

Qualidade de Produto de Software

Engenharia de Software II
Sérgio Portari

Qualidade de Software

Qualidade de software é uma área dentro da Engenharia de Software que visa garantir bons produtos a partir de processos de boa qualidade.

Pode-se falar então de dois aspectos da qualidade:

- a qualidade do produto em si e
- a qualidade do processo.

Embora não exista uma garantia de que um bom processo vá produzir um bom produto, usualmente admite-se que a mesma equipe com um bom processo vá produzir produtos melhores do que se não tivesse processo algum.



Tipos de qualidade (Pressman)

Qualidade de software é um assunto amplo e de definição difusa.

Existem várias dimensões de qualidade e nem sempre é simples avaliar objetivamente cada uma das dimensões.

Pressman (2005) define dois tipos de qualidade para o produto de software:

- Qualidade de projeto, que avalia quão bem o produto foi projetado.

- Qualidade de conformação, que avalia quão bem o produto atende aos requisitos.



SWEBOK(*)

Em relação à qualidade, o SWEBOK faz uma distinção entre técnicas estáticas e dinâmicas.

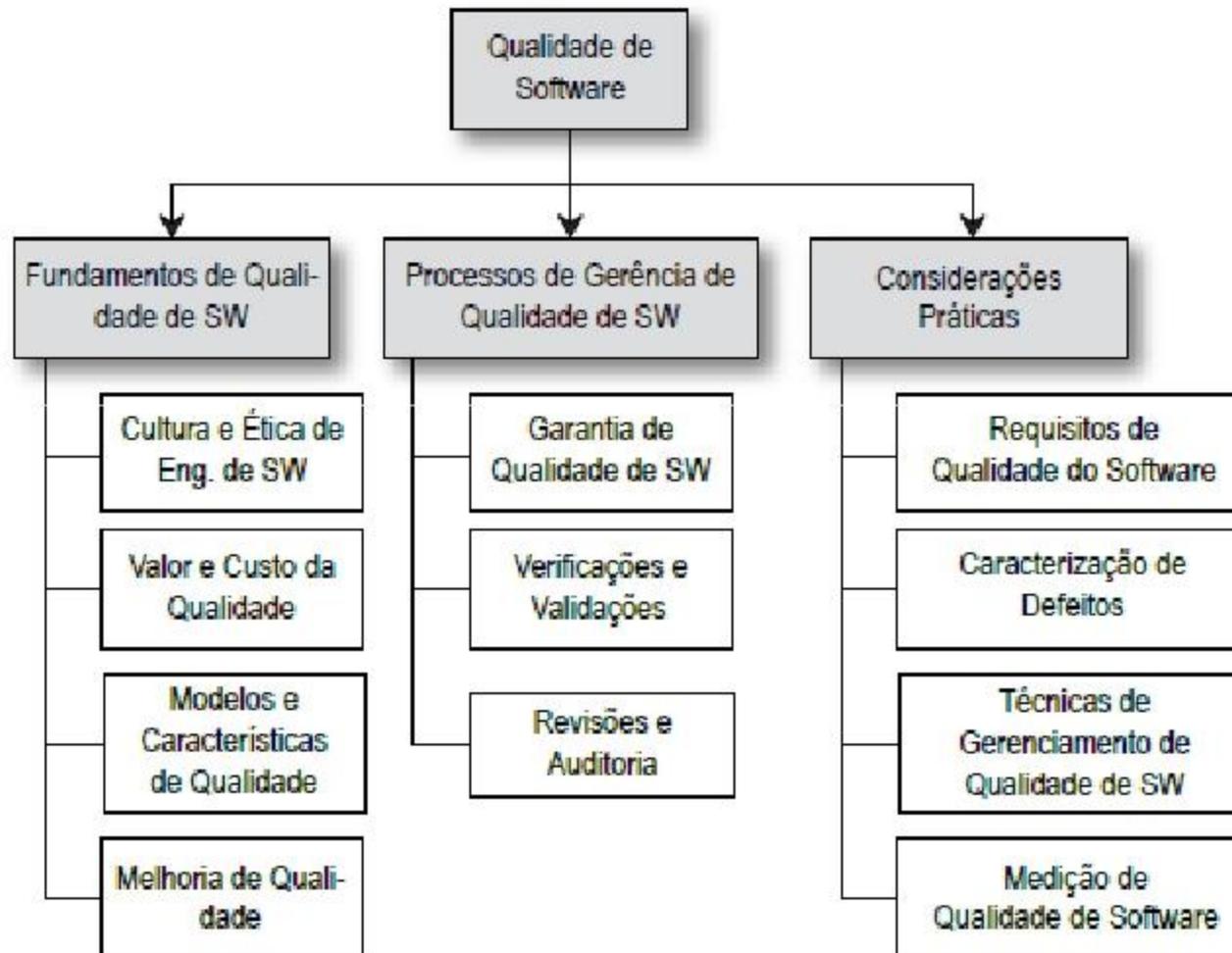
As técnicas estáticas são apresentadas nesta aula como “qualidade de software”

As técnicas dinâmicas são relacionadas ao teste do software.

* <http://www.computer.org/portal/web/swebok>



Subáreas da qualidade de software (SWEBOK)



Normas

A norma NBR 13596, originada a partir da ISO/IEC 9126, define qualidade de software como:

“a totalidade de características de um produto de software que lhe confere a capacidade de satisfazer necessidades explícitas e implícitas”.

A definição propositalmente não especifica os possuidores de tais necessidades, visto que ela se aplica a quaisquer atores envolvidos com a produção, encomenda, uso, ou pessoas afetadas pelas conseqüências do software ou de seu processo de produção.



Fatores externos e internos

As necessidades referidas podem também ser identificadas como necessidades percebidas, ou fatores externos de qualidade (como otimização e facilidade de uso), como fatores internos, normalmente só percebidos pela equipe de desenvolvedores (como organização arquitetural, facilidade de teste, facilidade de manutenção, etc.).



A norma 9128 definem o modelo de qualidade a partir de características e subcaracterísticas,mas não trata da avaliação da qualidade.

Essa avaliação é o assunto da série de normas ISO/IEC 14598 (Evaluation of software products).

Essas normas guiam a avaliação da qualidade em produtos de software em três contextos distintos:

- Desenvolvimento.

- Aquisição.

- Avaliação independente.



Tanto a norma 9126 quanto a 14598 tem como pressuposto que a qualidade do processo de desenvolvimento influencia a qualidade dos atributos internos do software, estes por sua vez influenciam a qualidade dos atributos externos, os quais, por sua vez influenciam a qualidade dos atributos de uso do software.



Indicadores de qualidade

O modelo de qualidade então avalia 4 tipos de indicadores de qualidade:

Medidas de qualidade do processo.

Medidas de qualidade internas.

Medidas de qualidade externas.

Medidas de qualidade do software em uso.



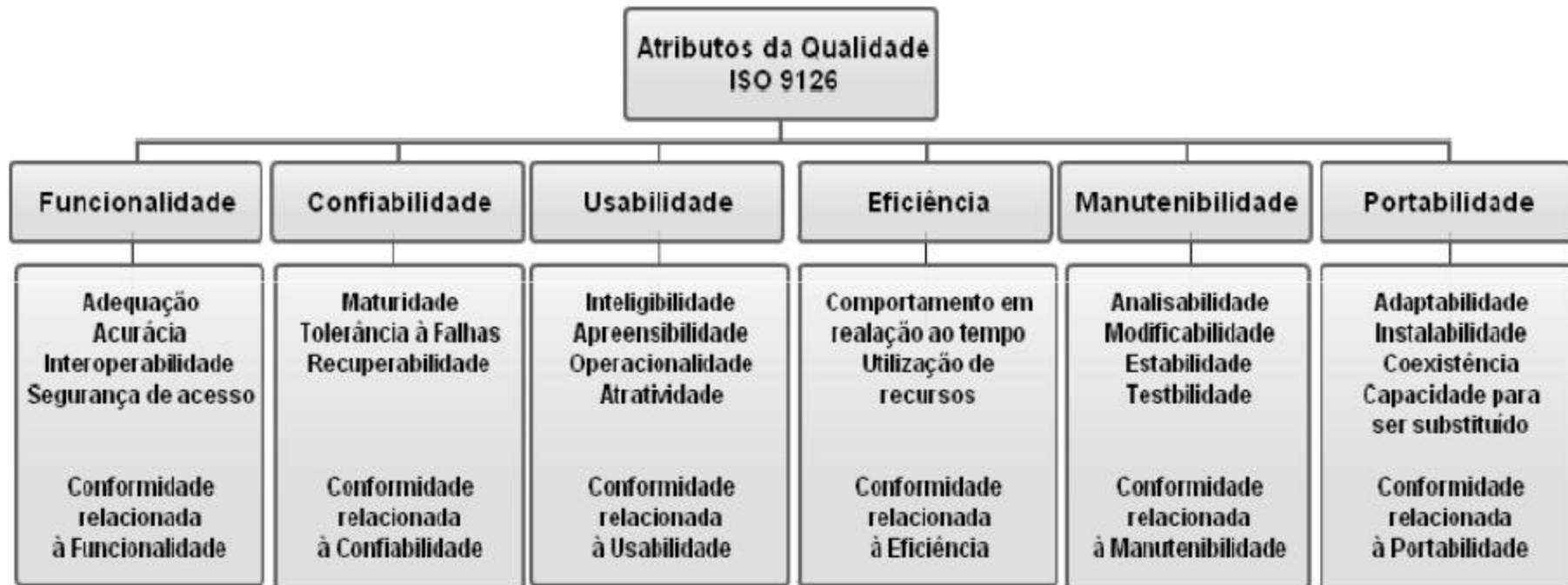
As qualidades internas, assim, permitem que a equipe de desenvolvimento atinja seus objetivos de forma eficiente.

As qualidades externas permitem que o usuário final do sistema atinja seus objetivos.

As qualidades internas então nem sempre são importantes para o usuário final, pois este não as percebe diretamente, mas pode ser afetado indiretamente por elas (no tempo de manutenção ou evolução do software, por exemplo).



ISO/IEC 9126



Funcionalidade

As características de funcionalidade avaliam o grau em que o software satisfaz as necessidades implícitas e explícitas de seus usuários finais.

São características externas e, portanto, percebidas por estes usuários.



Subcaracterísticas da Funcionalidade

Adequação:

O software efetivamente possibilita executar as funções que são apropriadas, ou seja, as entradas e saídas de dados necessárias são possíveis? Um software no qual faltem algumas funções necessárias não apresenta a qualidade de adequação.

Acurácia:

O software gera dados ou consultas corretos e precisos de acordo com sua definição? Um software que apresenta dados incorretos ou com grau de imprecisão acima de um limite definido como tolerável não apresenta a qualidade de acurácia.

Interoperabilidade:

O software é capaz de interagir com outros sistemas com os quais se espera que ele interaja? Um software incapaz de se comunicar adequadamente com outros sistemas chave não apresenta a qualidade de interoperabilidade.

Segurança de acesso:

embora o termo possa ter outros significados, a norma define segurança de acesso como a garantia de que dados ou funções não serão acessados proposital ou acidentalmente por pessoas ou sistemas que não tenham autorização para tal acesso.

Conformidade:

esta qualidade avalia o grau em que o software está de acordo com normas, templates, leis, descrições ou quaisquer outras restrições que limitem a maneira como ele deve funcionar ou se apresentar. A conformidade é uma qualidade relativa, pois depende de que tenham sido identificadas ou estabelecidas as normas às quais o software deve se adequar.



Confiabilidade

Um software confiável é aquele que ao longo do tempo se mantém com um comportamento dentro do esperado. A confiabilidade tem relação com a minimização da quantidade de falhas ou erros do software e com a forma como ele funciona perante situações anômalas.



Subcaracterísticas da confiabilidade

Maturidade:

A maturidade é a medida da frequência com que um software apresenta falhas. Um software mais maduro é aquele que apresenta menos falhas ao longo de um período fixo de tempo. Espera-se que a maturidade de um sistema aumente com o tempo, mas processos de manutenção mal gerenciados, especialmente aqueles que deixam de realizar testes de regressão ou refatoração, podem fazer com que a maturidade de um sistema diminua com o passar do tempo, ou seja, ao invés de reduzir a frequência dos erros eles podem passar a aumentar.

Tolerância a falhas:

A sub-característica de tolerância a falhas tem relação à forma como o software reage quando em situação anômala. Idealmente, requisitos de tolerância a falhas deveriam ser definidos durante o projeto do software. Um software que consegue continuar funcionando mesmo quando falhas ocorrem tem boa avaliação em relação a esta qualidade. Deve-se deixar claro, porém, que falhas e erros são coisas diferentes. A maturidade tem a ver com a minimização de erros que são indesejáveis em qualquer sistema. Uma falha, porém, é uma situação que pode ocorrer mesmo que o software não apresente nenhum erro. Por exemplo, uma linha de comunicação pode ser temporariamente interrompida, um dispositivo de armazenamento pode ser danificado, um processador pode apresentar defeito, etc. Estas falhas são imprevisíveis e na maioria das vezes inevitáveis. A qualidade então tem relação com a maneira como o software reage a estas situações externas indesejadas.

Recuperabilidade:

a recuperabilidade de um sistema está relacionada a sua capacidade de recuperar dados e colocar-se novamente em operação após uma situação de desastre. Idealmente deveria haver requisitos de recuperabilidade definidos para a maioria dos projetos de software, pois situações negativas como uma falha generalizada ou perda de dados nem sempre são previstas em um projeto a não ser que sejam explicitamente recomendadas. Em relação a tolerância a falhas, pode-se dizer que a recuperabilidade trata de situações mais críticas, onde o problema ocorrido não é apenas temporário, como uma falha de comunicação, mas mais definitivo, como a perda completa de um disco de dados.

Conformidade a normas de confiabilidade.



Usabilidade

A usabilidade de software por si só é uma área de conhecimento bastante desenvolvida, com muito material publicado, sejam normas técnicas, sejam estudos empíricos.

A usabilidade é uma característica externa do software, e portanto, visível para os usuários.

Talvez seja uma das características de qualidade mais claramente perceptíveis por usuários de software, visto que tem relação com a atividade de uso do software.



Subcaracterísticas da usabilidade

Inteligibilidade:

A inteligibilidade de um software é difícil de mensurar. Ela tem relação com o grau de facilidade que um usuário tem em entender os conceitos chave do software e assim tornar-se competente no seu uso. Essa característica tem também relação com a facilidade com que um potencial usuário compreende se o software é ou não adequado às tarefas que ele necessita efetuar e condições de uso específicas.

Apreensibilidade:

A apreensibilidade tem relação com o grau de treinamento necessário para que um novo usuário seja capacitado a usar o software.

Operacionalidade:

A operacionalidade avalia o grau em que é fácil controlar a operação do software.

Conformidade a normas de usabilidade.

As três primeiras sub-características são parecidas à primeira vista, mas um software pode ser fácil de aprender, mas difícil de operar, ou fácil de aprender e operar, mas difícil de perceber seu potencial. Assim, elas se complementam como características distintas e individualmente avaliáveis.



Eficiência

A eficiência trata da otimização do uso de recursos de tempo e espaço.

Espera-se que um sistema seja o mais eficiente possível de acordo com o tipo de problema que ele soluciona.

Existem problemas para os quais as soluções computacionais mais eficientes demandam quantidades de tempo e memória que as tornam intratáveis.

Nestes casos, a eficiência pode ser conseguida com sacrifício, por exemplo, da acurácia, como no caso de algoritmos que buscam soluções aproximadas para problemas intratáveis em tempo razoável.



Subcaracterísticas da eficiência

Eficiência de tempo:

essa qualidade mede o tempo que o software leva para processar suas funções. Existe para todos os problemas algoritmos um limite mínimo de complexidade que pode ser demonstrado formalmente. Um sistema eficiente em termos de tempo então é aquele cujo tempo de processamento se aproxima deste mínimo.

Eficiência de recursos:

normalmente associada a espaço de armazenamento ou memória, a eficiência de recursos também pode ser associada a outros recursos necessários como por exemplo, banda de transmissão de rede. Em algumas situações pode-se obter eficiência de tempo com sacrifício da eficiência de espaço e vice-versa.

Conformidade a normas de eficiência.



Manutenibilidade

A manutenibilidade é uma característica interna do software que só é diretamente percebida pelos desenvolvedores, embora os clientes possam ser afetados por ela na medida do tempo gasto pelos desenvolvedores para executar atividades de manutenção ou evolução do software.

A manutenibilidade então mede a facilidade de se realizar alterações no software para sua evolução, ou de detectar e corrigir erros.



Subcaracterísticas da manutenibilidade

Analisabilidade:

um sistema é analisável quando permite encontrar falhas facilmente quando elas ocorrem. Sistemas que quando falham travam o computador, por exemplo, podem ter um nível de analisabilidade baixo, porque é difícil encontrar uma falha em um sistema travado. Já sistemas que ao falharem apresentam mensagens relacionadas com exceções ocorridas são mais facilmente analisáveis.

Modificabilidade:

a modificabilidade tem relação com a facilidade que o sistema oferece para que erros sejam corrigidos quando detectados. Boas práticas de programação, arquitetura bem definida, refatoração quando necessário, aplicação de padrões de projeto e padrões de programação e testes automatizados são exemplos de disciplinas que podem colaborar para que sistemas tenham melhor modificabilidade.

Estabilidade:

a estabilidade tem relação com a probabilidade de novos erros ocorrerem quando modificações são feitas no software. Esta qualidade pode estar diretamente ligada à qualidade da modificabilidade, mas a anterior tem relação com a facilidade de realizar a modificação enquanto esta tem relação com potenciais problemas gerados pela modificação.

Testabilidade:

a testabilidade mede a facilidade de se realizar testes de regressão. Essa qualidade não diz tanto respeito ao software em si quanto ao processo estabelecido para permitir que o software seja testado.

Conformidade a normas de manutenibilidade.



Portabilidade

A portabilidade é uma característica que tem relação com a capacidade de instalação e execução do sistema em diferentes plataformas de hardware.

O grau de facilidade dessa adaptação pode ser associado então com essa qualidade.



Subcaracterísticas da portabilidade

Adaptabilidade:

avalia o quanto é fácil adaptar o software a outros ambientes sem a necessidade de aplicar ações ou meios além daqueles fornecidos com o próprio software.

Capacidade de instalação:

avalia a facilidade de se instalar o software.

Capacidade de substituição:

avalia a facilidade de substituir o software por outros sistemas se necessário.

Conformidade de portabilidade:

avalia se o software está de acordo com padrões ou convenções de portabilidade.



Normas ISSO/IEC 25000 – SQuaRE(*)

Desde 2005 a ISO/IEC 9126 foi sucedida pela família de normas ISO/IEC 25000:

Software engineering: Software product Quality Requirements and Evaluation (SQuaRE): Guide to SQuaRE.

* Disponível para aquisição em:

http://www.iso.org/iso/catalogue_detail.htm?csnumber=35683.



Modelo SQuaRE

O modelo SQuaRE (Suryan & Abran, 2003) é formado por um conjunto de normas dividido da seguinte forma:

ISO/IEC 2500n – Divisão Gestão da Qualidade.

ISO/IEC 2501n – Divisão Modelo de Qualidade.

ISO/IEC 2502n – Divisão Medição da Qualidade.

ISO/IEC 2503n – Divisão Requisitos de Qualidade.

ISO/IEC 2504n – Divisão Avaliação da Qualidade.



Este conjunto de normas, apesar de já contar com algumas publicações, ainda está em desenvolvimento.

AABNT também vem trabalhando na adaptação destas normas para um padrão brasileiro que venha a substituir a NBR 13596.

Uma das motivações para a criação de uma nova norma está no fato de que as antigas aplicavam-se apenas ao processo de desenvolvimento e uso do produto de software, mas pouco tinham a dizer em relação à definição do produto.



Para a definição de uma segunda geração de normas de qualidade, foi então utilizado um modelo genérico de processo de desenvolvimento baseado na norma ISO/IEC 15288 – System Life Cycle Processes:

Processo de Definição de Requisitos dos Interessados	Requisitos de Qualidade do Produto de Software
Processo de Análise dos Requisitos	Desenvolvimento do Produto
Processo de Projeto Arquitetural	
Processo de Implementação	
Processo de Integração	
Processo de Verificação	
Processo de Transição	
Processo de Validação	
Processo de Operação	Uso do Produto
Processo de Manutenção	
Processo de Aposentadoria	

O lado esquerdo da figura apresenta as fases originais da norma I5288 e o lado direito o agrupamento destas fases para efeito da aplicação das normas de qualidade SQuaRE.



Gestão da Qualidade

A divisão de gestão da qualidade do modelo SQuaRE apresenta os modelos comuns, padrões básicos termos e definições usados por toda a série de normas SQuaRE.

Esta divisão inclui duas unidades:

Guia do SQuaRE, que apresenta a estrutura, terminologia, visão geral do documento, público alvo, modelos de referência e partes associadas da série.

Planejamento e Gerenciamento, que apresenta os requisitos para planejar e gerenciar o processo de avaliação da qualidade de produtos de software.



Modelo de Qualidade

O modelo de qualidade apresenta as características e subcaracterísticas de qualidade interna, externa e de uso.



Medidas de Qualidade

Os padrões na área de medidas de qualidade derivas da ISO/IEC 9126 e 14598, cobrindo as definições matemáticas e detalhamento da aplicação de medidas práticas de qualidade interna, externa e de uso. O documento inclui:

Modelo de referência e guia de medição:

que apresenta uma introdução e explicação sobre a aplicação das medidas de qualidade.

Medidas Primitivas:

um conjunto de medições básicas usadas para a definição das demais.

Medidas Internas:

o conjunto de medidas quantitativas em termos de características e subcaracterísticas internas.

Medidas Externas:

o conjunto de medidas quantitativas em termos de características e subcaracterísticas externas.

Medidas de Uso:

o conjunto de medidas quantitativas em termos de características e subcaracterísticas de uso do software.



Requisitos de Qualidade

A divisão de requisitos de qualidade contém o padrão para suportar a especificação de requisitos de qualidade, tanto para a fase de eliciação dos requisitos de qualidade do software quanto como entrada para o processo de avaliação da qualidade do software.



Avaliação da Qualidade

A divisão de avaliação da qualidade provê as ferramentas para a avaliação da qualidade de um sistema de software tanto por desenvolvedores, compradores ou avaliadores independentes. Os seguintes documentos são disponibilizados:

Guia e visão geral da avaliação da qualidade:

que apresenta um framework para a avaliação da qualidade de um produto de software.

Processo para desenvolvedores:

recomendações práticas para a avaliação da qualidade de um produto quando esta é feita paralelamente ao seu desenvolvimento.

Processo para compradores:

recomendações práticas para a avaliação de produtos comprados em prateleira (COTS – commercial-off-the-shelf), feitos por encomenda, ou ainda para avaliação de modificações em sistemas existentes.

Processo para avaliadores:

recomendações práticas para a avaliação do software por terceiros enfatizando a participação de vários agentes que precisam compreender e aceitar essa avaliação.

Documentação para o módulo de avaliação:

define a estrutura e conteúdo da documentação usada no processo de avaliação.





Figura 3 — Processo de avaliação segundo NBR ISO/IEC 14598-1.

Tabela 1 — Tipos e técnicas de teste

Tipo/Técnica	Descrição
Teste Baseado na Especificação	Conjunto de técnicas pelas quais os casos de testes são derivados de modelos, formais ou informais, que especificam um problema a ser resolvido, o software ou seu componente. Testes podem ser especificados a partir de casos de uso ou cenários de negócios (ISTQB, 2007).
Análise de Valores Limites	Técnica de teste baseada em especificação, pela qual casos de testes são desenvolvidos com foco nos valores fronteiros para entradas e saídas de determinada função (QAI, 2006).
Indução de Erros	Técnica que visa determinar a capacidade da aplicação de processar transações incorretas. Consiste em testar a introdução de erros, seu processamento, controles e reteste após a correção (QAI, 2006).
Teste de <i>stress</i>	Tipo de teste o qual valia um sistema ou componente em relação e além dos limites de seus requisitos especificados (ISTQB, 2007).
Teste de Carga	Tipo de teste que mede o comportamento de um componente ou sistema por meio do aumento de carga, por exemplo, número de usuários paralelos e/ou número de transações para determinar qual o tamanho de carga que pode ser suportada pelo componente ou sistema (ISTQB, 2007).
Teste de Longa Duração	Também conhecidos como testes de longevidade, saturação, imersão ou resistência, são técnicas que procuram por problemas de performance ou segurança que só aparecem depois de um longo período de operação do software. Diferem dos testes de estresse típicos, que aplicam uma carga pesada ao sistema, na medida em que verificam se o <i>buffer</i> de dados e as filas de mensagens operam corretamente durante longos períodos (Woody, 2009).
Teste de Penetração	Método que avalia a segurança de um sistema ou de uma rede, simulando ataques maliciosos. O processo envolve uma análise das vulnerabilidades do sistema (QAI, 2006).

Tabela 2 — Especialização do Modelo de Qualidade para o estudo de caso

RRBT		ISO 9126	
Requisito	Risco	Característica	Subcaracterística
Envio e recepção de nova versão à base centralizada.	R001 - Indisponibilidade do sistema para o usuário.	Confiabilidade	Tolerância a Falhas
			Maturidade
Recuperabilidade			
Envio e recepção de informações dos sistemas relacionados.	R002 - Insuficiência dos recursos envolvidos com a produção do sistema, causando indisponibilidade.	Eficiência	Utilização de Recursos
	R003 - Interceptação de informações sigilosas no tráfego de rede utilizado pelo sistema.	Funcionalidade	Segurança de Acesso
R004 - Acesso liberado, aos usuários da aplicação, às informações que ficam inseridas no banco local instalado na estação de trabalho do usuário.			



Tabela 3 - Aplicação de GQM para o Risco 001

R001 - Confiabilidade / Maturidade		
Objetivo		
Propósito: Avaliar Questão: a capacidade de prevenção de falhas Objeto: do sistema Ponto de vista: conforme usuário		
Questão	Métrica	Tipo/Técnica de Teste
Quantas falhas foram detectadas durante um período definido de experimentação?	Número de falhas detectadas / Número de casos de testes	Teste de longa duração Testes de indução de falhas Teste baseados na especificação
R001 - Confiabilidade / Tolerância a falhas e Recuperabilidade		
Objetivo		
Propósito: Avaliar Questão: a disponibilidade Objeto: do sistema Ponto de vista: conforme usuário		
Questão	Métrica	Tipo/Técnica de Teste
Quanto padrões de falhas são mantidos sob controle para evitar falhas críticas e sérias?	Número de ocorrências de falhas sérias e críticas evitadas conforme os casos de testes de indução de falhas / Número de casos de testes de indução de falhas executados	Teste de longa duração Testes de indução de falhas
Quão disponível é o sistema para uso durante um período de tempo específico?	a) { Tempo de operação / (Tempo de operação + Tempo de reparo) } b) Total de casos em que o sistema estava disponível e foi utilizado com sucesso pelo usuário / Número total de casos em que o usuário tentou usar o software durante um período de tempo	Teste de longa duração Testes de indução de falhas
Qual é o tempo médio em que o sistema fica indisponível quando uma falha ocorre, antes da inicialização?	Tempo ocioso total (indisponível) / Número de quedas do sistema	Teste de longa duração Testes de indução de falhas
Qual o tempo médio que o sistema leva para completar a recuperação desde o início?	Soma de todos os tempos de recuperação do sistema inativo em cada oportunidade / Número total de casos em que o sistema entrou em recuperação	Teste de longa duração Testes de indução de falhas

Tabela 4 - Aplicação de GQM para o Risco 002

R002 – Eficiência /Utilização de Recursos		
Objetivo		
Propósito: Avaliar Questão: a eficiência na utilização de recursos Objeto: de produção Ponto de vista: conforme usuário		
Questão	Métrica	Tipo/Técnica de Teste
Qual é o limite absoluto de transmissões necessárias para cumprir uma função?	Número máximo de mensagens de erro e falhas relacionadas a transmissão do primeiro ao último item avaliado / Máximo requerido de mensagens de erro e falhas relacionadas a transmissão	Teste stress Valores limites de usuários simultâneos Valores limites de dados trafegados (simular carga máxima)
O sistema é capaz de desempenhar tarefas dentro da capacidade de transmissão esperada?	Capacidade de transmissão / Capacidade de transmissão específica projetada para ser usada pelo software durante sua execução	Teste stress Valores limites de usuários simultâneos Valores limites de dados trafegados (simular carga máxima)



Tabela 5 - Aplicação de GQM para o Risco 003

R003 – Funcionalidade / Segurança de Acesso		
Objetivo		
Propósito: Avaliar Questão: a integridade dos dados Objeto: do sistema Ponto de vista: conforme usuário		
Questão	Métrica	Tipo/Técnica de Teste
Qual é a frequência de eventos de corrupção de dados?	a) 1 - (Número de vezes que o maior evento de corrupção de dados ocorreu / Número de casos de testes executados que causaram eventos de corrupção de dados) b) 1- (Número de vezes que o menor evento de corrupção de dados ocorreu / Número de casos de testes executados que causaram eventos de corrupção de dados)	Testes de penetração Simulação de padrões de ataques às vulnerabilidades da conexão



Tabela 6 - Aplicação de GQM para o Risco 004

R004 – Funcionalidade / Segurança de Acesso		
Objetivo		
Propósito: Avaliar Questão: o controle de acesso Objeto: ao sistema Ponto de vista: conforme usuário		
Questão	Métrica	Tipo/Técnica de Teste
Quão completa é a trilha de auditoria sobre o acesso do usuário ao sistema e dados?	Número de acessos do usuário ao sistema e dados gravados no log de acesso / Número de acessos do usuário ao sistema e dados realizados durante a avaliação	Testes de penetração; Simulação de padrões de ataques às vulnerabilidades da conexão
Quão controlável é o acesso ao sistema?	Número (tipos diferentes) de operações ilegais detectadas/ Número (tipos diferentes) de operações ilegais especificadas	Testes de penetração Simulação de padrões de ataques às vulnerabilidades da conexão



Bibliografia

Pressman, R.S. (2005). Software Engineering: A Practitioner's Approach (6th edition ed.). McGraw-Hill Education.

Santos, L.B., & Pretz, E. (2009). Framework para Especialização de Modelos de Qualidade de Produtos de Software. monografia, Centro Universitário Feevale.

Suryn, W., & Abran, A. (2003). ISO/IEC SQuaRE. The second generation of standards for software product quality. Proceedings of 7th IASTED International Conference on Software Engineering and Applications, SEA 2003. Marina del Rey, CA, USA.

