



UNIDADE III

Engenharia de Software

Prof. Me. Edson Moreno

Introdução

- Esta unidade é essencial para compreender o papel crítico da interface homem-computador na qualidade do software, as normas que regem essa qualidade e as práticas relacionadas à evolução, reengenharia e manutenção de software, bem como as tendências emergentes na indústria de software.
- Uma abordagem de grande importância na indústria do software é a Interação Humano-Computador (IHC). Com interfaces entre homem e computador cada vez mais aprimoradas, é uma indústria do design que cada vez mais necessita de analistas de dados e programadores.
- A indústria do software não para e isso ocorre pelo processo de melhoria contínua de automatização do conhecimento.
 - A proposta da disciplina é capacitar o aluno no conhecimento e práticas profissionais da Engenharia de Software.

Interface homem-computador, qualidade e avaliação

A Interface Homem-Computador (IHC) desempenha um papel crucial ao garantir que os sistemas sejam acessíveis, usáveis e eficazes para os usuários.

Esta unidade aborda aspectos como: princípios da interação humano-computador, qualidade em uso, modos de acesso, retorno de investimento e normas de qualidade.

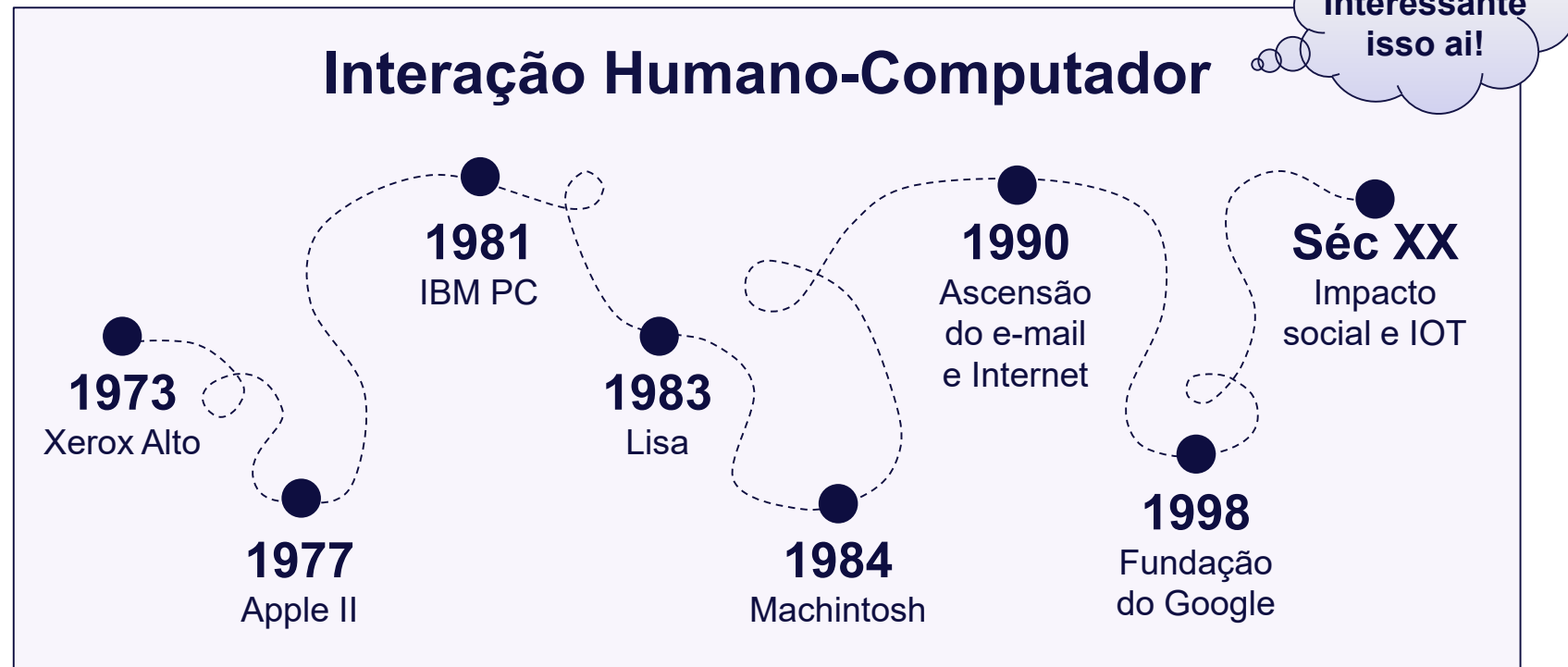
- A IHC concentra-se na criação de sistemas que proporcionem uma interação intuitiva e eficiente entre humanos e computadores.
 - Os princípios da IHC abrangem a usabilidade, a acessibilidade e a comunicabilidade, visando garantir que os sistemas atendam às necessidades e às expectativas dos usuários.
 - Modos de acesso e a interação com dispositivos.
 - Retorno de Investimentos (ROI).

Princípios da interação humano-computador (IHC)



Fonte: <https://x.gd/GpIX1>. Acesso em: 22 jan. 2024.

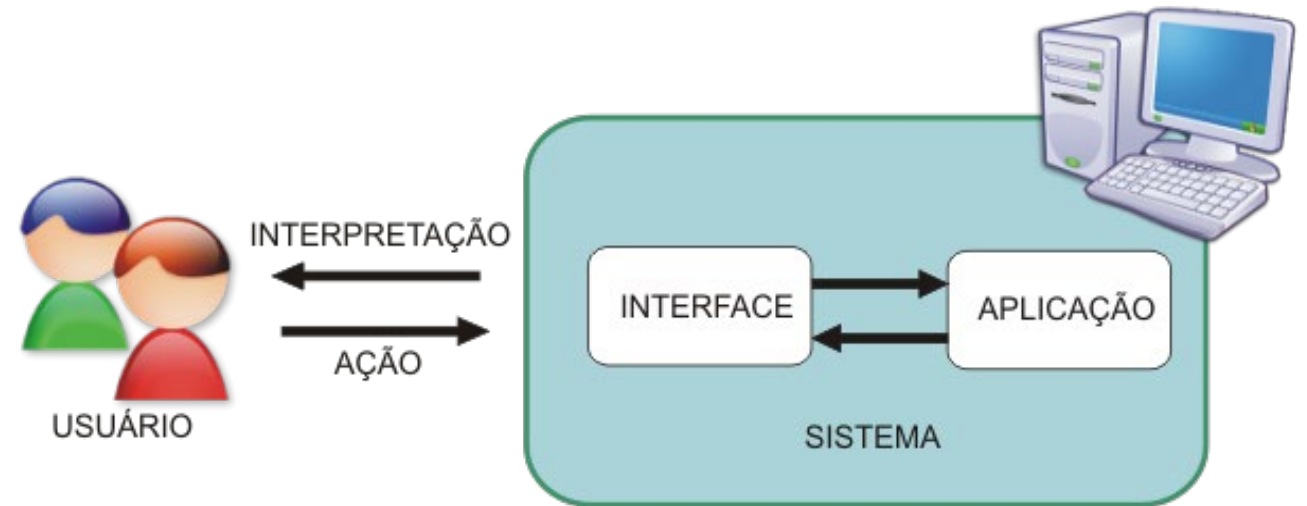
- Frustração e ansiedade são partes da vida diária para muitos usuários de sistemas de software.
- A figura ao lado ilustra bem essa situação em que os usuários lutam para entender um menu de opções que se destinam a ajudá-los.



Fonte: Softplan. IHC: Saiba o que é e os focos de estudo da Interação Humano-Computador. Adaptado de: <https://www.softplan.com.br/tech-writers/ihc/>. Acesso em 22/01/2024.

Definição de interface e interação

- Na computação, a interface é definida como a parte do sistema computacional com a qual a pessoa entra em contato: físico, perceptivo ou conceitualmente.
- Exemplos: teclado para entrada de dados no computador pelo ser humano, vídeo para visualizar dados do computador para o ser humano.



Fonte: Fortunato (2020).

- Na computação, interação é definida como o processo de comunicação entre pessoas e sistemas interativos.
- Exemplos: execução de um comando pelo computador bem como o resultado fornecido pelo computador.

Interface com o usuário: modos de acesso dos sistemas operacionais

- O conceito de projeto de interface com o usuário é criar um meio efetivo de comunicação entre o ser humano e o computador. Seguindo um conjunto de princípios de projeto de interface, o projeto identifica objetos e ações de interface e depois se cria um layout de tela que forma a base para um protótipo de interface com o usuário (Pressman, 2011).

Existem basicamente três modos de acesso aos sistemas de software:

1. Interface de texto, linha de comando ou 1D.
2. Interface gráfica ou 2D (duas dimensões).
3. Interfaces 3D (três dimensões).

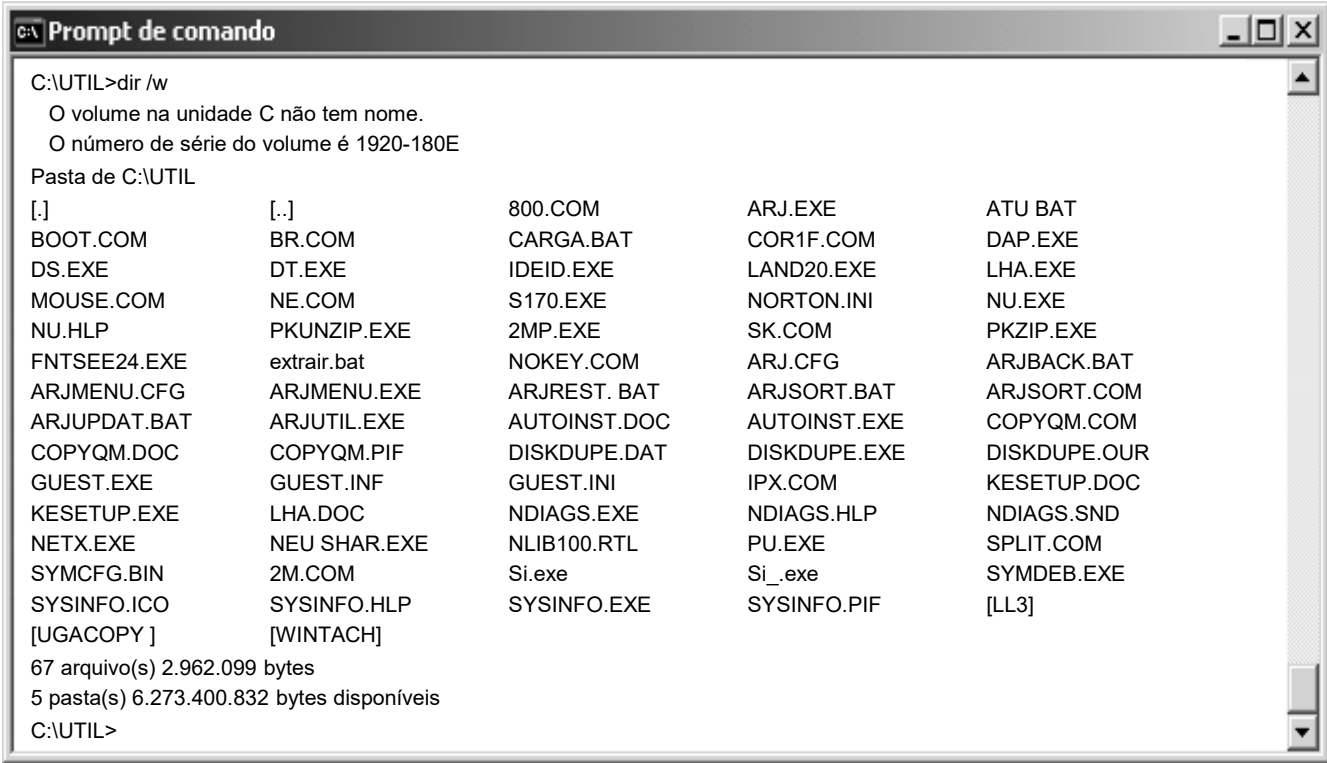
Modo de acesso 1D – Interface de texto ou linha de comando

```
root /usuario]# ls -l..

drwxr-xr-x  2  root  root  4096  Jul   6   2001  bin
drwxr-x--x 11  root  root  4096  Mar  26  17:50  root
drwxrwxrwt 10  root  root  4096  Mar  26  17:43  etc
drwxr-xr-x  2  root  root  4096  Mar  26  17:51  usuario
drwxr-xr-x  2  root  root  4096  Mar  26  18:15  docs
- rw - - - - - 1  root  root  1599  Mar  26  18:20  ipsec.conf
- rw -r--r-- 1  root  root    37  Mar  26  18:22  issue
drwxr-xr-x  2  root  root  4096  Mar  26  18:13  texto
```

Fonte: adaptado de: Moreno (2013).

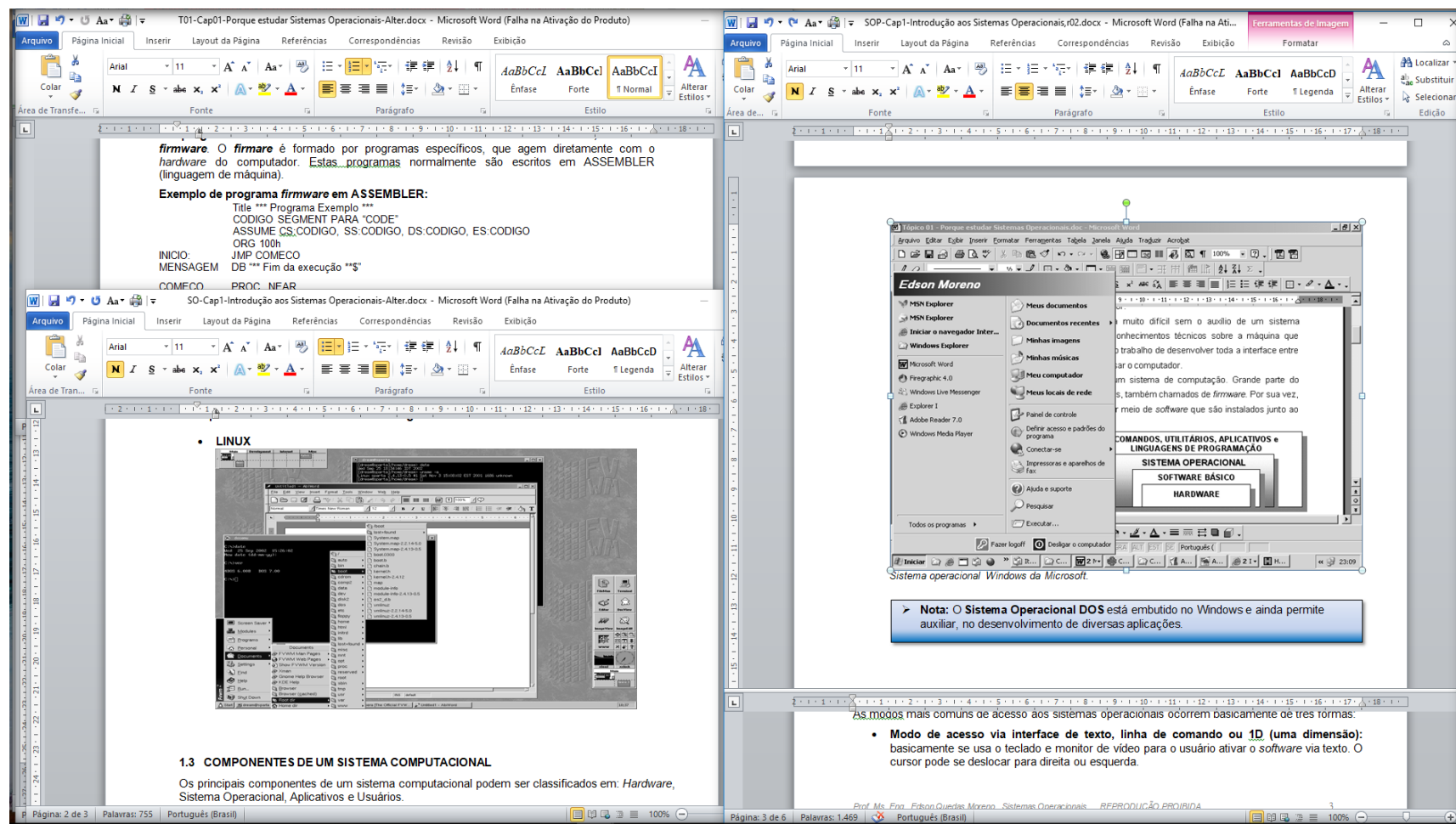
- É 1D porque o cursor se desloca na horizontal, para direita ou esquerda. Usa-se o teclado e monitor de vídeo para comandar o computador.
- Exemplos: Telas do DOS e do LINUX em modo texto. Veja as figuras. A do lado é do LINUX e a de baixo é do DOS.



Modo de acesso 2D – Interface gráfica

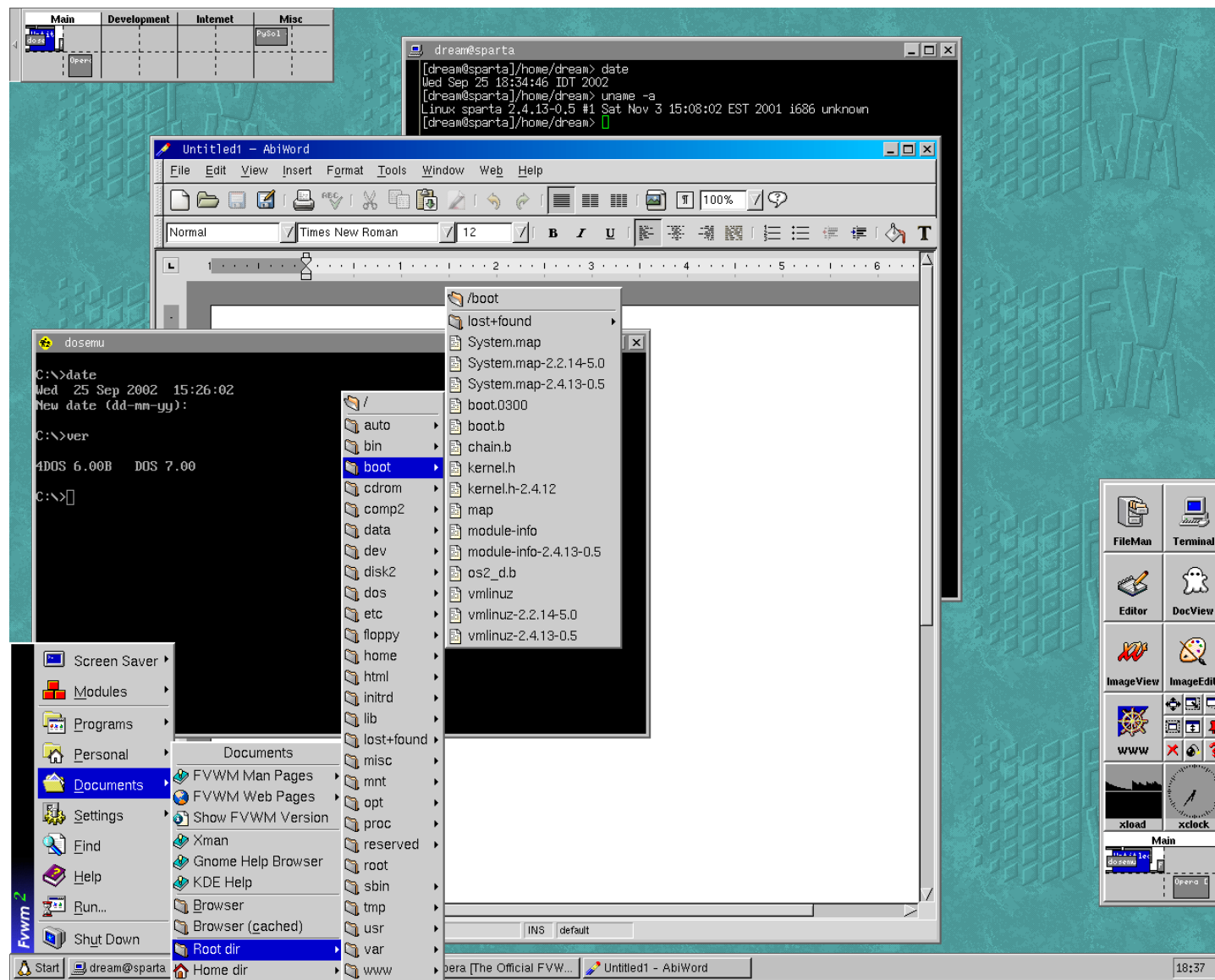
- É 2D porque o cursor se desloca na horizontal e vertical. Além do teclado e vídeo, usa-se também o mouse para comandar o computador.

- Exemplo: Interface Gráfica do Usuário (GUI – Graphical User Interface) do Windows.



Modo de acesso 2D – Interface gráfica

- Exemplo: Linux.



Modo de acesso 3D

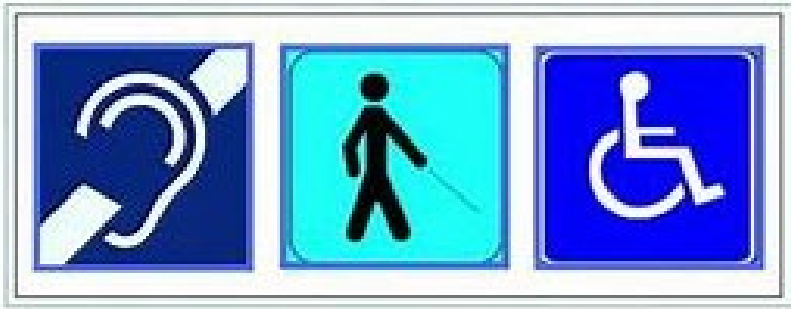
- É 3D porque o ambiente operacional é tridimensional (horizontal, vertical e profundidade).
- Esse tipo de interface permite, por meio dos sentidos e características dos humanos, a comunicação com o computador.
- É uma interface que imita alguns aspectos do ser humano, tais como: manipulação por voz, por movimentos, pressão, movimento dos olhos e várias outras formas.



Fonte: Computerworld.
Disponível em:
<https://brasil.uxdesign.cc/aplicacao-a7b5es-da-realidade-aumentada-na-educacao-a7a3oe9c6c114cb50>

Qualidade em uso: usabilidade, comunicabilidade e acessibilidade

ACESSIBILIDADE Direito de Todos



Fonte: CMDPD (2014).

- A qualidade em uso se refere à facilidade de uso, às funcionalidades e suporte apropriado às atividades de uso em cenário real.
- Usabilidade: capacidade do software ser compreendido, aprendido, operado e atraente ao usuário, quando usado sob condições específicas. Exemplo: uma característica que permite facilitar a navegação do usuário é colocar em destaque o “botão” a ser pressionado.
- Comunicabilidade: propriedade que transmite ao usuário as intenções e princípios de interação que guiaram o seu design. Exemplo: Menu de acesso de app.
- Acessibilidade: capacidade do software remover barreiras de acesso baseado em limitações técnicas, ambientais e de deficiência. Exemplo: interface de conversão de texto em voz e vice-versa.

Retorno de investimento (ROI)

- O objetivo principal do software é definir uma boa funcionalidade e ser de fácil uso.
- O usuário deve se sentir satisfeito com a aplicação.



Fonte: ClipArt.

O retorno do investimento (do inglês, ROI – Return Over Investment) na IHC pode ser observado na implementação de uma boa interface humano-computador, que atenda às seguintes características:

- Rápido treinamento;
- Motivação à exploração ou navegação;
- Uso de termos associados ao paradigma do usuário, que não os confunda ou induzam ao erro;
- Satisfação de uso;
- Aumento da produtividade.

Atender a essas características melhora a qualidade de uso e aumenta o benefício em relação ao capital investido.

Interatividade

A IHC se apresenta de várias formas. Por exemplo: os faróis de trânsito de grandes metrópoles são controlados por computadores. Dispositivos colocados no asfalto fazem contagem dos veículos. Essa contagem é enviada para os computadores do controle de trânsito que, por sua vez, decidem pela temporização dos luminosos dos faróis de trânsito. Na computação, esses contadores colocados no asfalto são considerados:

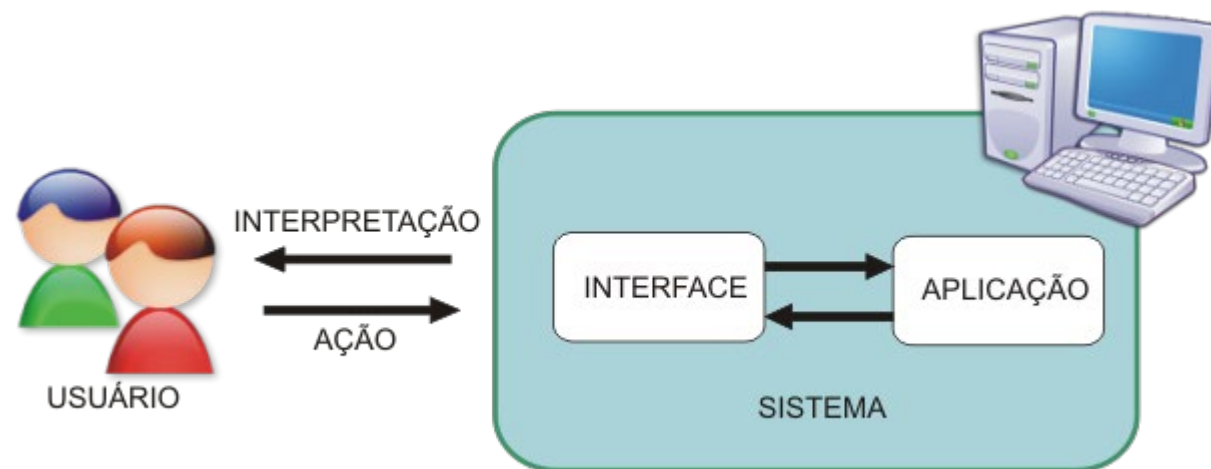
- a) Computadores.
- b) Interfaces.
- c) Memorizadores.
- d) Sistemas.
- e) Usuários.

Resposta

A IHC se apresenta de várias formas. Por exemplo: os faróis de trânsito de grandes metrópoles são controlados por computadores. Dispositivos colocados no asfalto fazem contagem dos veículos. Essa contagem é enviada para os computadores do controle de trânsito que, por sua vez, decidem pela temporização dos luminosos dos faróis de trânsito. Na computação, esses contadores colocados no asfalto são considerados:

- a) Computadores.
- b) Interfaces.**
- c) Memorizadores.
- d) Sistemas.
- e) Usuários.

Fonte: Fortunato (2020).

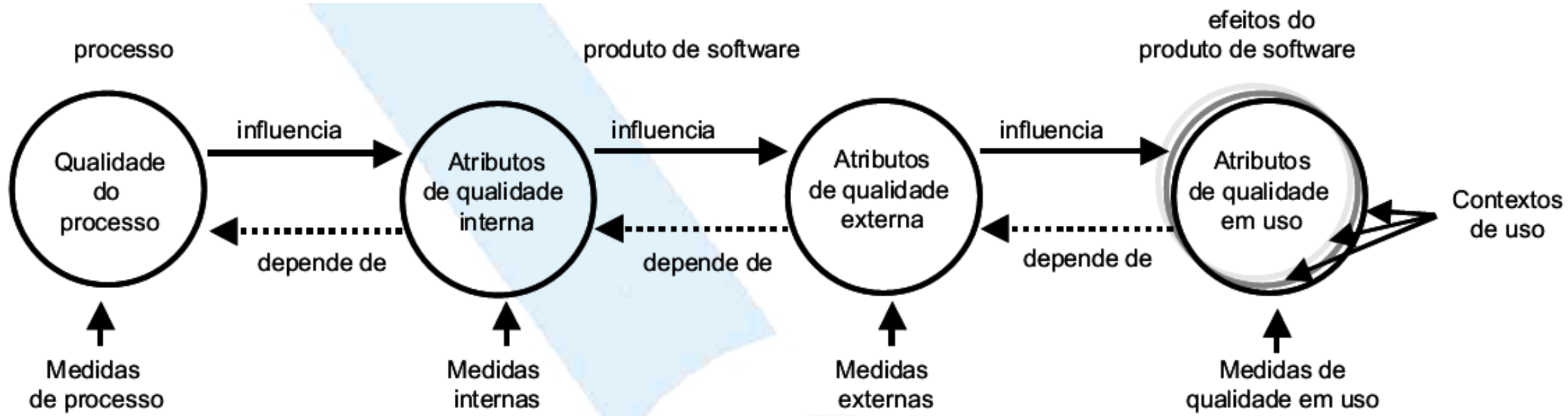


Normas de qualidade: ISO 9126, ISO 14598 e ISO 25000

As normas da qualidade conduzem às implementações e melhorias na IHC e em todo o ciclo de vida do desenvolvimento de software, tais como:

- ISO 9126 (Qualidade do produto da engenharia de software): Define um conjunto de características de qualidade de software, como funcionalidade, confiabilidade, usabilidade e eficiência, fornecendo diretrizes para avaliar e melhorar a qualidade do produto de software;
- ISO 14598 (Avaliação do produto de software): Fornece um conjunto de diretrizes para a avaliação de produtos de software, incluindo métodos e critérios para medir a qualidade do software;
 - ISO 25000 (Guia do SQuaRE - Engenharia de software - Requisitos de qualidade e avaliação): É uma norma que oferece orientação abrangente para a qualidade e avaliação de produtos de software e que engloba a ISO 9126, a ISO 14598 e outras normas relacionadas.

NBR ISO/IEC 9126 – Qualidade do produto da engenharia de software

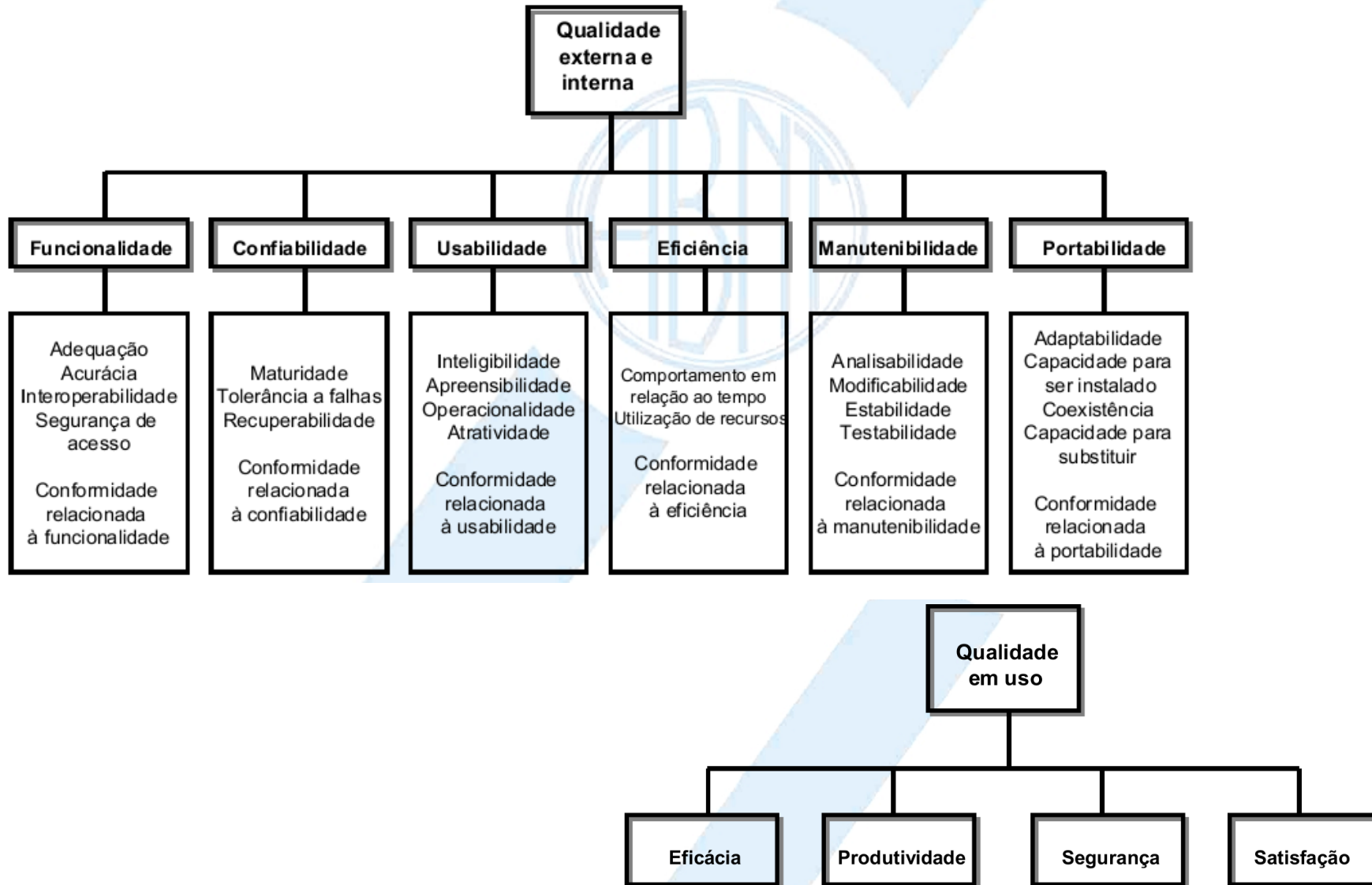


Fonte: NBR ISO/IEC 9126-1 (2003).

O objetivo da NBR ISO/IEC 9126 é descrever um modelo de qualidade do produto de software composto de duas partes:

- a) qualidade interna e qualidade externa.
- b) qualidade em uso.

NBR ISO/IEC 9126: requisitos de qualidade externa e interna e em uso

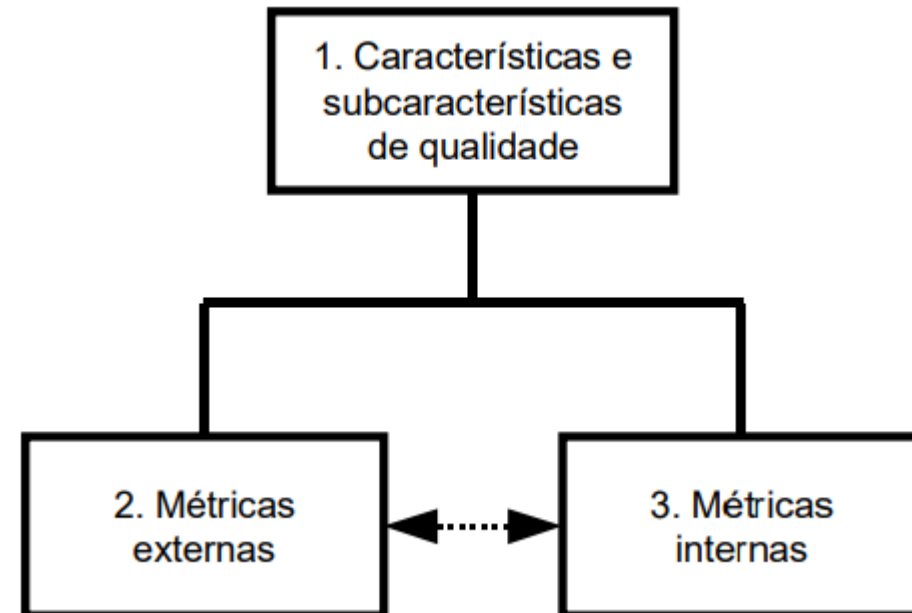


Fonte: NBR ISO/IEC 9126-1 (2003).

- Requisitos de qualidade externa: incluem requisitos derivados das necessidades de qualidade dos usuários, incluindo os requisitos de qualidade em uso.
- Requisitos de qualidade interna: especificam o nível de qualidade requerido sob o ponto de vista interno do produto.
- Requisitos de qualidade em uso: é a visão da qualidade do produto de software do ponto de vista do usuário.

NBR ISO/IEC 14598 – Avaliação do produto de software

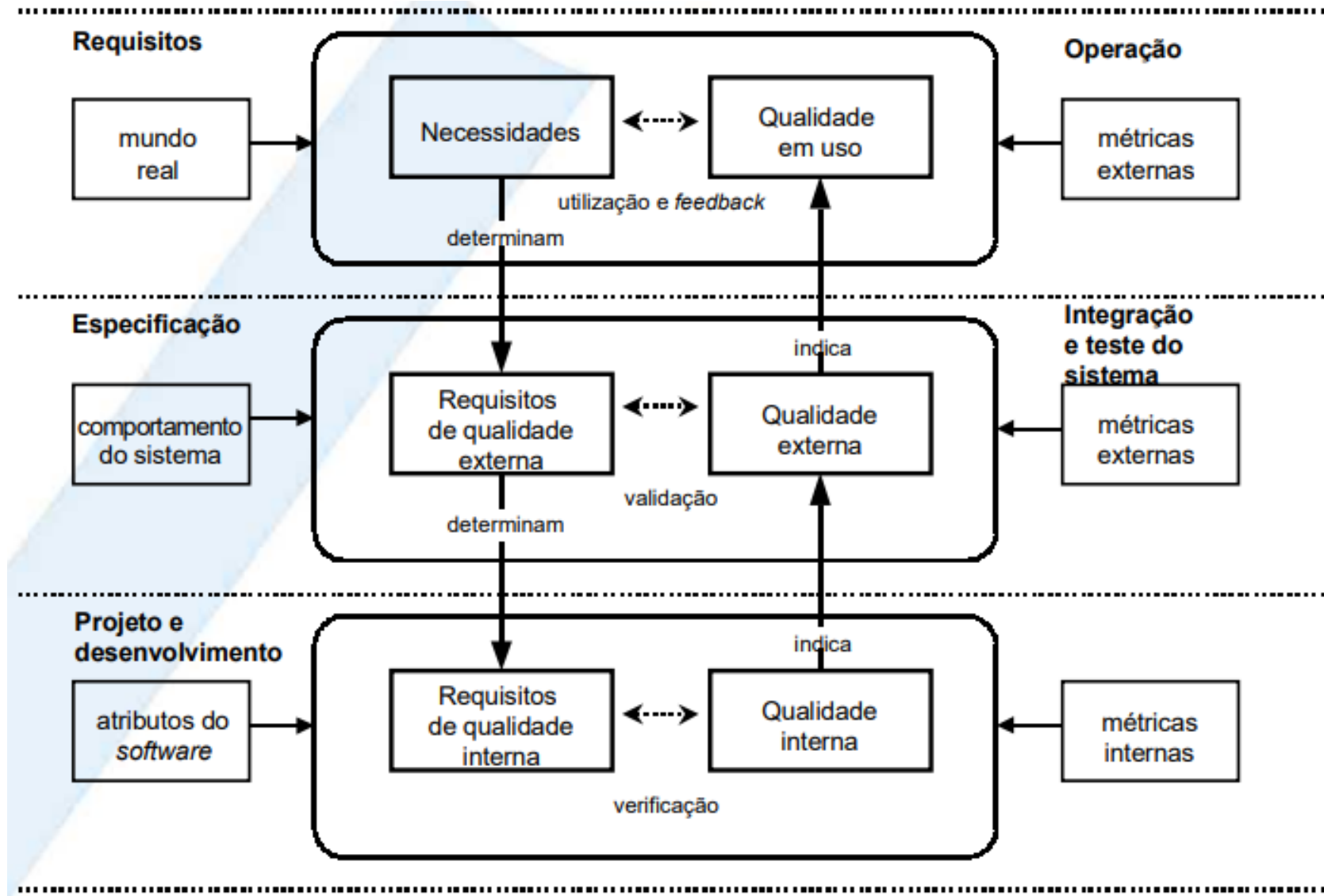
- A NBR ISO/IEC 14598 fornece uma estrutura para avaliar a qualidade de quaisquer produtos de software e estabelece os requisitos para métodos de medição e avaliação de produtos de software.
- Define termos técnicos e nomenclatura comum entre as partes.
- A NBR ISO/IEC 14598 pode ser usada por organizações que produzem novos módulos de avaliação, sendo conveniente sua utilização com a NBR ISO/IEC 9126, que descreve as características de qualidade e métricas de software, como mostra a figura abaixo.



Fonte: NBR ISO/IEC 14598 (2001).

NBR ISO/IEC 14598: framework

- Na qualidade externa, avalia apenas se o sistema estiver completo com hardware e software.
- Na qualidade interna, avalia de acordo com as ligações dos componentes: hardware, software, BD e rede de computadores, código-fonte e desempenho global.



NBR ISO/IEC 25000 – Guia do SQuaRE

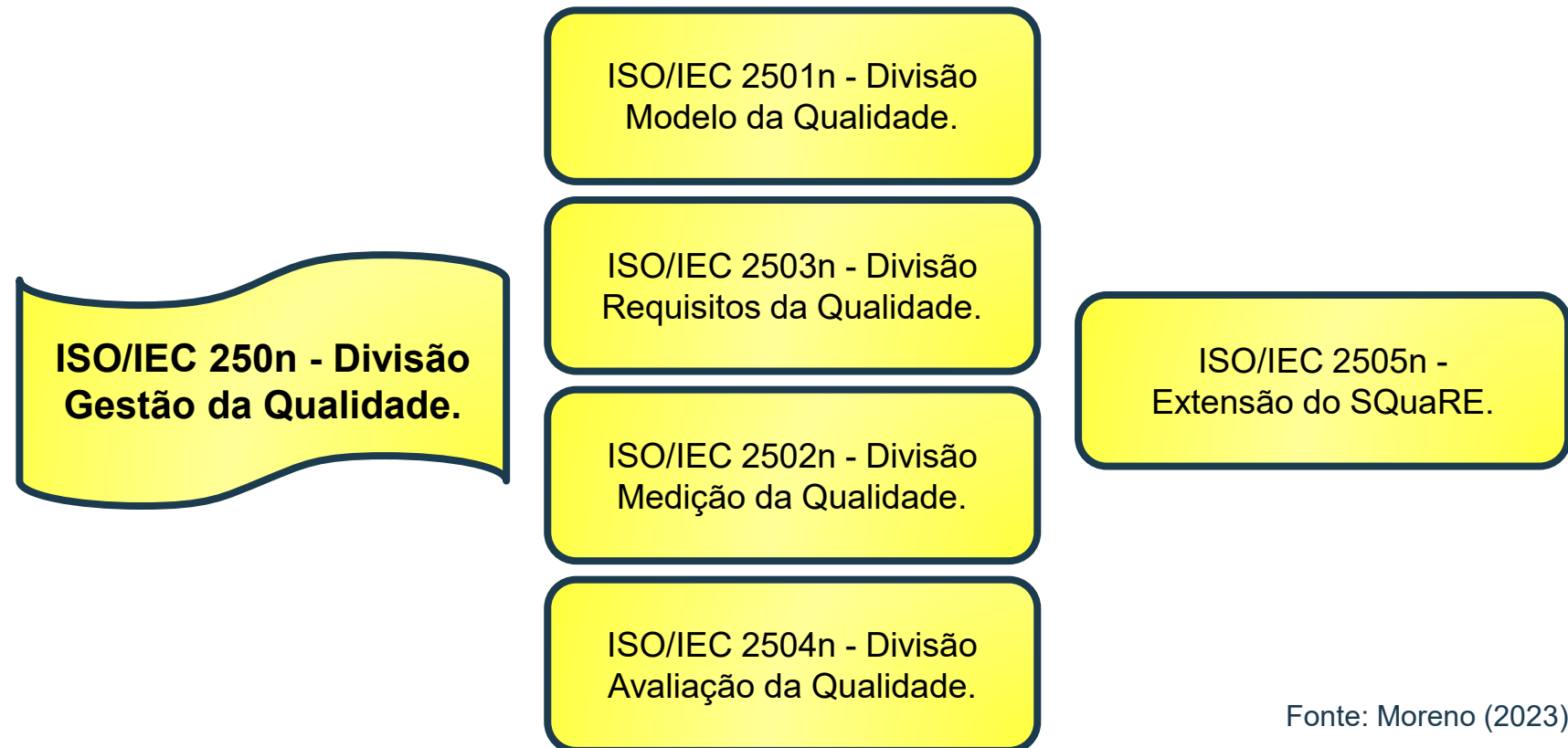
- Significado da sigla SQuaRE: Software Product Quality Requirements and Evaluation (Requisitos de Qualidade e Avaliação de Produtos de Software).
- A NBR ISO/IEC 25000 é uma evolução das séries de NBR ISO/IEC 9126 e NBR ISO/IEC14598.

A reorganização da NBR ISO/IEC 25000 está dividida em cinco tópicos, sendo que cada um trata de um assunto em particular, como mostra a figura abaixo:

Requisitos de Qualidade	Modelo de Qualidade	Avaliação
	Gerenciamento de qualidade	
	Medições	

NBR ISO/IEC 25000: estrutura

- A série de normas SQuaRE estabelece um conjunto de padrões e diretrizes para a avaliação da qualidade de software e produtos relacionados. Essa norma fornece uma estrutura que permite avaliar e medir a qualidade do software de forma objetiva e consistente.
- De acordo com a Softex (2013), o núcleo principal do SQuaRE, como mostra a figura, é composto de quatro divisões de normas e uma sequência prevista para extensão do modelo.



Interatividade

A NBR ISO/IEC 9126 descreve um modelo de qualidade do produto de software composto de: qualidade interna e externa e qualidade em uso. Assinale a alternativa correspondente aos atributos de qualidade referentes à qualidade em uso.

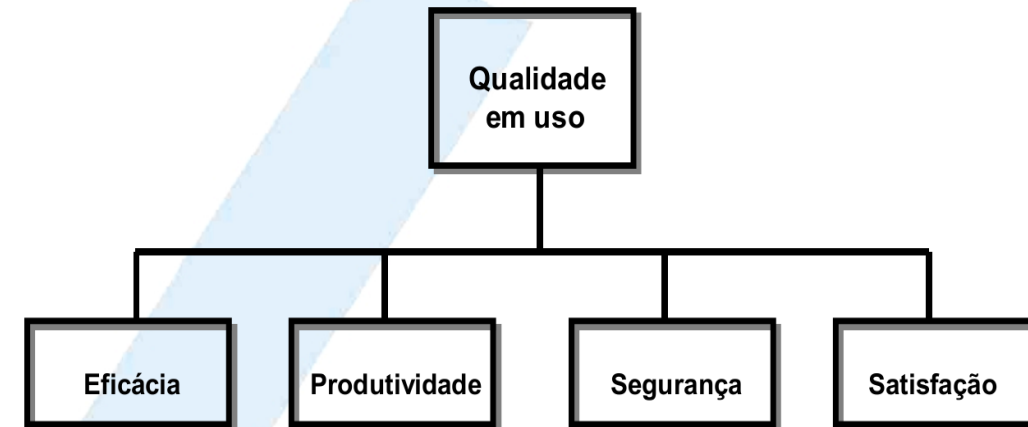
- a) Eficiência, portabilidade, eficácia e produtividade.
- b) Eficácia, produtividade, segurança e satisfação.
- c) Funcionalidade, confiabilidade, segurança e satisfação.
- d) Funcionalidade, confiabilidade, usabilidade e manutenibilidade.
- e) Usabilidade, manutenibilidade, eficácia e produtividade.

Resposta

A NBR ISO/IEC 9126 descreve um modelo de qualidade do produto de software composto de: qualidade interna e externa e qualidade em uso. Assinale a alternativa correspondente aos atributos de qualidade referentes à qualidade em uso.

- a) Eficiência, portabilidade, eficácia e produtividade.
- b) **Eficácia, produtividade, segurança e satisfação.**
- c) Funcionalidade, confiabilidade, segurança e satisfação.
- d) Funcionalidade, confiabilidade, usabilidade e manutenibilidade.
- e) Usabilidade, manutenibilidade, eficácia e produtividade.

- Requisitos de qualidade em uso: compõem a visão da qualidade do produto de software do ponto de vista do usuário.



Evolução, reengenharia, manutenção e tendências

“O que vem pela frente? Qual o caminho?” São essas e outras questões que serão abordadas nessa última seção da Engenharia de Software.

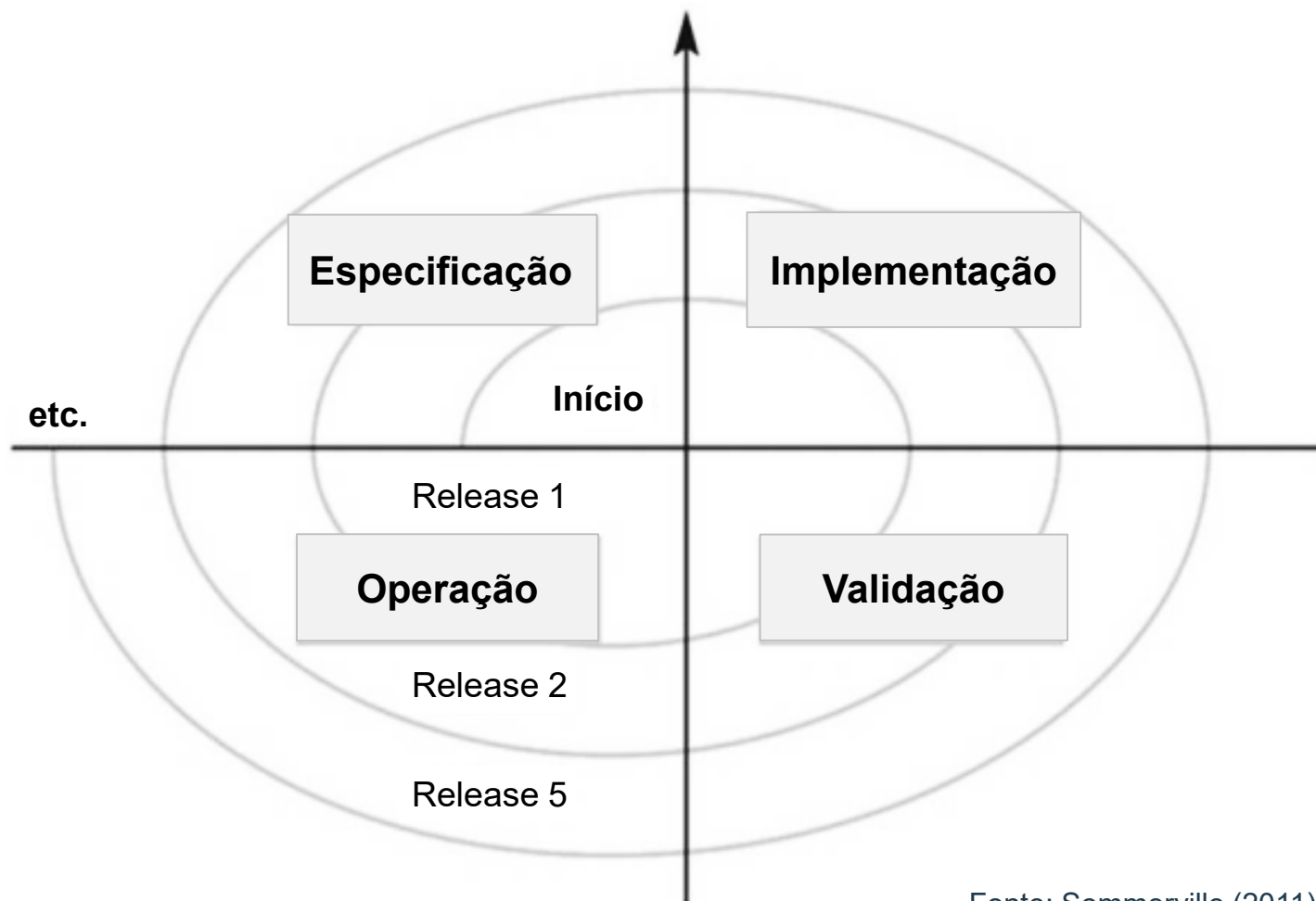
- A evolução refere-se às mudanças planejadas e não planejadas em um sistema de software ao longo do seu ciclo de vida, incluindo atualizações, correções de bugs e aprimoramentos.
- A reengenharia de software busca uma solução a partir do produto pronto, pode-se dizer que: a engenharia reversa inicia a partir da informação produzida e percorre o caminho inverso até os dados que a geraram. Depois de “descoberto” o software, este é reprojetoado com as mesmas funcionalidades e tecnologia atualizada.
 - A manutenção apresenta o processo contínuo de gerenciamento do software, para garantir que ele atenda às necessidades dos usuários e funcione sem problemas.
 - E em relação às tendências do software, vão ser abordadas a adoção de novas tecnologias em diversas áreas do conhecimento, a demanda por software que só cresce e a resposta da engenharia de software para sustentar a fábrica de software.

Evolução de software

- A evolução do software decorre basicamente: das necessidades de negócios em constante mudança e possíveis falhas que ocorrem em seu ciclo de vida devido a mudanças no ambiente operacional do sistema de software.
- A evolução do software é importante, pois as organizações investem grandes quantias de dinheiro em seu software e são totalmente dependentes desses sistemas. Seus sistemas são ativos críticos de negócios e as organizações devem investir nas mudanças de sistemas para manter o valor desses ativos (Sommerville, 2016).
 - A evolução do software é avaliada em conjunto com os componentes que integram o sistema de software.
 - Os componentes envolvem a lógica de processamento, implementação de novas funcionalidades, o hardware, o banco de dados, a rede de computadores e o impacto das mudanças no ambiente operacional.

Ciclo de vida de evolução do software

- No ciclo de vida de evolução do software, os requisitos dos sistemas de software instalados mudam à medida que o negócio e o ambiente os requerem; portanto, geram novos releases que incorporam as mudanças requeridas.
- De acordo com Sommerville (2016), a engenharia de software é um processo em espiral com requisitos, design, implementação e testes durante toda vida útil do sistema.



Reengenharia de software

O objetivo da reengenharia de software é o de redesenhar, reestruturar e otimizar sua arquitetura, código e funcionalidades.

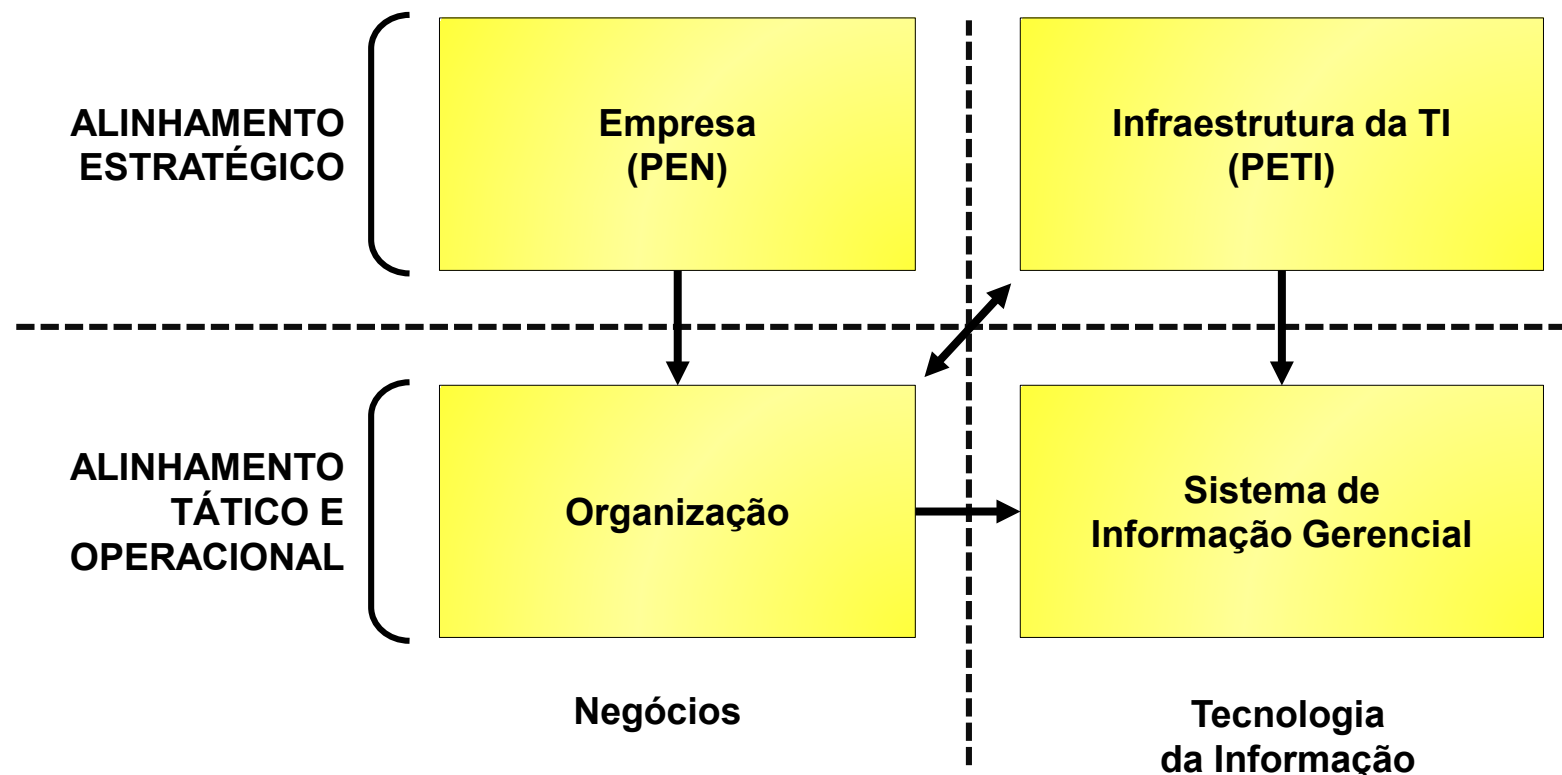


Software eficiente, escalável e adaptável às demandas atuais.

- Essa abordagem permite eliminar redundâncias, melhorar a qualidade e a segurança do código e agregar novas tecnologias, sem comprometer a funcionalidade original.
 - A reengenharia de software é uma poderosa ferramenta para revitalizar e prolongar a vida útil de sistemas legados, impulsionando a inovação e garantindo a competitividade das empresas em um ambiente tecnologicamente dinâmico.

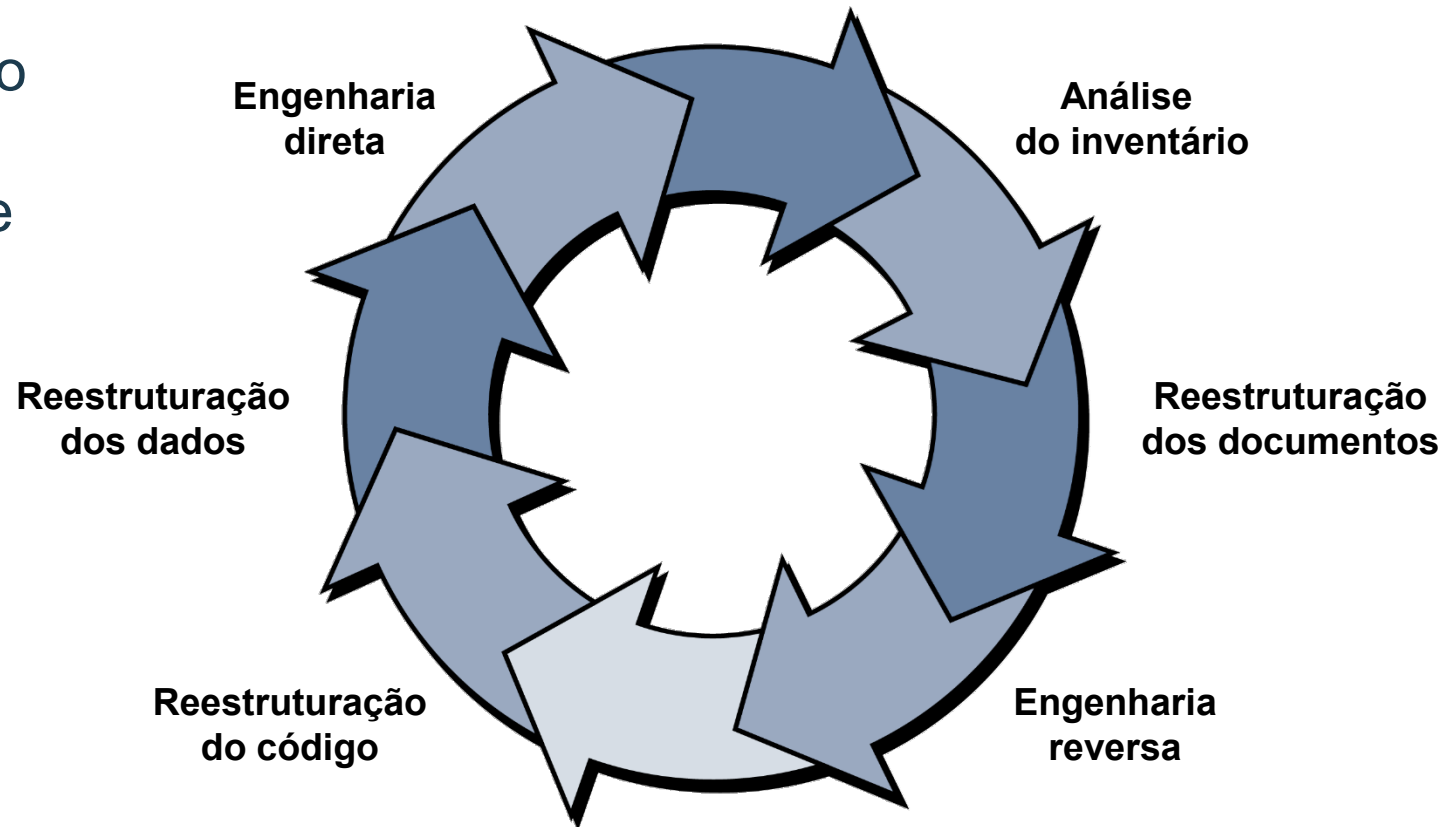
Reengenharia dos processos de negócio

- A Reengenharia dos Processos de Negócio (do inglês, Business Process Redesign – BPR) consiste no alinhamento do Planejamento Estratégico de Negócios (PEN) com o Planejamento Estratégico de TI (PETI).
- O diagrama ao lado mostra como funciona a integração funcional dos negócios com a Tecnologia da Informação.



Modelo de processo da reengenharia de software

- A engenharia reversa por diversas vezes é aplicada quando se parte da informação ou do conhecimento para os dados que podem gerá-la. Essa atividade é também bem utilizada no processo de manutenção do software.
- Em termos industriais, a engenharia reversa tem suas origens no mundo do hardware, para conhecer os segredos de projeto e fabricação do concorrente (Pressman, 2011).



Sistemas legados

Sistemas legados são sistemas antigos que ainda operam e precisam sempre estar ativos com bom desempenho.

- A reengenharia do software se faz útil nesse caso, porém alguns parâmetros devem ser avaliados antes da prática da reengenharia ou manutenção do software.
- Na reengenharia as organizações têm que decidir como obter o melhor retorno sobre o seu investimento.

Isso implica fazer uma avaliação realista dos seus sistemas legados para decidir sobre a melhor estratégia para a evolução desses sistemas. E de acordo com Sommerville (2016) existem quatro estratégias:

1. Descartar completamente o sistema.
2. Deixar o sistema inalterado e continuar com a manutenção regular.
3. Reestruturar o sistema para melhorar sua manutenibilidade.
4. Substituir todo ou parte do sistema por um novo.

Interatividade

Grande parte dos sistemas de software em utilização foi desenvolvida no passado, com tecnologia obsoleta. Esses sistemas ainda operam importantes funções corporativas e controlam grandes quantidades de eventos em uma grande massa de dados. Assinale a alternativa que não justifica uma reengenharia do sistema de software.

- a) As operações não contribuem para os negócios, porém seus dados são úteis para outros negócios e um novo sistema deve permitir o acesso a esses dados.
- b) As operações são complexas e estão com bom desempenho, de forma que o sistema deve ser mantido.
- c) O ambiente operacional está caótico, contudo dados, operações e estruturas de classes são de grande importância e o sistema deve ser substituído em parte ou por um novo.
 - d) É de importante contribuição para a empresa, pois o sistema pode ser reestruturado para melhorar sua manutenibilidade.
 - e) É de importante contribuição para a empresa, porém o sistema não pode ser alterado e deve continuar com manutenção corretiva e adaptativa.

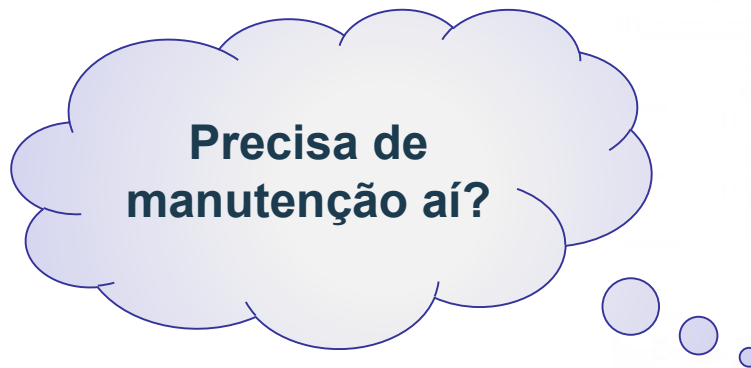
Resposta

Grande parte dos sistemas de software em utilização foi desenvolvida no passado, com tecnologia obsoleta. Esses sistemas ainda operam importantes funções corporativas e controlam grandes quantidades de eventos em uma grande massa de dados. Assinale a alternativa que não justifica uma reengenharia do sistema de software.

- a) As operações não contribuem para os negócios, porém seus dados são úteis para outros negócios e um novo sistema deve permitir o acesso a esses dados.
- b) As operações são complexas e estão com bom desempenho, de forma que o sistema deve ser mantido.
- c) O ambiente operacional está caótico, contudo dados, operações e estruturas de classes são de grande importância e o sistema deve ser substituído em parte ou por um novo.
 - d) É de importante contribuição para a empresa, pois o sistema pode ser reestruturado para melhorar sua manutenibilidade.
 - e) É de importante contribuição para a empresa, porém o sistema não pode ser alterado e deve continuar com manutenção corretiva e adaptativa.

Manutenção de software

- Das diversas peculiaridades da engenharia de software, diferente das demais engenharias, está o processo de manutenção do software.
- Seja na implantação de um sistema de software ou até na simples instalação de um novo aplicativo, a manutenção começa quase que em seguida.
- São problemas de configurações, adaptações ao ambiente operacional, interfaces não operacionais e, o maior deles, que são os novos requisitos de mudanças por parte do usuário.



SETUPGSM. IHC: Veja os cinco sinais de que está na hora de trocar o seu Computador.

Fonte: <https://x.gd/mA4AN>

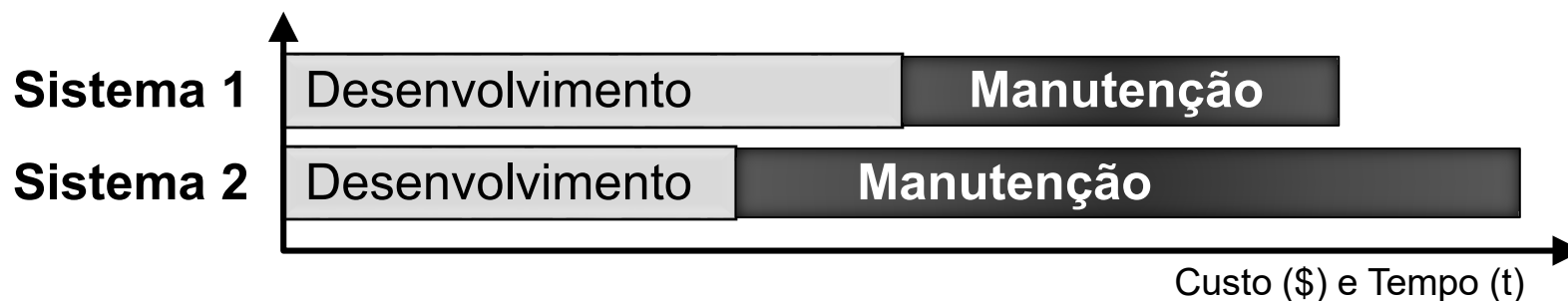
Procedimentos de manutenção

- O processo de mudança do software ocorre em todo o seu ciclo de vida, para manter o software operacional e adaptado a novas tecnologias.



Das práticas de manutenção, além de se ter o controle de versionamento e testes, a que mais se destaca são mudanças no código-fonte do software.

- Observe o comparativo entre desenvolvimento e manutenção na figura. O Sistema 1 com custo e tempo maiores no desenvolvimento, exigiu no período de manutenção um menor custo e menos tempo. O Sistema 2 foi desenvolvido mais rapidamente com custo e tempo inferiores ao do Sistema 1, porém exigiu no período de manutenção mais tempo e recursos. O resultado final é que o Sistema 1 obteve sobre o Sistema 2 um menor custo total e em menos tempo.



Tipos de manutenção

- A manutenção de software é o processo geral de mudança em um sistema depois que ele é liberado para uso. O termo geralmente se aplica ao software customizado em que grupos de desenvolvimento separados estão envolvidos antes e depois da liberação (Preesman, 2016).
- A manutenção pode ser corretiva, preventiva ou adaptativa, permitindo ajustes conforme as mudanças tecnológicas e demandas dos usuários.
 - Manutenção corretiva: Manutenção para reparar defeitos no software.
 - Manutenção preventiva: Envolve ações proativas para evitar possíveis problemas futuros no software.
 - Manutenção adaptativa: Para adaptar o software a um novo ambiente operacional, modificar ou fazer acréscimos à funcionalidade do sistema de software.

Manutenção corretiva: depuração de falhas (debug)

Depuração é a prática de verificação do código-fonte.

- A depuração é a atividade de rastreamento do código, com objetivo de corrigir e reduzir falhas no programa de computador.
- A atividade de verificação é iniciada pelo código e consiste em confrontar o requisito do software com o resultado obtido pelo software, para garantir que o que foi projetado foi construído de acordo.
- Na verificação do código fonte, faz-se a depuração de falhas (debug), que é a atividade de limpeza ou exclusão de partes indesejáveis.

Degugging (atividade de depuração de falhas)

1. Rastreamento do erro (ou bug): identificação da causa de um determinado problema.
2. Diagnóstico das causas do erro: avaliação sobre a gravidade, prioridade, riscos e impactos no comportamento do software.
3. Correção: implementação e testes para as correções.
4. Reflexão: medidas corretivas para a solução do problema.

Manutenção preventiva: análises, revisões e segurança no código



Fonte: ClipArt.

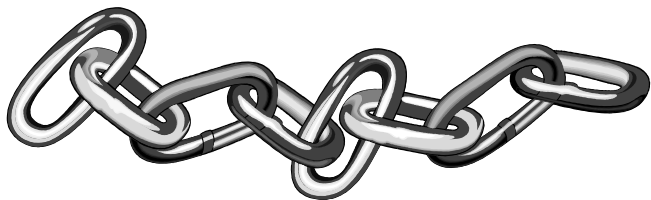
- Inclui-se aqui a manutenção preventiva por refatoração, que é a prática de reestruturar e melhorar o código-fonte sem alterar seu comportamento externo.
- É o processo de fazer ajustes no código para torná-lo mais legível, eficiente, organizado e sustentável, sem afetar a funcionalidade do software.

Caso: manutenção na computação em nuvem

- Na computação em nuvem (cloud computing) a dinâmica de atualizações de código é muito alta.
- Equipes de manutenção realizam verificações regulares no código-fonte para identificar áreas que possam apresentar vulnerabilidades de segurança, corrigindo-as antes que se tornem um problema.

Manutenção adaptativa: garantia de software em funcionamento

- As adaptações são feitas para garantir que o software continue funcionando e esteja ajustado às novas condições.
- A manutenção adaptativa faz alterações necessárias no software para mantê-lo operacional devido a mudanças no ambiente operacional, instalações de novos programas diferentes até do software em manutenção e mudanças de requisitos legais ou tecnológicas.



Fonte: ClipArt.

Caso: manutenção no compartilhamento de dados

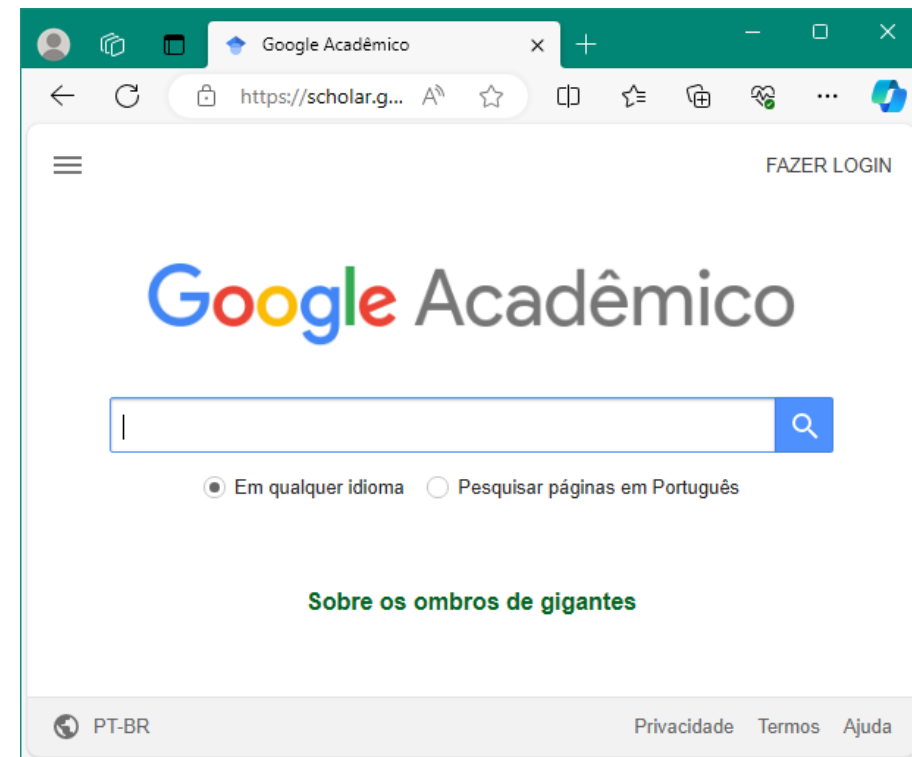
Constantemente supermercados atualizam seu sistema de software. Por exemplo: nova parceria com fornecedores.

- Nesse caso, é necessário abrir um link com a base de dados do fornecedor para o computador servidor do supermercado.
- Algumas regras devem ser embutidas no software do supermercado para interpretação e link com os dados do fornecedor.
- E do lado do fornecedor, é necessário estabelecer permissões de acesso aos dados.

Tendências do software

- Quando é introduzida uma tecnologia bem sucedida, o conceito inicial transforma-se em “ciclo de inovação”, razoavelmente previsível (Gaines, 1995 *apud* Pressman, 2011).
- Devido aos avanços tecnológicos e às mudanças nas demandas dos usuários e do mercado, as tendências do software na engenharia de software e em várias outras áreas estão em constante evolução.
- Aprimore seu conhecimento e trabalhos acadêmicos. Pesquise sobre as tendências da engenharia de software.

Fonte: GOOGLE®. Tela principal do Google Acadêmico. GOOGLE®. Disponível em: <https://scholar.google.com.br/schhp?hl=pt-BR>. Acesso em 24 jan. 2024.



Tendências do software: inteligência artificial e computação em nuvem

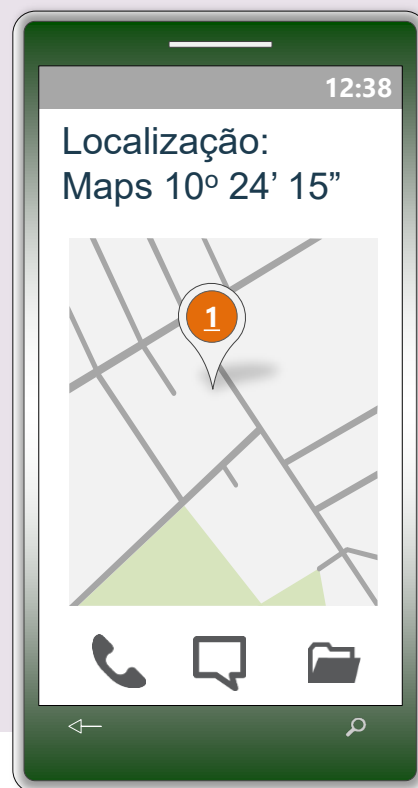
- Inteligência Artificial – IA (do inglês, Artificial Intelligence – AI): “Faz uso de algoritmos não numéricos para resolver problemas complexos que não sejam favoráveis à computação ou à análise direta” (Pressman, 2011).

IBM®. Estudo IBM: 41% das empresas no Brasil já implementaram ativamente Inteligência Artificial em seus negócios. IBM®, 29/09/2022. Disponível em: <https://www.ibm.com/blogs/ibm-comunica/estudo-ibm-41-das-empresas-no-brasil-ja-implementaram-ativamente-inteligencia-artificial-em-seus-negocios/>. Acesso em: 24 jan. 2024.

- Computação em Nuvem (Cloud Computing): serviços oferecidos reúnem diversos recursos de processamento, aplicações e armazenamento em servidores, que podem ser acessados de qualquer lugar do mundo pela internet.
- A computação em nuvem tem sido adotada por diversas empresas e alguns desses sistemas de software são disponibilizados ao público para uso geral. Acesse Google® Drive ou Outlook Drive da Microsoft®.

Tendências do software: aplicativos móveis e IoT

- Aplicações móveis: O uso crescente de dispositivos móveis impulsionou a criação de aplicativos móveis eficientes, intuitivos e altamente funcionais, abrangendo áreas como e-commerce, saúde, redes sociais e entretenimento.



- Internet das Coisas (IoT - Internet of Things): O controle e a interconexão pela internet de dispositivos inteligentes está impulsionando a demanda por software, de forma a coletar, analisar e atuar sobre dados gerados por esses dispositivos, em setores como casas inteligentes, manufatura e cidades inteligentes.



Fonte: https://digicomp.com.br/wp-content/uploads/2017/06/iot-internet_das_coisas-digicomp-engenharia_tecnologia.png. Acesso em: 24 jan. 2024.

Tendências do software: metodologia ágil / DevOps e interfaces avançadas

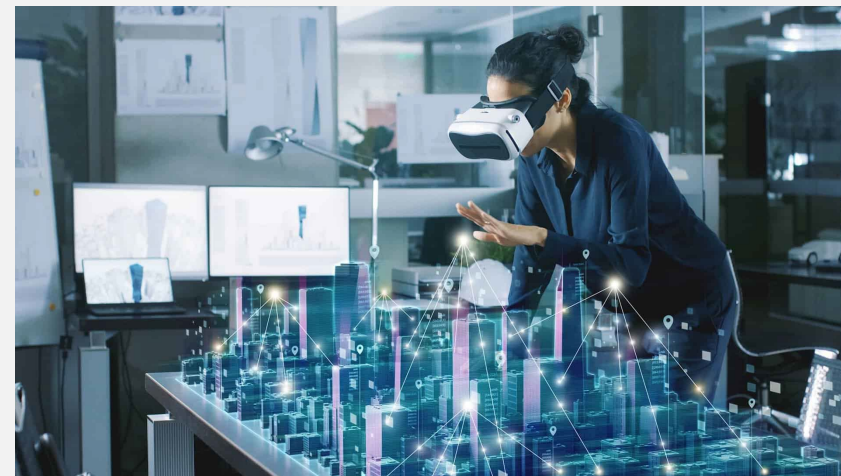
- Desenvolvimento ágil e DevOps: a abordagem ágil permite um maior iteração e colaboração, enquanto o DevOps integra desenvolvimento e operações para acelerar a entrega e melhorar a qualidade do software.

Fonte: IBM®. O que é DevOps? IBM®. Disponível em: <https://www.ibm.com/br-pt/topics/devops/>. Acesso em: 24 jan. 2024.

Interfaces de usuário avançadas: estão se tornando mais comuns, proporcionando interações mais imersivas e naturais. Interfaces de usuário como:

- Realidade Virtual
(do inglês, Virtual Reality - VR);
- Realidade Aumentada - RA
(do inglês, Augmented Reality - AR);
- Interfaces de Voz.

Fonte:
<https://i.pinimg.com/564x/b1/2f/76/b12f762904b95783d2c8c84812f39fde.jpg>. Acesso em: 24 jan. 2024.



Tendências do software: segurança, blockchain, sustentabilidade e análise

- Segurança cibernética: técnicas avançadas de criptografia, autenticação e proteção contra ameaças, a segurança cibernética é um campo crítico em evolução na proteção de sistemas.
- Blockchain: está sendo utilizado para garantir a segurança e a rastreabilidade de transações e processos.
- Sustentabilidade e ética: as preocupações ambientais e éticas estão moldando a forma como o software é desenvolvido, com uma ênfase crescente em práticas de programação responsável e sustentável.
 - Análise de dados: a capacidade de coletar e analisar dados em tempo real está ajudando as empresas a tomar decisões informadas e prever tendências futuras.

Interatividade

Na necessidade de se instalar um novo serviço de gerenciamento de dados em um determinado servidor, devido às suas características, o analista decidiu por instalar um novo sistema de backup. Assinale a alternativa correspondente ao tipo de manutenção que deve ser aplicado.

- a) A manutenção é adaptativa para um novo ambiente operacional em que serão necessárias novas funcionalidades.
- b) A manutenção é adaptativa para inserir um novo sistema de backup, porque o servidor já está configurado para isso.
- c) A manutenção é corretiva devido a problemas com o sistema de backup anterior.
- d) A manutenção é corretiva devido à obsolescência do ambiente operacional anterior.
- e) A manutenção é preventiva para evitar problemas proativos no software.

Resposta

Na necessidade de se instalar um novo serviço de gerenciamento de dados em um determinado servidor, devido às suas características, o analista decidiu por instalar um novo sistema de backup. Assinale a alternativa correspondente ao tipo de manutenção que deve ser aplicado.

- a) A manutenção é adaptativa para um novo ambiente operacional em que serão necessárias novas funcionalidades.
- b) A manutenção é adaptativa para inserir um novo sistema de backup, porque o servidor já está configurado para isso.
- c) A manutenção é corretiva devido a problemas com o sistema de backup anterior.
- d) A manutenção é corretiva devido à obsolescência do ambiente operacional anterior.
- e) A manutenção é preventiva para evitar problemas proativos no software.

Referências

- ABNT. *NBR ISO/IEC 9126-1 – Engenharia de software – Qualidade de produto*. Rio de Janeiro: Associação Brasileira de Normas Técnicas, 2003.
- ABNT. *NBR ISO/IEC 14598-1 – Tecnologia de informação – Avaliação de produto de software*. Rio de Janeiro: Associação Brasileira de Normas Técnicas, 2001.
- PRESSMAN, R. S. *Engenharia de software*. 7. ed. São Paulo: McGraw-Hill, 2011.
- SOMMERVILLE, Ian. *Software engineering*. 10. ed. USA: Pearson Education Limited, 2016.

ATÉ A PRÓXIMA!