presença de itens de produto,
- Presença do sistema necessário e,
- Treinamento (se mencionado na descrição do produto).

As atividades de teste consistem em testar se estão de acordo com os requisitos de qualidade:

- Descrição do produto
- Documentação do usuário
- Programas e dados

Os registros de teste devem conter informações suficientes para permitir a repetição do teste:

- plano de teste
- casos de teste

- registrar resultados (falhas/sucessos)
- identificar pessoas envolvidas

O Relatório de testes deve abordar:

- Produto
- Hw/Sw utilizado no teste
- Documentos usados
- Resultados dos testes (descrições, documentação, programas e dados)
- Lista de não conformidades dos requisitos
- Lista de não conformidade de recomendações
- Data do término do teste

REFLEXÃO:

O esforço para a elaboração conjunto de normas do Modelo ISO 9000 foi uma grande colaboração para o auxílio na obtenção da qualidade.

As normas vistas nessa aula, (ISO 9126 para avaliação de produto de software e a ISO 12119 para avaliação de pacote de software), são amplamente reconhecidas e, cada vez mais utilizadas pelas empresas

ATIVIDADE (EM GRUPO)

- **Elaborar Mapa Mental**
- **Responder as questões**

1. Comente algumas vantagens da realização da garantia de qualidade de software.
2. Explique as subcaracterísticas da característica funcionalidade da norma ISO 9126.
3. O que é um pacote de software? Dê alguns exemplos
4. Quais são os pré-requisitos de testes da norma ISO 12119?
5. O que deve ser testado de acordo com a norma ISO 12119?.

REFERÊNCIAS

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

ISO/IEC 9126. **International Standard**. Information Technology. Software Product Evaluation. Quality characteristics and guidelines for their use. Geneve, 1991.

ISO/IEC 12119. **International Standard. Information Technology** – Software Packages – Quality Requirements and testing. 1994.

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

PRESSMAN, Roger S. **Engenharia de Software**. Rio de Janeiro: MacGraw Hill, 2002.

SOMERVILLE, IAN. **Engenharia de Software**. 6ª edição. São Paulo: Addison Wesley, 2003