

# Qualidade de Produto de Software

Prof. Raul Sidnei Wazlawick



# Qualidade de software

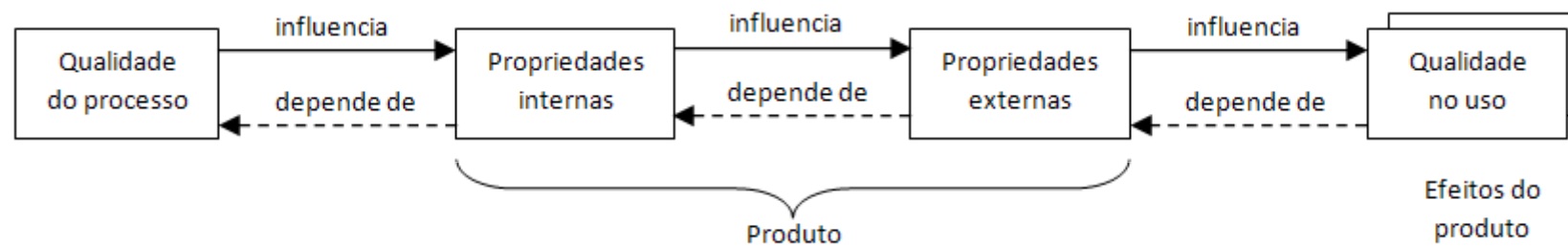
- *Qualidade de software* é uma área dentro da engenharia de software que visa garantir bons produtos a partir de bons processos.
- Pode-se falar então de dois aspectos da qualidade:
  - *a qualidade do produto em si e*
  - *a qualidade do processo.*
- Embora não exista uma garantia de que um bom processo 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.



# Modelo de Qualidade SquaRE – ISO/IEC 25010:2011

- Quatro indicadores de qualidade:
  - *Medidas de qualidade do processo.*
  - *Medidas de qualidade internas.*
  - *Medidas de qualidade externas.*
  - *Medidas de qualidade do software em uso.*

# Abordagem conceitual para qualidade





# Modelo Square

- 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.



## *Gestão da qualidade*

- 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*

- Apresenta as características e subcaracterísticas de qualidade interna, externa e de uso.
- Os padrões na área de medidas de qualidade são derivados das normas 9126 e 14598, cobrindo as definições matemáticas e o 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 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*

- Provê as ferramentas para a avaliação da qualidade de um sistema de software tanto por desenvolvedores, compradores ou avaliadores independentes.
- Inclui processos de avaliação para desenvolvedores, compradores e avaliadores.

Tabela 11-2: Modelo de qualidade da ISO 25010:2011.

Tipo	Características	Subcaracterísticas	
Características do produto	Adequação funcional (functional suitability)	Compleitude funcional Corretude funcional (acurácia) Funcionalidade apropriada	Functional completeness Functional correctness (accuracy) Functional appropriateness
	Confiabilidade (reliability)	Maturidade Disponibilidade Tolerância a falhas Recuperabilidade	Maturity Availability Fault tolerance Recoverability
	Usabilidade (usability)	Apropriação reconhecível Inteligibilidade Operabilidade Proteção contra erro de usuário Estética de interface com usuário Acessibilidade	Appropriateness recognisability Learnability Operability User error protection User interface aesthetics Accessibility
	Eficiência de desempenho (performance efficiency)	Comportamento em relação ao tempo Utilização de recursos Capacidade	Time behaviour Resource utilization Capacity
	Segurança (security)	Confidencialidade Integridade Não-repúdio Rastreabilidade de uso Autenticidade	Confidentiality Integrity Non-repudiation Accountability Authenticity
	Compatibilidade (compatibility)	Coexistência Interoperabilidade	Co-existence Interoperability
	Manutenibilidade (maintainability)	Modularidade Reusabilidade Analísabilidade Modificabilidade Testabilidade	Modularity Reusability Analysability Modifiability Testability
	Portabilidade (portability)	Adaptabilidade Instalabilidade Substituibilidade	Adaptability Instalability Replaceability
Características do uso	Efetividade (effectiveness)	Efetividade	Effectiveness
	Eficiência (efficiency)	Eficiência	Efficiency
	Satisfação (satisfaction)	Utilidade Prazer Conforto Confiança	Usefulness Pleasure Comfort Trust
	Uso sem riscos (freedom from risk)	Mitigação de risco econômico Mitigação de risco a saúde e segurança Mitigação de risco ambiental	Economic risk mitigation Health and safety risk mitigation Environmental risk mitigation
	Cobertura de contexto (context coverage)	Compleitude de contexto Flexibilidade	Context completeness Flexibility

# Adequação funcional

- Mede o grau em que o produto disponibiliza funções que satisfazem necessidades estabelecidas e implicadas quando o produto é usado sob condições especificadas. Suas subcaracterísticas são:
  - *Compleitude funcional.*
    - O software efetivamente possibilita executar as funções que são apropriadas, ou seja, as entradas e saídas de dados necessárias para o usuário atingir seus objetivos são possíveis?
  - *Corretude funcional.*
    - Também denominada *acurácia*, essa subcaracterística avalia o quanto o software gera dados e consultas corretos e precisos de acordo com sua definição.
  - *Funcionalidade apropriada.*
    - Esta subcaracterística indica qual o grau em que as funções do sistema facilitam a realização de tarefas e objetivos para os quais o sistema foi especificado.



# Confiabilidade

- O software, ao longo do tempo, se mantém com um comportamento consistente com o esperado.
  - *Maturidade.*
    - Medida da frequência com que um software apresenta defeitos.
  - *Disponibilidade.*
    - O quanto o software está operacional e disponível para uso quando se tornar necessário.
  - *Tolerância a falhas.*
    - A forma como o software reage quando em situação anômala.
  - *Recuperabilidade.*
    - Capacidade de recuperar dados e colocar-se novamente em operação após uma situação de desastre.



# Usabilidade

- Avalia o grau no qual o produto tem atributos que permitem que seja entendido, aprendido, usado e que seja atraente ao usuário, quando usado sob condições especificadas.
  - *Apropriação reconhecível.*
    - Mede o grau em que os usuários reconhecem que o produto é apropriado para suas necessidades.
  - *Inteligibilidade.*
    - 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.
  - *Operabilidade.*
    - Avalia o grau no qual o produto é fácil de usar e controlar.
  - *Proteção contra erro de usuário.*
    - Avalia o grau em que o produto foi projetado para evitar que o usuário possa cometer erros.
  - *Estética de interface com usuário.*
    - Avalia o grau em que a interface com o usuário proporciona prazer e uma interação satisfatória.
  - *Acessibilidade.*
    - Avalia o grau em que o produto foi projetado para atender a usuários com necessidades especiais.



# *Eficiência de desempenho*

- Trata da otimização do uso de recursos de tempo e espaço.
  - *Comportamento em relação ao tempo.*
    - Mede o tempo que o software leva para processar suas funções.
  - *Utilização 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.
  - *Capacidade.*
    - Avalia o grau em que os limites máximos do produto atendem aos requisitos.

# Segurança

**Segurança** (*security*) tem relação com a segurança dos dados e funções.

**Uso sem riscos** (*safety*), é uma qualidade do software em uso, relacionada com a segurança das pessoas, instalações e meio ambiente.

- Avalia o grau em que as funções e dados são protegidos de acesso não autorizado e o grau em que são disponibilizados para acesso autorizado.
  - *Confidencialidade.*
    - Avalia o grau em que as informações e funções do sistema estejam acessíveis por quem tenha a devida autorização para isso.
  - *Integridade.*
    - Avalia o grau em que os dados e funções do sistema são protegidos contra acesso por pessoas ou sistemas não autorizados.
  - *Não-repúdio.*
    - Avalia o grau em que o sistema permite constatar que ações ou acessos foram efetivamente feitos, de forma que não possam ser posteriormente negados.
  - *Rastreabilidade de uso.*
    - Avalia o grau em que as ações realizadas por uma pessoa ou sistema podem ser rastreadas de forma a comprovar que foram efetivamente realizadas por esta pessoa ou sistema.
  - *Autenticidade.*
    - Avalia o grau em que a identidade de uma pessoa ou recurso seja efetivamente aquela que se diz ser.



# *Compatibilidade*

- A avalia o grau em que dois ou mais sistemas ou componentes podem trocar informação e/ou realizar suas funções requeridas enquanto compartilham o mesmo ambiente de hardware e software.
  - *Coexistência.*
    - Avalia o grau no qual o produto pode desempenhar as funções requeridas eficientemente enquanto compartilha ambiente e recursos comuns com outros produtos, sem impacto negativo nos outros produtos.
  - *Interoperabilidade.*
    - Avalia o grau no qual o software é capaz de interagir com outros sistemas com os quais se espera que ele interaja.



# Manutenibilidade

- Mede a facilidade de se realizar alterações no software para sua evolução, ou de detectar e corrigir erros.
  - *Modularidade.*
    - Avalia o grau em que o sistema é subdividido em partes lógicas coesas, de forma que mudanças em uma dessas partes tenha impacto mínimo nas outras.
  - *Reusabilidade.*
    - Avalia o grau em que partes do sistema podem potencialmente ser usadas para construir outros sistemas.
  - *Analísabilidade.*
    - Um sistema é analisável quando permite encontrar defeitos (depurar) facilmente quando erros ou falhas ocorrem.
  - *Modificabilidade:*
    - Tem relação com a facilidade que o sistema oferece para que erros sejam corrigidos quando detectados, sem que as modificações introduzam novos defeitos, ou degradando sua organização interna.
  - *Testabilidade.*
    - Mede a facilidade de se realizar testes de regressão.

# Portabilidade

- Avalia o grau em que o software pode ser efetivamente e eficientemente transferido de um ambiente de hardware ou software para outro.
  - *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.
  - *Instalabilidade.*
    - Avalia a facilidade de se instalar o software.
  - *Substituibilidade.*
    - Avalia o grau em que o sistema pode substituir outro no mesmo ambiente e com os mesmos objetivos.



# Qualidades do Software em Uso

- As características de qualidade do software em uso são fatores externos que só podem ser plenamente avaliados quando o software está efetivamente em seu ambiente de uso final, ou seja, é muito difícil avaliá-las em ambiente de desenvolvimento.



## *Efetividade*

- É a capacidade que o produto de software tem para fazer com que o cliente atinja seus objetivos de negócio de forma correta e completa, no ambiente real de uso.



## *Eficiência*

- Avalia o retorno que o produto dá ao cliente, ou seja, a razão entre o que o cliente investiu e investe no sistema em relação ao que recebe em troca.
- Essa medida, nem sempre é financeira.



# Satisfação

- É a capacidade de o produto satisfazer aos usuários durante seu uso no ambiente final.
  - *Utilidade.*
    - Avalia o grau no qual o usuário é satisfeito com a obtenção percebida de metas pragmáticas, incluindo os resultados e as consequências do uso do software.
  - *Prazer.*
    - Avalia o grau em que o usuário sente prazer em usar o sistema para satisfazer seus objetivos.
  - *Conforto.*
    - Avalia o conforto físico e mental do usuário ao usar o sistema.
  - *Confiança.*
    - Avalia o grau em que o usuário ou outros interessados confiam que o sistema faça o que é esperado dele.

## *Uso sem riscos*

- É a capacidade de o produto estar dentro de níveis aceitáveis de segurança relativamente a riscos envolvendo pessoas, negócios e meio ambiente.
  - *Mitigação de risco econômico.*
    - Avalia o grau no qual o produto minimiza riscos financeiros potenciais, incluindo danos à propriedade e reputação de pessoas.
  - *Mitigação de risco a saúde e segurança.*
    - Avalia o grau no qual o produto minimiza riscos físicos às pessoas em seu contexto de uso.
  - *Mitigação de risco ambiental.*
    - Avalia o grau no qual o produto minimiza riscos ambientais ou à propriedade em seu contexto de uso.



## *Cobertura de contexto*


- Avalia o grau no qual o produto ou sistema pode ser usado com efetividade, eficiência, sem riscos e com satisfação tanto no contexto inicialmente especificado quanto em contextos além daquele.
  - *Completude de contexto.*
    - Avalia o grau no qual o produto ou sistema pode ser usado com efetividade, eficiência, sem riscos e com satisfação em todos os contextos especificados de uso.
  - *Flexibilidade.*
    - Avalia o grau no qual o produto ou sistema pode ser usado com efetividade, eficiência, sem riscos e com satisfação em contextos diferentes daqueles inicialmente especificados.





# Instalação de um Programa de Melhoria de Qualidade

- A instalação inicial do programa implica em estabelecer uma anistia geral na empresa, ou seja, não se busca culpados para o que aconteceu até o momento. Busca-se promover uma melhoria geral conjunta.
- Para que o programa de melhoria de qualidade funcione, é necessário que ele seja, primeiro, acordado e conhecido por todos os envolvidos e, segundo, que se torne parte da cultura da empresa.
  - Planos apenas colocados no papel que são abandonados frente à primeira dificuldade, logo são esquecidos.

- 
- Os planos devem ser consistentes, factíveis, gradualmente implementados e, principalmente, todos devem levar os planos a sério.
  - Note-se que planos inconsistentes, impossíveis de executar e que caem do céu prontos e acabados dificilmente serão levados a sério.



# Princípios para o sucesso de um programa de qualidade

- Melhorar constantemente o sistema de produção e serviços de forma maximizar o binômio qualidade/produtividade.
- Institucionalizar os novos métodos de treinamento no trabalhos.
- Institucionalizar e fortalecer os papéis de liderança.
- Eliminar os medos.
- Quebrar as barreiras entre departamentos.
- Eliminar slogans, exortações e metas de produtividade, pois isto pode levar a queda na qualidade.
- Eliminar cotas padrão arbitrarias e gerenciamento por objetivos.
- Institucionalizar um vigoroso programa de educação e automelhoria.
- Colocar todos para trabalhar pela modificação em prol da qualidade.



# Gestão da Qualidade

- Pode ser considerada uma atividade de gerenciamento que pode ser efetuada pelo gerente de projeto, mas preferencialmente deveria ser realizada por um gerente ou equipe especializados.
- Ela consiste no planejamento e execução das ações necessárias para que o produto satisfaça os requisitos de qualidade estabelecidos.



# Modelo de Maturidade de Crosby em relação à Qualidade

- *Desconhecimento.*
  - Quando a empresa não sabe sequer que tem problemas com qualidade.
- *Despertar.*
  - A empresa reconhece que tem problemas com a qualidade e que precisa começar a lidar com eles, mas ainda vê isso como um mal necessário, não como fonte de lucro para a empresa.
- *Alinhamento.*
  - O gerenciamento da qualidade se torna uma ferramenta institucional e os problemas vão sendo priorizados e resolvidos à medida que surgem.
- *Sabedoria.*
  - A prevenção de problemas, e não apenas sua correção, torna-se rotina na empresa.
  - Problemas são identificados antes que surjam e todos os processos e rotinas estão abertos a mudança visando a melhoria da qualidade.
- *Certeza.*
  - A gestão da qualidade é uma constante e uma parte essencial do funcionamento da empresa.
  - Quase todos os problemas são prevenidos e eliminados antes de surgirem.



# Técnicas para controle da qualidade

- *walkthrough*
- Inspeções Fagan




# Walkthrough

- É uma forma de avaliação do produto que utiliza uma equipe de especialistas, onde cada um faz uma análise prévia do produto, e depois reúnem-se (3 a 5 pessoas) por um período de cerca de duas horas, para trocar suas impressões sobre o produto e sugerir melhorias.
- Além dos analistas, a reunião de *walkthrough* deve contar preferencialmente com desenvolvedores e usuários, que poderão apresentar rapidamente respostas a eventuais dúvidas dos analistas, como por exemplo, “este requisito devia ter sido implementado desta forma mesmo?”.
- Ao término da reunião os participantes votam pela *aceitação do produto*, *aceitação com modificações parciais* ou *rejeição*. Sempre que houver modificações recomendadas o produto deverá passar por um novo *walkthrough*.

# Papeis em uma reunião *walkthrough*

- *Apresentador.*
  - Geralmente é o autor do artefato que o descreve, bem como as razões para ele ser desta forma.
  - Antes da reunião, ele entrega as especificações do artefato ao coordenador, que as distribui à equipe com antecedência.
- *Coordenador.*
  - É o moderador da reunião.
  - Seu trabalho é manter todos focados nas tarefas e não se envolver em discussões.
  - O ideal é que esse papel seja executado por alguém de fora da equipe.
- *Secretário.*
  - É o responsável por tomar nota das discussões e decisões.
  - Possivelmente suas notas deverão ser, ao final, aprovadas pelos participantes.
- *Oráculo de manutenção.*
  - É o inspetor de garantia de qualidade, cujo trabalho é certificar-se que o código produzido seja compreensível e manutenível, de acordo com os padrões da empresa.
- *Guardião dos padrões.*
  - Seu trabalho é certificar-se que o código produzido esteja de acordo com os padrões de programação estabelecidos previamente pela equipe. Se não houver padrões estabelecidos, possivelmente muito tempo da reunião será perdido com a discussão de irrelevâncias.
- *Representante do usuário.*
  - Ele pode estar presente em algumas reuniões, especialmente aquelas que discutem requisitos, para garantir que o cliente realmente receba o produto que ele espera.
- Outros desenvolvedores poderão participar também para dar sua visão e contribuição à discussão sob outros pontos de vista.



- 
- A reunião deve seguir estritamente o planejamento inicial, mantendo-se a discussão produtiva e objetiva, e que o objetivo principal é avaliar os defeitos, e não os desenvolvedores.
  - O objetivo da reunião não consiste em corrigir defeitos, apenas encontrá-los.
    - O processo de reparação vai ocorrer depois.
  - Erros triviais, como erros ortográficos em janelas, não necessitam de discussão. Apenas os erros mais graves.


# Perfis psicológicos em uma reunião walkthrough

- *Programadores gênios.*
  - Especialmente se for aquele tipo de gênio arrogante, impaciente e de mente estreita, ele pode causar problemas.
  - Devem ser valorizados, pois são capazes de detectar defeitos com facilidade (alimentam seu ego com isso).
  - O coordenador da sessão deve ter humildade e controle para não iniciar discussões com eles, nem deixar que outros o façam, pois tornará o trabalho improdutivo.
- *Pessoas defensivas e inseguras.*
  - Deve-se ter cuidado com esses, pois poderão se sentir atingidos pessoalmente pelas críticas feitas ao seu código.
  - É preciso tomar muito cuidado para que o trabalho seja mantido na discussão do produto, e não dos programadores.
  - A reunião de *walkthrough* não é o momento para tentar resolver a vida deles.
- *Conservadores.*
  - Também poderão causar problemas algumas vezes, pois buscam se manter fieis às tradições estabelecidas.
  - Deve-se dar atenção às suas opiniões porque a área de programação é muito sujeita a modismos.
  - Mas deve-se também procurar evitar que discussões improdutivas sejam iniciadas por eles.
- *Alienados.*
  - Estes não estão interessados no mundo real.
  - Eles primam mais pelo processo do que pelo produto, e podem ser uma incomodação séria.
  - O coordenador deve ter em mente que o processo só é útil quando ajuda a produzir o produto da melhor forma possível.
  - O processo não é uma religião que se deva seguir cegamente.
  - As regras existem porque tem objetivos a alcançar, e não porque foram ditadas por alguma divindade da computação.
  - Mas os alienados muitas vezes não percebem isso.



# Inspeções Fagan

- Consistem em um processo estruturado para tentar encontrar defeitos no código, diagramas ou especificações.
- Uma inspeção Fagan parte do princípio de que toda atividade que tenha critérios de entrada e saída bem definidos, pode ser avaliada de forma a verificar se ela efetivamente produz a saída especificada.
- Como as atividades de processos de software devem ser sempre definidas em termos de artefatos de entrada e saída, elas se prestam bem a serem avaliadas por inspeções Fagan.

- 
- Então os artefatos de entrada e saída equivalem aos critérios de entrada e saída para as inspeções e qualquer desvio encontrado nos artefatos de saída é considerado um *defeito*.
  - Defeitos podem ser classificados em diferentes tipos, como, por exemplo, defeitos graves e triviais.
    - Um *defeito grave* é caracterizado por um não funcionamento do produto, como por exemplo, uma função faltando.
    - Um *defeito trivial* é uma característica errada que não afeta a capacidade de funcionamento do software, como, por exemplo, um erro ortográfico em uma janela de sistema.



# Papeis em uma inspeção Fagan

- *Autor.*
  - O programador, *designer* ou analista, ou seja, a pessoa que produziu o artefato.
- *Narrador.*
  - Ele analisa, interpreta e resume o artefato e seus critérios de aceitação.
- *Revisores.*
  - Eles revisam o artefato com o objetivo de detectar eventuais defeitos.
- *Moderador.*
  - É o responsável pela sessão de inspeção e pelo andamento do processo.



# Atividades de uma inspeção Fagan

- *Planejamento:*
  - inclui a preparação dos materiais (artefatos), convite aos participantes e alocação do espaço de trabalho.
- *Visão geral:*
  - inclui a instrução prévia (apresentação) aos participantes sobre os materiais a serem inspecionados e a atribuição de papéis aos participantes.
- *Preparação:*
  - onde os participantes analisam os artefatos sob inspeção e material de suporte de forma a anotarem possíveis defeitos e questões para a reunião de inspeção.
- *Reunião de inspeção:*
  - é quando efetivamente se discutem e decidem quais os defeitos encontrados.
- *Retrabalho:*
  - é a atividade sob responsabilidade do autor do artefato na qual ele corrige os defeitos apontados na reunião de inspeção.
- *Prosseguimento (follow-up):*
  - a atividade de prosseguimento considera que todos os defeitos foram corrigidos e o produto está aprovado para prosseguir para a fase seguinte ou entrega.



# Requisitos de Qualidade

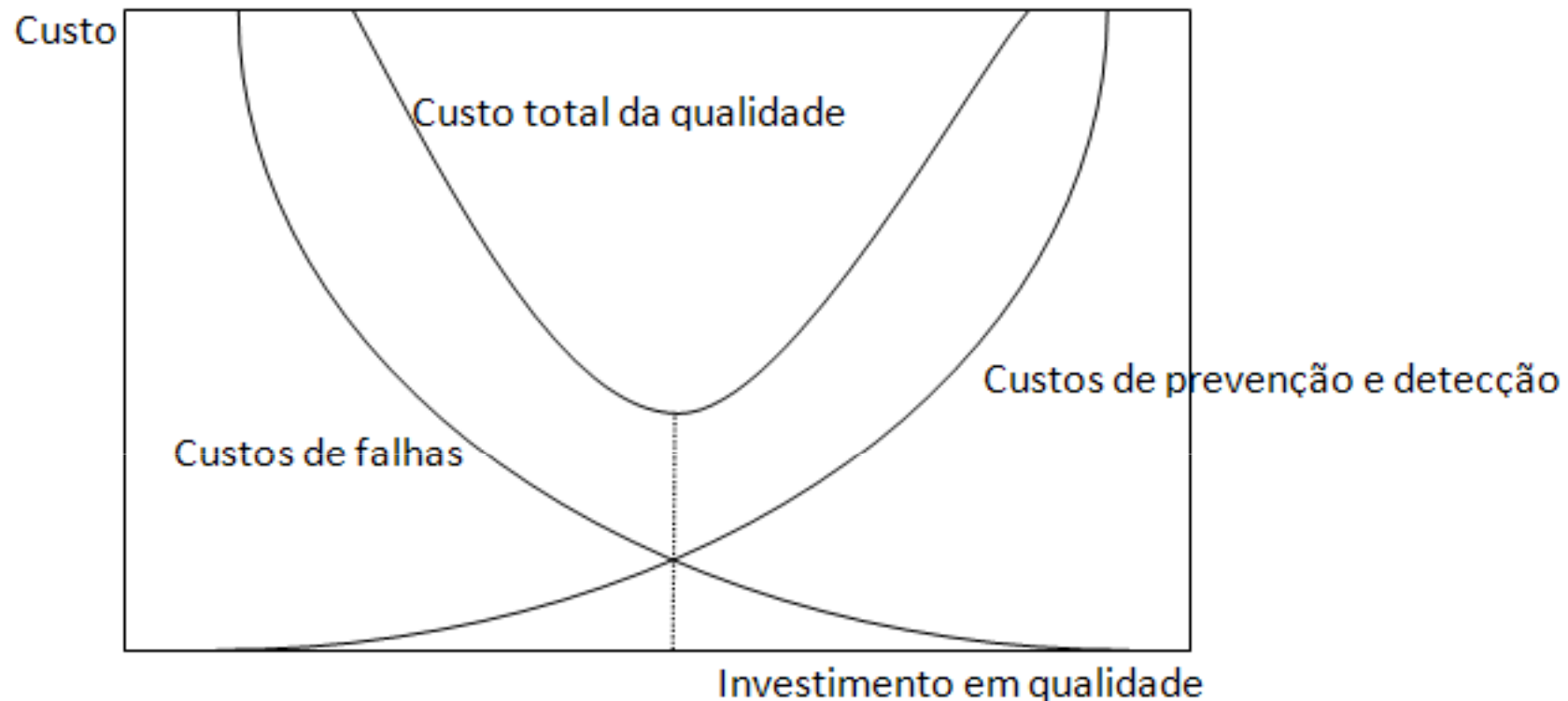
- Podem ser catalogados, mas cada produto terá um conjunto de requisitos diferente, pois qualidade também tem custo.
- Algumas subcaracterísticas de qualidade são sempre desejáveis, e possivelmente podem ser obtidas a partir de um bom processo de desenvolvimento, como por exemplo, o software ser livre de defeitos.
- Mas outras qualidades (como portabilidade) poderão ser eletivas e o custo de sua inclusão no software poderá não ser justificável.
- Qualidade não é sinônimo de perfeição, mas algo factível, relativo, dinâmico e evolutivo que se amolda aos objetivos a serem atingidos.



# Moscow

- Como os requisitos de qualidade são suplementares ou não funcionais, é de se esperar que possam ser classificados em diferentes graus de obrigatoriedade.
- Pode-se usar aqui o padrão *MOSCOW* (*Must*, *Should*, *Could* e *Would*), para determinar o grau de necessidade que um determinado requisito de qualidade seja cumprido.



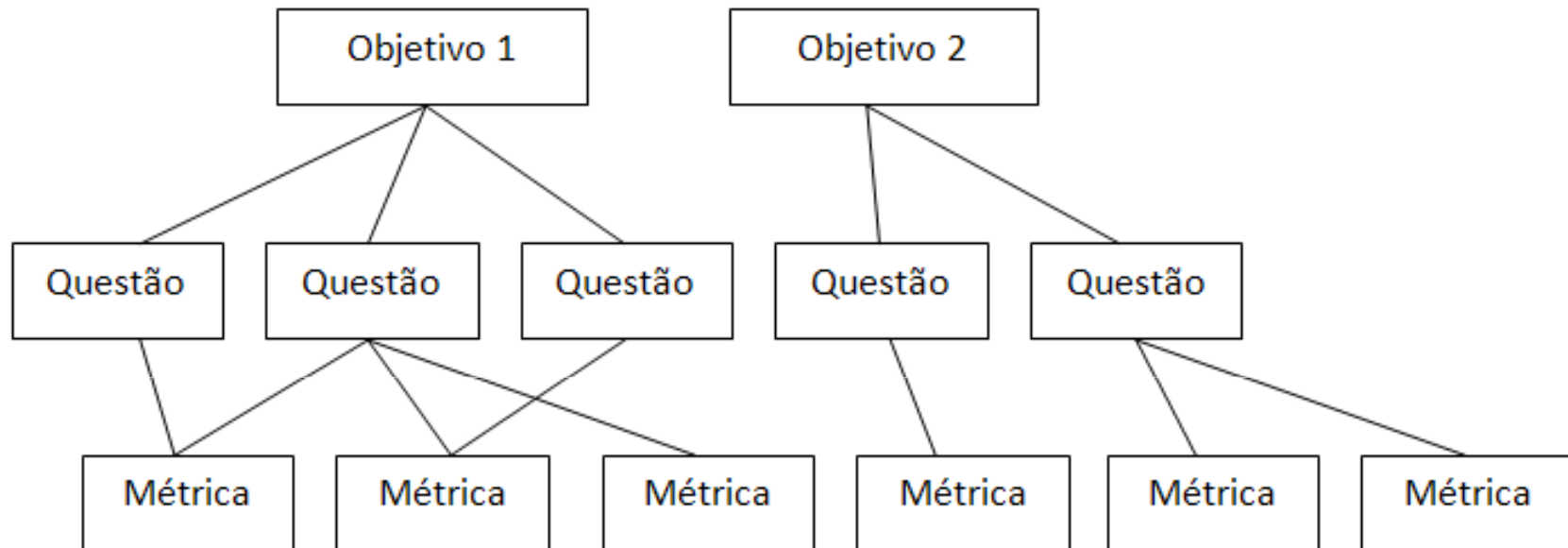


- Existe um ponto ótimo para o investimento em qualidade que baixa os custos com falhas o suficiente para compensar o investimento.





# GQM (*Goal/Question/Metric*) e Avaliação da Qualidade

- *Nível conceitual (Goal/objetivo):*
  - um objetivo é definido para um objeto por uma variedade de razões, com respeito a vários modelos de qualidade, a partir de vários pontos de vista e relativamente a um ambiente em particular (os *objetos* a serem medidos podem ser: produtos, processos ou recursos).
- *Nível operacional (Question/questão):*
  - um conjunto de questões é usado para definir modelos de objetos de estudo e então focar no objeto que caracteriza a avaliação ou a obtenção de um objetivo específico.
- *Nível quantitativo (Metric/métrica):*
  - um conjunto de dados, baseados nos modelos, é associado com cada questão para respondê-la de forma quantitativa (os dados podem ser *objetivos*, se dependem apenas do objeto avaliado, ou *subjetivos*, se dependem de uma interpretação do avaliador).




- Em GQM a definição do processo de avaliação é feita de forma *top-down*, ou seja, dos objetivos até as métricas, enquanto que a interpretação dos resultados é feita de forma *bottom-up*, ou seja, das métricas até os objetivos.

- 
- Santos e Pretz (2009) apresentam um estudo de caso onde GQM é usado para avaliar um projeto de desenvolvimento de software.
  - Cada risco importante do sistema é analisado como um objetivo, para o qual são definidas questões e métricas.

- 
- Inicialmente os autores associam os riscos identificados às características e subcaracterísticas de qualidade da norma 9126, na época ainda em vigência.


Sistema Exemplo		NBR ISO/IEC 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.		
	R004 – Acesso liberado, aos usuários da aplicação, às informações que ficam inseridas no banco local instalado na estação de trabalho do usuário.		
		Funcionalidade	Segurança de acesso

Figura 11-5: Associação de riscos às características e subcaracterísticas de qualidade<sup>159</sup>.

- 
- Assim, para cada risco identificado e possivelmente para cada subcaracterística de qualidade associada ao risco, um **objetivo** é estabelecido.
  - Para cada objetivo, uma ou mais **questões** são colocadas e, para cada questão, uma ou mais **métricas** são definidas.

Risco R001 - Indisponibilidade do sistema para o usuário				
Característica	Subcaracterística	Objetivo	Questão	Métrica
Confiabilidade	Maturidade	Avaliar a capacidade de prevenção de falhas do sistema do ponto de vista do usuário	Quantas falhas foram detectadas durante um período definido de experimentação?	Número de falhas detectadas / número de casos de testes
	Tolerância a falhas e Recuperabilidade	Avaliar a disponibilidade do sistema do ponto de vista do usuário	Quanto padrões de defeitos 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
			Quão disponível é o sistema para uso durante um período de tempo específico?	Tempo de operação / (Tempo de operação + Tempo de reparo)  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
			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
			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



- 
- Como se pode ver na figura, o *objetivo* é especificado de acordo com um padrão estabelecido pelo próprio GQM, que sugere que objetivos sejam estabelecidos a partir de diferentes dimensões:
    - *Propósito*. Um verbo que representa o objetivo, como, por exemplo, “avaliar”.
    - *Questão*. Um adjetivo referente ao objeto, como, por exemplo, “a maturidade de”.
    - *Objeto*. O objeto em avaliação, como, por exemplo, “o software”.
    - *Ponto de vista*. Para quem a avaliação é feita, como, por exemplo, “do ponto de vista do cliente”.