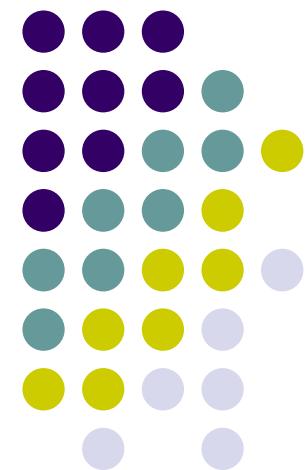


# ISO/IEC 9126

---

Prof. Alexandre Luís Franco





# ISO/IEC 9126

- Contém as seguintes partes, sobre o título genérico de Engenharia de Software – Qualidade do Produto
  - Parte 1 – Modelo de Qualidade
  - Parte 2 – Métricas Externas
  - Parte 3 – Métricas Internas
  - Parte 4 – Métricas de Qualidade em Uso



# ISO/IEC 9126

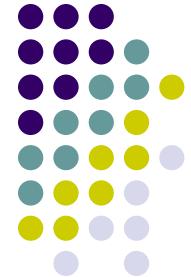
- A versão inicial da ISO/IEC 9126 descrevia seis características de qualidade, bem como um modelo de processo de avaliação de produtos de software
- Como características de qualidade e suas métricas associadas podem ser úteis não somente para a avaliação de um produto de software, mas também para definir requisitos de qualidade, entre outros, a ISO/IEC 9126 (1991) foi substituída por dois padrões relacionados:
  - ISO/IEC 9126 (Qualidade do Produto de Software)
  - ISO/IEC 14598 (Avaliação do Produto de Software)

# ISO/IEC 9126: Modelo de Qualidade



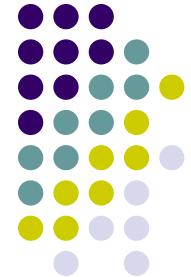
- O modelo de qualidade para produtos de software está dividido em duas partes:
  - Qualidade Interna e Qualidade Externa
  - Qualidade em Uso
- A primeira parte define seis características para qualidade interna e externa, que são divididas em subcaracterísticas
- Essas subcaracterísticas são manifestadas externamente quando o software é usado como parte de um sistema computacional, e como resultado de atributos internos do software

# ISO/IEC 9126: Modelo de Qualidade



- A segunda parte do modelo especifica quatro características de qualidade em uso
- As características definidas são aplicáveis a qualquer tipo de software, incluindo programas de computador e dados contidos em um firmware
- As características e subcaracterísticas provêm uma terminologia consistente para qualidade de produto de software
- Elas também provêm um framework para a especificação de requisitos de qualidade de software, e para a tomada de decisão entre diferentes capacidades de um produto de software

# ISO/IEC 9126: Modelo de Qualidade



- Exemplos de uso do modelo de qualidade da ISO/IEC 9126 são:
  - Validar a completude da definição de requisitos
  - Identificar requisitos de software
  - Identificar objetivos para o design de software
  - Identificar objetivos para teste de software
  - Identificar critérios de garantia da qualidade
  - Identificar critérios de aceitação para um produto de software completo

# ISO/IEC 9126 e a ISO/IEC 12207



- A ISO/IEC 9126 pode ser usada em conjunto com a ISO/IEC 12207 para estabelecer:
  - Um framework para definição de requisitos de qualidade de produtos de software, nos processos primários do ciclo de vida
  - Suportar a revisão, verificação e validação nos processos de suporte de ciclo de vida

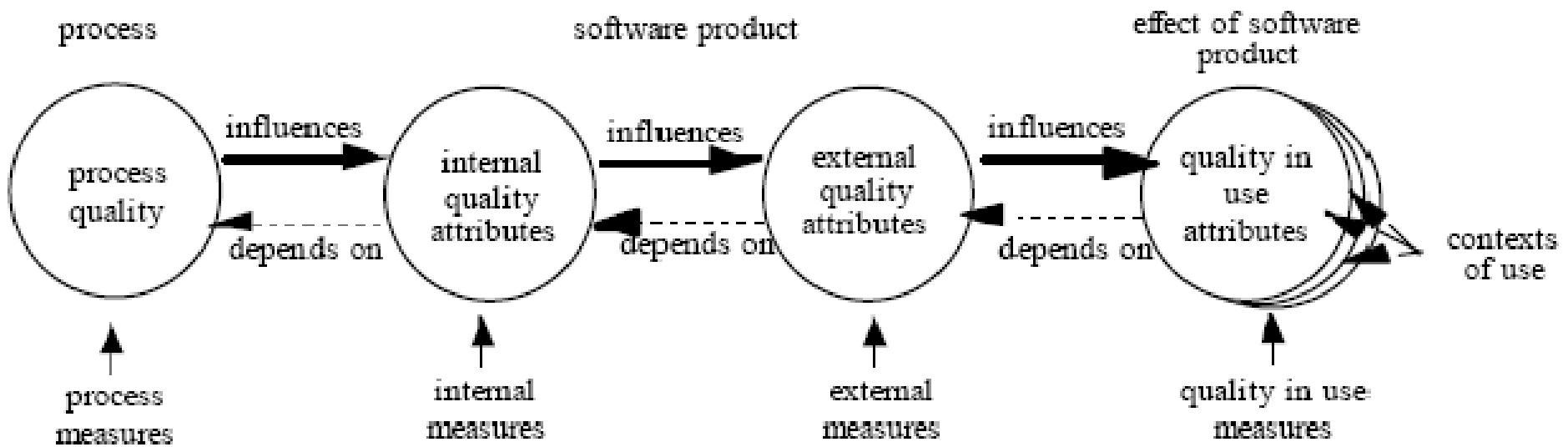
# ISO/IEC 9126 e a ISO/IEC 15504



- A ISO/IEC 9126 pode ser usada em conjunto com a ISO/IEC 15504 para estabelecer:
  - Um framework para definição de qualidade de produto dentro do grupo de processo CUS
  - Suportar a revisão, verificação e validação, e um framework para avaliação quantitativa da qualidade, nos processos SUP
  - Suportar a definição de objetivos organizacionais de qualidade nos processos MAN



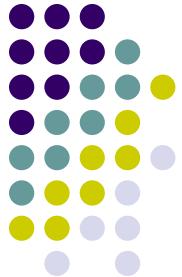
# Caminhos para a Qualidade





# Caminhos para a Qualidade

- Necessidades do usuário por qualidade inclui requisitos de qualidade em uso em contextos específicos de uso
- Essas necessidades podem ser usadas para especificar a qualidade interna e externa, usando as características e subcaracterísticas da qualidade de produtos de software



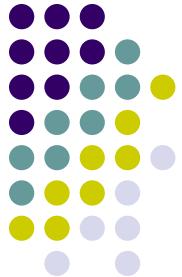
# Caminhos para a Qualidade

- A avaliação de produtos de software de forma a satisfazer necessidades de qualidade de software é um dos processos no ciclo de vida de desenvolvimento de software
- A qualidade do produto de software pode ser avaliada pela medição de:
  - Atributos internos (tipicamente medidas estáticas de produtos intermediários)
  - Atributos externos (tipicamente pela medição do comportamento do código quando executado)
  - Atributos de qualidade em uso



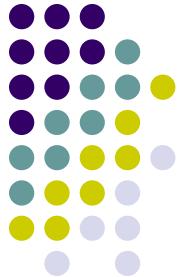
# Caminhos para a Qualidade

- Qualidade do processo (como definida na ISO/IEC 12207) contribui para a melhoria da qualidade do produto,
- Qualidade do produto contribui para melhorar a qualidade em uso
- Desta forma, avaliar e melhorar um processo é um meio para melhorar a qualidade do produto
- Avaliar e melhorar a qualidade do produto é um meio de melhorar a qualidade em uso
- A avaliação da qualidade em uso fornece feedback para melhorar o produto, bem como a avaliação do produto pode fornecer feedback para melhorar o processo



# Itens a serem avaliados

- Itens podem ser avaliados por medições diretas, ou indiretamente ao medir suas consequências
- Por exemplo, um processo pode ser avaliado indiretamente ao medir e avaliar seu produto, e um produto pode ser indiretamente avaliado ao se medir a execução da tarefa por um usuário (usando métricas de qualidade em uso)



# Itens a serem avaliados

- Além disso, deve ser feita uma distinção entre a avaliação do produto de software e a avaliação do sistema no qual é executado
  - Por exemplo, a confiança de um sistema é avaliada pela observação de todas as falhas de qualquer natureza (hardware, software, erros humanos, etc.), enquanto que a confiança de um produto de software é avaliada por apenas aqueles que foram causados por falhas (originados dos requisitos, design ou implementação) no software



# Itens a serem avaliados

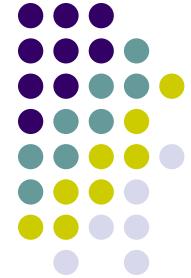
- As fronteiras do sistema também devem ser levadas em consideração, durante a avaliação
  - Por exemplo, se os usuários de uma aeronave com piloto automático computadorizado são os passageiros, então o sistema no qual eles dependem inclui a tripulação, a aeronave, o hardware e o software do piloto automático
  - Se, por outro lado, se apenas a tripulação é considerada como usuários, então o sistema no qual eles dependem consiste apenas da aeronave e do sistema de piloto automático

# Usando um modelo de qualidade



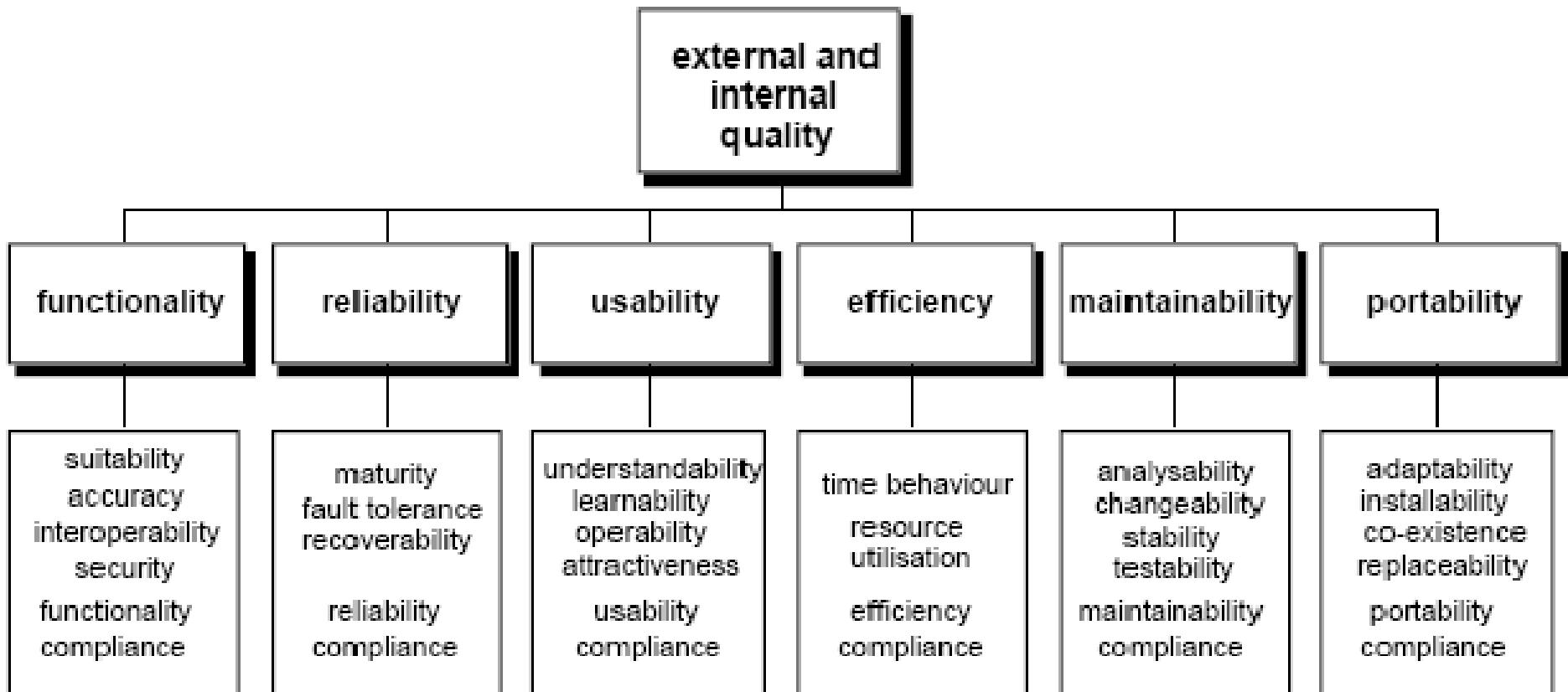
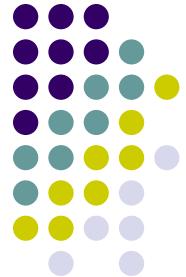
- A qualidade do produto de software deve ser avaliada usando um modelo definido de qualidade
- O modelo de qualidade deve ser usado ao especificar objetivos de qualidade para produtos de software, bem como produtos intermediários
- Qualidade do produto de software pode ser hierarquicamente decomposta em características e subcaracterísticas, que podem ser usadas como um checklist de problemas relacionados à qualidade

# Modelo de Qualidade para Qualidade Externa e Interna

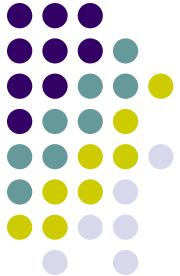


- Atributos de qualidade de software são categorizados em seis características (funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade)
- Cada característica é dividida em subcaracterísticas, que podem ser medidas por métricas internas ou externas

# Modelo de Qualidade para Qualidade Externa e Interna



# Modelo de Qualidade para Qualidade Externa e Interna



- Definições serão dadas para cada característica de qualidade e as subcaracterísticas de software que influenciam na característica de qualidade
- Para cada característica e subcaracterística, a capacidade do software é determinada por um conjunto de atributos internos que podem ser mensurados



# Funcionalidade

- É a capacidade do produto de software de prover funções que atinjam as necessidades declaradas e intrínsecas, quando o software é usado em condições específicas
  - Esta característica se preocupa com o que o software faz para atingir as necessidades, enquanto as demais características são principalmente voltadas para quando e como as necessidades são atingidas



# Funcionalidade: Adequação

- A capacidade do produto de software de prover um conjunto adequado de funções para tarefas específicas e objetivos do usuário



# Funcionalidade: Acurácia

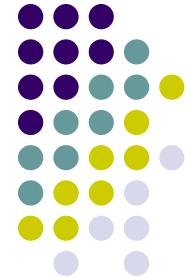
- A capacidade do produto de software de prover os resultados corretos ou acordados, ou efeitos com o grau necessário de precisão

# Funcionalidade: Interoperabilidade

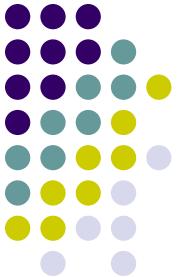


- A capacidade do produto de software de interagir com um ou mais sistemas específicos

# Funcionalidade: Segurança de Acesso



- A capacidade do produto de software de proteger informações e dados
- Pessoas ou sistemas não autorizados não podem ler ou modificar informações e dados
- Pessoas ou sistemas autorizados não tem acesso negado às informações e dados
- Também se aplica à transmissão de dados
- Nota: segurança como um todo é definida como característica de qualidade em uso, já que não se relaciona com o software sozinho, mas com todo o sistema



# Funcionalidade: Conformidade

- A capacidade do produto de software de aderir à padrões, convenções ou regulamentos em Leis e prescrições similares relacionadas à funcionalidade



# Confiabilidade

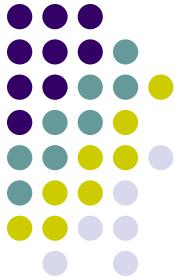
- A capacidade do produto de software de manter um nível específico de performance quando usado em condições específicas
- Desgaste ou envelhecimento não ocorrem em software. Limitações em confiabilidade ocorrem devido à faltas em requisitos, design e implementação. Falhas devido à esses problemas dependem da forma como o produto de software é usado e as opções acessadas, e não devido à idade do software



# Confiabilidade: Maturidade

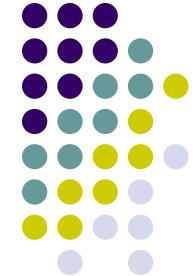
- A capacidade do produto de software evitar falhas como resultado de faltas no software

# Confiabilidade: Tolerância à Falhas



- A capacidade do produto de software de manter um nível especificado de performance em casos de faltas de software ou de mal uso das suas interfaces especificadas

# Confiabilidade: Recuperabilidade



- A capacidade do produto de software de re-estabelecer um nível especificado de performance e recuperar os dados diretamente afetados em caso de falha
- Após uma falha, o produto de software irá, algumas vezes, ficar fora do ar por um certo período de tempo, cuja extensão é avaliada pela sua recuperabilidade

# Confiabilidade: Recuperabilidade



- Disponibilidade é a capacidade do produto de software estar em um estado para executar uma função necessária em um dado momento, sob certas condições de uso.
- Externamente, a disponibilidade pode ser avaliada pela proporção do tempo total durante o qual o produto de software está funcionando.
- Disponibilidade é então uma combinação de maturidade (que governa a freqüência das falhas), tolerância à falhas e recuperabilidade (que governa a extensão do tempo para o ar após cada falha).
- Por essa razão, disponibilidade não é considerada uma subcaracterística separada



# Confiabilidade: Conformidade

- A capacidade do produto de software de aderir a padrões, convenções ou regulamentos relacionados à confiabilidade



# Usabilidade

- A capacidade do produto de software de ser entendido, aprendido, usado e atraído ao usuário, quando usado em condições específicas
- Usuários podem incluir operadores, usuários finais e usuários indiretos que estão sob influência de ou dependem do uso do software
- Usabilidade deve endereçar todos os diferentes ambientes de usuários que o software pode afetar



# Usabilidade: Intelligibilidade

- A capacidade do produto de software de possibilitar o usuário a entender se o software é aplicável, e como ele pode ser usado em tarefas e condições particulares de uso
- Isso dependerá da documentação e impressões iniciais dadas pelo software



# Usabilidade: Apreensibilidade

- A capacidade do produto de software de habilitar o usuário a aprender sua aplicação

# Usabilidade: Operacionabilidade



- A capacidade do produto de software de habilitar o usuário a operar e controlar o software
- Aspectos de adequação, modificabilidade, adaptabilidade e instalabilidade podem afetar a operacionabilidade
- Para um sistema que é operado por um usuário, a combinação de funcionalidade, confiabilidade, usabilidade e eficiência podem ser medidas externamente pela qualidade em uso



# Usabilidade: Atratividade

- A capacidade do produto de software de ser atrativo ao usuário
- Refere-se a atributos do software intencionados a fazê-lo mais atrativo ao usuário, como o uso de cores e a natureza do design gráfico



# Usabilidade: Conformidade

- A capacidade do produto de software de aderir a padrões, convenções, guias de estilo ou regulações relativas à usabilidade



# Eficiência

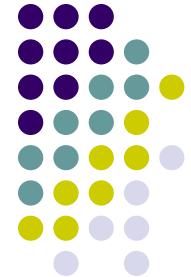
- A capacidade do produto de software prover performance apropriada, relativa à quantidade de recursos usados, dentro de condições específicas
- Recursos podem incluir outros produtos de software, a configuração de software e hardware do sistema, e materiais (papel, disquetes, etc.)
- Para um sistema operado por um usuário, a combinação de funcionalidade, confiabilidade, usabilidade e eficiência podem ser mensuradas externamente pela qualidade em uso

# Eficiência: Comportamento em Relação ao Tempo



- A capacidade do produto de software de prover tempos apropriados de resposta e processamento, bem como taxas de rendimentos quando executando sua função, sobre condições específicas

# Eficiência: Utilização de Recursos



- A capacidade do produto de software de usar quantidades e tipos apropriados de recursos, quando o software executa sua função sobre condições específicas
- Observação: recursos humanos são incluídos como parte da produtividade – qualidade em uso



# Eficiência: Conformidade

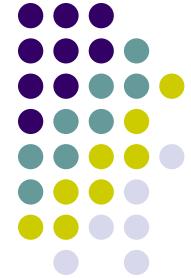
- A capacidade do produto de software de aderir a padrões ou convenções relativos à eficiência



# Manutenabilidade

- A capacidade do produto de software de ser modificado
- Modificações podem incluir correções, melhorias ou adaptações do software à mudanças no ambiente, requisitos e especificações funcionais

# Manutenabilidade: Analisabilidade



- A capacidade do produto de software de ser diagnosticado em relação à deficiências ou causas de falhas no software, ou para a identificação das partes a serem alteradas

# Manutenabilidade: Modificabilidade



- A capacidade do produto de software de possibilitar que uma modificação especificada possa ser implementada
- Implementação inclui codificação, design e mudança na documentação

# Manutenabilidade: Estabilidade



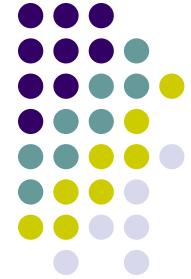
- A capacidade do produto de software de evitar efeitos inesperados de modificações no software

# Manutenabilidade: Testabilidade



- A capacidade do produto de software de possibilitar o software modificado de ser validado

# Manutenabilidade: Conformidade



- A capacidade do produto de software de aderir a padrões ou convenções relativas à manutenabilidade



# Portabilidade

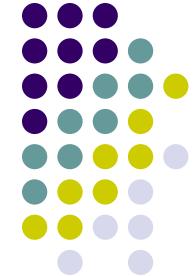
- A capacidade do produto de software de ser transferido de um ambiente para outro
- O ambiente pode ser organizacional, hardware ou software



# Portabilidade: Adaptabilidade

- A capacidade do produto de software de ser adaptado para diferentes ambientes especificados, sem tomar ações ou meios além daquelas providas para este propósito, pelo software considerado

# Portabilidade: Capacidade para ser instalado



- A capacidade do produto de software de ser instalado em um ambiente específico
- Se o software é para ser instalado pelo usuário final, a capacidade para ser instalada pode afetar a adequação e operabilidade



# Portabilidade: Co-existência

- A capacidade do produto de software de co-existir com outros software independentes em um ambiente comum, compartilhando recursos comuns

# Portabilidade: Capacidade para Substituir



- A capacidade do produto de software ser usado no lugar de outro software específico para o mesmo propósito no mesmo ambiente
- Por exemplo, a capacidade para substituir de uma nova versão de um produto de software é importante quando o usuário faz o upgrade



# Portabilidade: Conformidade

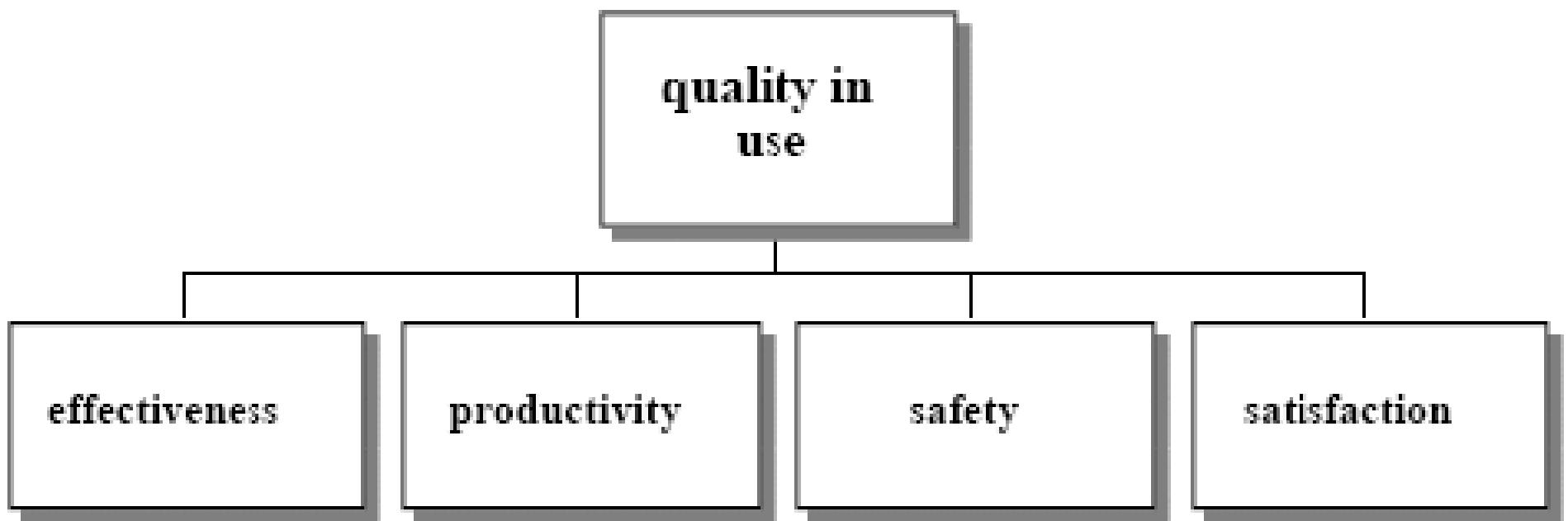
- A capacidade do produto de software de aderir a padrões ou convenções relacionadas à portabilidade

# Modelo de Qualidade para Qualidade em Uso



- Os atributos da qualidade em uso são categorizados em quatro características: Efetividade, Produtividade, Segurança, Satisfação
- Qualidade em uso é a visão do usuário sobre qualidade.
- Alcançar qualidade em uso é dependente de alcançar a necessária qualidade externa, que por sua vez é dependente de se alcançar a necessária qualidade interna.
- Mediçãoes são normalmente necessárias em todos os três níveis, já que os critérios para medições internas não são normalmente suficientes para garantir sucesso nos critérios de medições externas, bem como critérios para medições externas não são suficientes para garantir sucesso em medições da qualidade em uso

# Modelo de Qualidade para Qualidade em Uso





# Qualidade em Uso

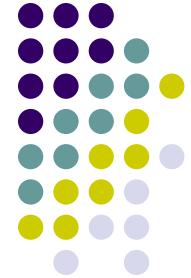
- A capacidade do produto de software de possibilitar que usuários específicos atinjam objetivos específicos com efetividade, produtividade, segurança e satisfação, em um contexto de uso específico
- Qualidade em uso é a visão do usuário sobre qualidade de um ambiente contendo software, e é mensurado dos resultados do uso do software neste ambiente, ao invés de propriedades do software



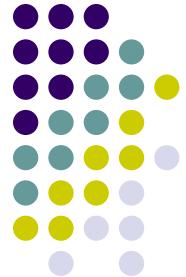
# Qualidade em Uso: Efetividade

- A capacidade do produto de software de possibilitar seus usuários a atingir os objetivos especificados com precisão e completude, em um contexto específico de uso

# Qualidade em Uso: Produtividade



- A capacidade do produto de software de possibilitar seus usuários a gastar quantidades apropriadas de recursos em relação à efetividade atingida em um contexto específico de uso
- Recursos relevantes podem incluir tempo para completar a tarefa, o esforço do usuário, materiais, ou o custo financeiro de uso



# Qualidade em Uso: Segurança

- A capacidade do produto de software de atingir níveis aceitáveis de risco a danos a humanos, negócios, propriedades ou ambiente em um determinado contexto de uso
- Riscos são normalmente resultados de deficiências na funcionalidade (incluindo segurança de acesso), confiabilidade, usabilidade ou manutenibilidade



# Qualidade em Uso: Satisfação

- A capacidade do produto de software de satisfazer usuários em um determinado contexto de uso
- Satisfação é a resposta do usuário para interação com o produto, e incluir atitudes em direção ao uso do produto