

Atributos essenciais de um bom software.

Características do produto

- 1- Aceitabilidade → O software deve ser aceitável para o tipo de usuário para o qual é projetado. Isso significa que ele deve ser inteligível, útil e compatível com os outros sistemas utilizados pelos usuários.
- 2- Dependabilidade e segurança da informação (security) → A dependabilidade do software inclui uma gama de características, incluindo confiabilidade, segurança da informação (security) e segurança (safety). O software com dependabilidade não deve causar danos físicos ou econômicos em caso de falha sistema. Ele também deve ser protegido para que usuários maliciosos não consigam acessar ou danificar o sistema.
- 3- Eficiência → O software não deve desperdiçar recursos do sistema, como a memória e os ciclos do processador. Portanto, a eficiência inclui responsividade, tempo de processamento, utilização de recursos etc.
- 4- Manutenibilidade → O software deve ser escrito de tal modo que possa evoluir e satisfazer as necessidades mutáveis dos clientes. Este é um atributo crítico, pois a modificação do software é um requisito inevitável de um ambiente empresarial mutável.

~~~~~ // ~~~~~

● { A abordagem sistemática utilizada na engenharia de software é, às vezes, chamada de processo de software, uma sequência de atividades que leva a produção de um software.

1- Especificação de software, etapa em que clientes e engenheiros definem o software que deve ser produzido e as restrições impostas à sua operação.

2- Desenvolvimento de software, etapa em que o software é projetado e programado.

3- Validação de software, etapa em que o programa é analisado para garantir que seja aquilo que o cliente precisa.

4- Evolução do software, etapa de modificação para refletir a mudança de requisitos tanto do cliente quanto do mercado.

————— || —————

talvez o fator mais importante na determinação de quais são os métodos mais importantes (na engenharia de software) é o tipo de aplicação que está sendo desenvolvido.

1- Aplicações stand-alone

→ São sistemas de aplicação executados em um computador pessoal ou aplicativos que rodam em dispositivos móveis. Elas incluem toda a funcionalidade necessária e podem não necessitar de conexão a uma rede.

ex {  
→ CAD  
→ software de manipulação de imagem  
→ aplicativos de viagens  
→ aplicativos de produtividade

## 2- Aplicações interativas baseadas em transações.

→ São aplicações executadas em um computador remoto e que são acessadas por usuários a partir de seus próprios computadores, smartphones ou tablets. Elas incluem aplicações web como as de comércio eletrônico, por exemplo, por meio das quais se interage com um sistema remoto para comprar bens e serviços.

## 3- Sistemas de controle embarcados

→ São sistemas de controle de software que controlam e gerenciam dispositivos de hardware. Em números, existem provavelmente mais sistemas embarcados do que qualquer outro tipo.

- software que controla o freio ABS em um carro
- software em um forno micro-ondas que controle o processo de cozimento
- software de um telefone celular

## 4- Sistemas de processamento em lotes (batch)

→ São sistemas de negócios concebidos para processar dados em grandes lotes. Eles processam números enormes de entradas individuais para criar as saídas correspondentes. Exemplos incluem o faturamento periódico, como as contas de telefone, e os sistemas de folha de pagamento.

## 5- Sistemas de entretenimento

→ São destinados para o uso pessoal, para entreter o usuário. A maioria desses sistemas consiste em jogos de gêneros variados, que

podem ser executados em um console concebido especificamente para essa finalidade. A qualidade da interação com o usuário é a característica diferenciadora mais importante dos sistemas de entretenimento.

#### 6- Sistemas para modelagem e simulação

→ São desenvolvidos por cientistas e engenheiros para modelar processos físicos ou situações que incluem muitos objetos que interagem. Costumam ser computacionalmente intensivos e demandam sistemas paralelos de alto desempenho para a sua execução.

#### 7- Sistemas de coleta de dados e análise

→ São aqueles que fazem a sua coleta no ambiente e enviam dados para outros sistemas, para processamento. O software pode ter de interagir com sensores e frequentemente é instalado em um ambiente hostil, como o interior de um motor, ou em uma localização remota. A análise de "Big Data" (grandes volumes de dados) pode envolver sistemas baseados na nuvem executando análises estatísticas e procurando relações entre os dados coletados.

#### 8- Sistemas de sistemas

→ São utilizados em empresas e outras grandes organizações e são compostos de uma série de outros sistemas de software. Alguns deles podem ser produtos de software genéricos, como um sistema ERP. Outros sistemas do conjunto podem ser desenvolvidos especialmente por esse ambiente.

# Fundamentos da Engenharia de Software que se aplicam a todos os tipos de sistemas de software:

- 1- Eles devem ser desenvolvidos com o uso de um processo gerenciado e compreendido. A organização que está desenvolvendo o software deve planejar o processo de desenvolvimento e ter ideias claras do que será produzido e de quando será concluído. Naturalmente, o processo específico deve utilizado depende do tipo de software que está desenvolvendo.
- 2- Dependabilidade e desempenho são importantes para todos os tipos de sistema. O software deve se comportar conforme o esperado, sem falhas, e deve estar disponível para uso quando for necessário. Deve ter uma operação segura e, na medida do possível, ter proteção contra ataques externos. O sistema deve ter um desempenho eficiente e não desperdiçar recursos.
- 3- É importante compreender e controlar a especificação e os requisitos do software (o que o software deve fazer). É preciso saber o que os diferentes clientes e usuários esperam do sistema e gerenciar as expectativas deles a fim de fornecer um sistema útil dentro do orçamento e do cronograma.
- 4- Os recursos existentes devem ser usados de modo eficaz. Isso significa que, onde for apropriado, deve-se reusar software que já tenha sido desenvolvido, em vez de escrever um novo.

Esses fundamentos independem da linguagem de programação utilizada no desenvolvimento do software.