1.1

O software profissional não se limita apenas aos programas desenvolvidos para clientes. Ele inclui todo o conjunto de atividades relacionadas ao desenvolvimento, manutenção e evolução de sistemas de software. Isso abrange desde a análise de requisitos até o suporte contínuo após a entrega.

1.2

Produtos genéricos são desenvolvidos para um amplo público, enquanto o software sob demanda é criado especificamente para atender às necessidades de um cliente ou organização.

A escolha entre software genérico e sob encomenda depende da necessidade. Se você busca uma solução rápida e econômica, o software genérico pode ser uma boa opção. Mas se você precisa de um software que atenda às suas demandas específicas e se integre com outros sistemas, o software sob encomenda é a melhor escolha.

1.3

Manutenibilidade: O software deve ser escrito de forma que possa evoluir para atender às necessidades dos clientes. Esse é um atributo crítico, porque a mudança de software é um requisito inevitável de um ambiente de negócio em mudança.

Confiança e proteção: A confiança do software inclui uma série de características como confiabilidade, proteção e segurança. Um software confiável não deve causar prejuízos físicos ou econômicos no caso de falha de sistema. Usuários maliciosos não devem ser capazes de acessar ou prejudicar o sistema.

Eficiência: O software não deve desperdiçar os recursos do sistema, como memória e ciclos do processador. Portanto, eficiência inclui capacidade de resposta, tempo de processamento, uso de memória etc.

Aceitabilidade: O software deve ser aceitável para o tipo de usuário para o qual foi projetado. Isso significa que deve ser compreensível, usável e compatível com outros sistemas usados por ele.

Outros significantes: segurança, escalabilidade, interoperabilidade e portabilidade.

1.4

* A rápida evolução das tecnologias exige que os profissionais de software se adaptem constantemente, aprendendo novas habilidades e ferramentas.
* A obsolescência tecnológica pode tornar softwares rapidamente desatualizados, impactando o investimento e a segurança dos sistemas.
* A integração de diferentes tecnologias e plataformas pode ser complexa e desafiadora.
* A necessidade de adaptar softwares a diferentes culturas e legislações pode aumentar a complexidade do desenvolvimento.

1.5

Diferentes tipos de aplicações demandam abordagens distintas de engenharia de software devido às suas características e requisitos específicos.

Exemplos:

* Aplicações stand-alone:
  + Exemplo: Um software de manipulação de fotos para um PC.
  + Requerimento: Essas aplicações são desenvolvidas para funcionar de forma autônoma em um computador local. Portanto, técnicas de otimização de desempenho e interfaces gráficas de usuário (GUI) eficientes são fundamentais. Além disso, é importante garantir que o software seja robusto o suficiente para lidar com diferentes

configurações de hardware.

* Aplicações interativas baseadas em transações:
  + Exemplo: Um sistema de comércio eletrônico.
  + Requerimento: Essas aplicações precisam lidar com múltiplas interações dos usuários, processar transações financeiras de forma segura e garantir alta disponibilidade. Portanto, técnicas de segurança da informação, gerenciamento de transações e escalabilidade são essenciais.
* Sistemas de controle embutidos:
  + Exemplo: Software de controle de antitravamento de freios em um carro.
  + Requerimento: Esses sistemas operam em dispositivos de hardware específicos e têm restrições de recursos e tempo real. Portanto, técnicas de programação de baixo nível, otimização de código, e verificação e validação rigorosas são necessárias para garantir o funcionamento correto e seguro do sistema.

1.6

Existem ideias fundamentais na engenharia de software que se aplicam a todos os tipos de sistemas. Isso inclui a necessidade de um processo bem gerenciado, garantia de confiança e desempenho, entendimento e gerenciamento de requisitos, e a promoção do reuso de recursos. Esses princípios são essenciais para garantir o desenvolvimento eficiente e de alta qualidade de qualquer software, independentemente de sua finalidade específica.

1.7

O uso universal da Internet impactou profundamente os sistemas de software, mudando sua natureza, processo de desenvolvimento e impacto social. Os sistemas de software agora são mais conectados, interoperáveis, escaláveis e seguros. O desenvolvimento de software se tornou mais ágil, distribuído e dependente de ferramentas online. A internet democratizou o acesso à informação, promoveu a inclusão digital e contribuiu para a economia global.

1.8

Alguns argumentam que a certificação pode garantir padrões de competência e ética na prática da engenharia, aumentando a confiança do público e a segurança dos projetos. Outros acreditam que a natureza multidisciplinar e variada da engenharia torna difícil estabelecer um único padrão de certificação que seja aplicável a todas as áreas. No entanto, acredito que os engenheiros profissionais devem sim ser certificados pois a certificação pode trazer benefícios em termos de qualidade e segurança.

1.9

Público: Um engenheiro de software que trabalha em uma empresa de redes sociais implementa medidas de segurança adicionais para proteger a privacidade dos usuários e evitar o uso indevido de seus dados pessoais.

Cliente e empregador: Um engenheiro de software que trabalha em uma empresa de consultoria de TI prioriza a entrega de soluções que atendam às necessidades do cliente dentro do orçamento acordado, ao mesmo tempo em que garante a conformidade com os padrões éticos e legais.

Produto: Uma equipe de desenvolvimento de software realiza testes rigorosos e revisões de código para garantir que um sistema bancário online seja seguro e confiável, protegendo assim as transações financeiras dos clientes contra fraudes e ataques cibernéticos.

Julgamento: Um engenheiro de software se recusa a participar de práticas antiéticas, como a manipulação de dados para distorcer resultados de testes de desempenho de um software, mantendo sua integridade profissional.

Gerenciamento: Um líder de equipe de desenvolvimento de software promove uma cultura de ética e responsabilidade, incentivando os membros da equipe a comunicarem preocupações éticas e respeitando suas decisões baseadas em princípios éticos.

Profissão: Um engenheiro de software participa ativamente de eventos da comunidade de tecnologia, compartilhando conhecimentos e boas práticas para promover a imagem positiva da engenharia de software e inspirar futuras gerações de profissionais.

Colegas: Um engenheiro de software oferece suporte e orientação a um colega que está enfrentando dificuldades técnicas, demonstrando solidariedade e colaboração dentro da equipe de trabalho.

Si próprio: Um profissional de engenharia de software reserva tempo regularmente para participar de cursos, workshops e conferências para atualizar suas habilidades e conhecimentos, garantindo assim sua contínua relevância e contribuição para a profissão.

1.10

Essa é uma decisão que deve ser tomada com cuidado, ponderando os argumentos a favor e contra. É essencial garantir o uso ético e responsável desses sistemas, com salvaguardas para proteger a privacidade e as liberdades individuais. Deve-se levar em consideração os seguintes pontos: Necessidade e proporcionalidade, Transparência e supervisão, e Proteção de dados.