

ENGENHARIA DE SOFTWARE - TEÓRICA

- 1.1) Explique por que software profissional não é apenas os programas que são desenvolvidos para clientes.
- 1.2) Qual é a diferença mais importante entre o desenvolvimento de um produto genérico de software sob demanda? O que isto pode significar na prática para usuários de produtos de softwares genéricos?
- 1.3) Quais são os quatro atributos importantes que todo software profissional deve possuir? Surgirão outros quatro atributos que, às vezes, podem ser significantes?
- 1.4) Além dos desafios de heterogeneidade, mudanças sociais e corporativas, confiança e proteção, identifique outros problemas e desafios que a engenharia de software provavelmente enfrentará no século XXI (referente ao Meio Ambiente).
- 1.5) Baseado em seu conhecimento de alguns tipos de aplicações discutidos na Seção 1.1.2. Explique, com exemplos, por que tipos de aplicações diferentes requerem técnicas especializadas de engenharia de software para apoiar seu projeto e desenvolvimento.
- 1.6) Explique por que existem ideias fundamentais na engenharia de software que se aplicam a todos os tipos de sistemas.
- 1.7) Explique como o uso universal da Internet mudou os sistemas de software.
- 1.8) Discuta se os engenheiros profissionais devem ser certificados da mesma forma que médicos e advogados.
- 1.9) Para cada uma das cláusulas no Código de Ética da ACM / IEE mostradas no Quadro 1.1. Sugira um exemplo adequado para ilustrar.
- 1.10) Para ajudar a combater o terrorismo, muitos países estão planejando desenvolver sistemas computacionais que rastreiam grandes números de cidadãos e suas ações. Obviamente, isso tem implicações em questões de privacidade. Discuta a ética de se trabalhar desenvolvendo esse tipo de sistema.

RESPOSTAS

1.1) O software profissional não se resume apenas ao código ou aos programas que são entregues ao cliente. Ele engloba um conjunto de atividades, processos e artefatos que garantem seu correto funcionamento, manutenção e evolução ao longo do tempo. Além do programa, há documentação técnica, manuais de usuário, planos de teste, suporte técnico, controle de versões, manutenção e atualizações contínuas. Dessa forma, o software é um produto muito mais completo e estruturado, que garante não só a entrega de funcionalidades, mas também sua confiabilidade, segurança e sustentabilidade operacional.

1.2) A principal diferença está no público-alvo. O software genérico é desenvolvido para atender a um mercado amplo, sendo vendido para vários usuários ou empresas, como é o caso de editores de texto ou sistemas de gestão empresarial. Já o software sob demanda é feito especificamente para atender às necessidades de um cliente, adaptando-se aos seus processos e demandas. Na prática, isso significa que usuários de software genérico muitas vezes precisam se adaptar ao funcionamento do software, enquanto quem adquire software sob demanda tem um produto personalizado, mais alinhado às suas atividades e objetivos.

1.3) Os atributos essenciais que o profissional deve possuir são:

- **Confiabilidade:** O sistema deve funcionar corretamente e de forma consistente.
- **Manutenibilidade:** Facilidade para realizar correções, melhorias e atualizações.
- **Eficiência:** Bom desempenho, otimizando o uso de recursos de memória e processamento.
- **Usabilidade:** Ser facilmente entendível e operacional por terceiros

1.4) Além dos desafios tradicionais, a engenharia de software enfrenta novos problemas e preocupações. A sustentabilidade ambiental se torna cada vez mais relevante, exigindo softwares mais eficientes energeticamente, que gerem menor impacto ambiental (Green IT). Outro desafio é lidar com a crescente integração de Inteligência Artificial, que traz questões éticas e de responsabilidade. A dependência crescente de software crítico, em áreas como saúde e transporte, torna necessário garantir níveis altíssimos de qualidade e confiabilidade. Além disso, promover acessibilidade e inclusão digital, considerando diferentes perfis de usuários, é uma responsabilidade cada vez maior para desenvolvedores e empresas.

1.5) Cada tipo de aplicação possui características e exigências específicas que demandam técnicas especializadas. Por exemplo, softwares embarcados, como os utilizados em automóveis e aeronaves, precisam ser altamente confiáveis, eficientes e seguros, pois uma falha pode colocar vidas em risco. Sistemas financeiros requerem foco extremo em segurança, integridade de dados e disponibilidade. Por outro lado, jogos digitais priorizam desempenho gráfico, resposta em tempo real e uma experiência de usuário envolvente. Portanto, o desenvolvimento de cada tipo de sistema requer metodologias, ferramentas e práticas adequadas às suas necessidades.

1.6) Apesar das diferenças entre os tipos de software, existem conceitos e práticas fundamentais que são comuns a todos eles. Todo desenvolvimento de software passa por etapas como levantamento de requisitos, projeto, implementação, testes, entrega e manutenção. Essas práticas ajudam a garantir que o software atenda às necessidades dos usuários, seja funcional, confiável e sustentável ao longo do tempo. Esses fundamentos formam a base da engenharia de software, independentemente da área de aplicação.

1.7) A popularização da internet transformou profundamente os sistemas de software. Antes, os softwares eram instalados localmente e utilizados de forma isolada. Hoje, muitos sistemas operam na nuvem, acessíveis de qualquer lugar, a qualquer hora, no modelo de Software como Serviço (SaaS). Além disso, os sistemas precisam ser projetados considerando segurança, escalabilidade, comunicação entre diferentes dispositivos e atualizações constantes. A internet também impulsionou o desenvolvimento colaborativo, com ferramentas integradas e compartilhamento de informações em tempo real.

1.8) Esse é um debate relevante na sociedade atual. Por um lado, sistemas de software estão cada vez mais envolvidos em atividades críticas, como na medicina, transportes e finanças, onde falhas podem gerar consequências graves. Isso sugere que a certificação poderia garantir maior responsabilidade e qualidade na atuação dos profissionais. Por outro lado, a tecnologia evolui rapidamente, e um processo de certificação rigoroso poderia criar barreiras à inovação e à entrada de novos profissionais no mercado. Uma solução intermediária seria exigir certificação obrigatória apenas para profissionais que trabalham em sistemas críticos, mantendo o desenvolvimento geral mais acessível.

1.9) O Código de Ética da ACM/IEEE orienta os profissionais de tecnologia a agirem de forma responsável, priorizando sempre o bem-estar da sociedade. Na prática, isso significa que o engenheiro deve desenvolver soluções que beneficiem as pessoas, como sistemas de alerta para desastres. Além disso, deve evitar causar danos, garantindo que dados sensíveis estejam protegidos. A honestidade também é essencial: o profissional deve relatar falhas, mesmo que isso impacte prazos. Outro ponto importante é agir de forma justa e sem discriminação, promovendo ambientes inclusivos. O respeito à privacidade e à confidencialidade dos dados é fundamental,

protegendo as informações dos usuários e garantindo que sejam usadas de maneira ética e legal.

“ Um engenheiro desenvolve um sistema de alerta para desastres naturais que informa a população sobre enchentes e deslizamentos, ajudando a salvar vidas. ”

O Estado de São Paulo implementou diversas soluções tecnológicas para alertar a população sobre desastres naturais, como enchentes, deslizamentos e tempestades. Essas iniciativas visam prevenir riscos e proteger vidas, utilizando diferentes canais de comunicação.

O **Alerta SP** é um aplicativo oficial da Defesa Civil do Estado de São Paulo, disponível para dispositivos iOS. Ele permite que os usuários cadastrem áreas de interesse para receber notificações sobre eventos climáticos e desastres naturais, como inundações, tempestades, deslizamentos, entre outros. Além dos alertas, o aplicativo fornece orientações sobre como agir antes, durante e depois desses eventos. O app é gratuito e foi desenvolvido pela Polícia Militar do Estado de São Paulo.

1.10) O desenvolvimento de sistemas de monitoramento para combate ao terrorismo e controle social levanta sérias questões éticas. Embora possam ser úteis para prevenir crimes e proteger a segurança pública, esses sistemas também ameaçam direitos fundamentais, como a privacidade e a liberdade individual. O engenheiro de software, ao participar desses projetos, deve refletir se o uso desse sistema está realmente alinhado com princípios éticos e se não será usado de forma abusiva por governos ou empresas. É necessário encontrar um equilíbrio entre segurança e privacidade, sempre buscando proteger os direitos da população.