INSTITUTO FEDERAL
PARANÁ

Ministério da Educação
Instituto Federal do Paraná

Atividade Avaliativa

Alunos:	Natan Vinícius Zaleski Cancian	28/04	**Conceito:**
Disciplina:	Engenharia de Software	**Turma:** TADS-1A	
Professor:	Nelson Bellincanta Filho		

1 - Pesquise e liste cinco exemplos de softwares para cada uma das categorias abaixo.

Dica:

- Tente incluir uma variedade de softwares amplamente utilizados e reconhecidos em cada categoria.
- Certifique-se de explicar brevemente a função ou aplicação principal de cada software identificado.

Software de Sistema:
a. MacOS - Sistema Operacional da Apple utilizado em computadores empresariais ou pessoais da família Mac.

b. Microsoft Windows - Sistema Operacional da Microsoft amplamente utilizado em computadores domésticos/empresariais , para propósitos diversos.

c. Ubuntu - Sistema Operacional de código aberto baseado em UNIX , sendo livre para ser utilizado, alterado e redistribuído, para diversas finalidades de uso, seja pessoal ou profissional.

d. NTFS - Sistema de arquivos padrão do Microsoft Windows, sendo responsável por organizar e gerenciar como os dados são escritos, armazenados e recuperados no disco rígido.

e. Oracle Solaris - Sistema Operacional proprietário, desenvolvido pela Oracle, conhecido pela sua robustez, escalabilidade e foco em segurança. É projetado para aplicações Oracle Database e Java, oferecendo compatibilidade e facilidade de uso.

Software de Aplicação:
a. Bling - O Bling é um sistema de gestão empresarial(ERP) online que centraliza e integra informações e processos de uma empresa, como gestão de vendas, estoque, finanças, entre outros, em uma única interface.

b. SAP - O SAP é um sistema de planejamento de recursos empresariais (ERP) desenvolvido pela empresa SAP SE que integra as principais funções de uma empresa em um único sistema como finanças, produção, RH, cadeia de suprimentos, vendas e procurement. O SAP ERP oferece funcionalidades para otimizar processos, aumentar a produtividade e fornecer insights em tempo real sobre a organização.

c. Google Chrome - O Google Chrome é navegador web (browser), da empresa Google LLC, disponível para Microsoft Windows, Mac OS X e Linux… é utilizado para navegar na internet, acessar sites e executar aplicações web.

d. Protheus - O Protheus é um sistema de gestão empresarial (ERP) da TOTVS, que oferece uma ampla gama de funcionalidades para empresas de diferentes tamanhos e setores.

e. Opera - O Opera é um navegador da web desenvolvido pela empresa Opera e disponibilizado para Microsoft Windows, Mac OS X e Linux…, sendo utilizado para navegar na internet, acessar sites e executar aplicações web.

Software de Engenharia/Científico:
a. AutoCAD - O AutoCAD é um software do tipo CAD, da empresa Autodesk. É utilizado principalmente para a elaboração de peças de desenho técnico em duas dimensões (2D) e para criação de modelos tridimensionais (3D).

b. SolidWorks - O SolidWorks é um software proprietário da empresa Dassault Systèmes S.A, para design de produtos e engenharia. É utilizado em indústrias como automotiva, aeroespacial e manufatura para criar modelos 3D, montagens e desenhos técnicos.

c. MATLAB - O MATLAB é um software proprietário da empresa MathWorks Inc., usado principalmente para cálculos numéricos, visualização de dados e desenvolvimento de algoritmos. Ele é bastante utilizado em áreas como engenharia, matemática, física, finanças e aprendizado de máquina.

d. Revit - O Revit é um software proprietário da empresa Autodesk, utilizado em arquitetura, engenharia e construção para projetar e gerenciar edificações de forma colaborativa e eficiente. Ele permite criar modelos 3D inteligentes, com informações detalhadas sobre os componentes da construção, como paredes, janelas, portas e sistemas estruturais.

e. SketchUp - O SketchUp é um software proprietário da empresa Trimble Navigation, utilizado em arquitetura, design de interiores, urbanismo, engenharia civil e até mesmo em design de produtos.

Software Embarcado:
a. Software de roteador: Gerencia a conexão de rede, distribuição de sinal Wi-Fi, firewall e segurança da rede doméstica ou empresarial.

b. Software de airbag: Processa dados de sensores e ativa os airbags em milissegundos quando detecta uma colisão.

c. Software de micro-ondas: Controla o tempo, potência e ciclos de aquecimento conforme os comandos do usuário.

d. Software de freio ABS: Utiliza sensores nas rodas que monitoram a velocidade de rotação de cada roda, para que em uma frenagem brusca seja evitado o travamento das 4 rodas e derrapagem.

e. Software de ECU: Gerencia várias funções do carro como injeção eletrônica, controle de tração, transmissão e economia de combustível.

Software para Linha de Produtos:
a. Microsoft Office - O Microsoft Office, é um pacote de aplicativos proprietários para escritório desenvolvido pela Microsoft para computadores que utilizam o sistema operacional Microsoft Windows, além de computadores Mac da Apple.

b. Libre Office - O LibreOffice é um pacote de aplicativos livres e de código aberto para escritório disponível para Windows, Unix, Solaris, Linux e macOS.

c. Google Workspace - O Google Workspace é um conjunto de ferramentas de produtividade e colaboração baseado na nuvem, todos os aplicativos são totalmente personalizáveis de forma independente.

d. Open Office - O OpenOffice é um conjunto de aplicativos livres para escritório disponível para Windows, Unix, Solaris, Linux e macOS, é mantido pela Oracle Corporation.

e. Only Office - O Only Office é um pacote de aplicativos de código aberto para escritório, disponível para Windows, Unix, Solaris, Linux e macOS.

Aplicações Web/Aplicativos Móveis:
a. Spotify: O Spotify é uma plataforma de streaming de música e podcasts, permitindo escutar milhões de músicas e episódios de podcasts sob demanda, possui versão web e mobile.

b. Google Maps: O Google Maps é um serviço de pesquisa e visualização de mapas e imagens de satélite da Terra gratuito, possui versão web e mobile.

c. Netflix : A Netflix é uma plataforma de streaming de filmes e séries, permitindo visualizar milhares de filmes e episódios de séries sob demanda, possui versão web e mobile.

d. Whatsapp: O WhatsApp é um aplicativo de mensagens instantâneas e chamadas de voz para smartphones. Além de mensagens de texto, os usuários podem enviar imagens, vídeos e arquivos PDF, além de fazer ligações grátis por meio de uma conexão com a internet, possui versão web e mobile.

e. Slack: O Slack é uma plataforma de comunicação e colaboração para equipes, fornecendo mensagens instantâneas, compartilhamento de arquivos e integração com outras ferramentas em tempo real, tudo via navegador.

Software de Inteligência Artificial:
a. TensorFlow: O TensorFlow é uma biblioteca de código aberto desenvolvida pelo Google para aprendizado de máquina e redes neurais profundas (deep learning). É utilizado em aplicações como reconhecimento de imagem, processamento de linguagem natural, e sistemas de recomendação.

b. ChatGPT: O ChatGPT é um modelo de linguagem natural baseado em aprendizado profundo, capaz de compreender e gerar texto humano, utilizado em chatbots, assistentes virtuais, geração de conteúdo, tradução automática, e mais.

c. H20.ai: O H20.ai é uma plataforma de código aberto para aprendizado de máquina e IA empresarial, utilizado em bancos, seguros e saúde para análise preditiva, modelagem de risco e personalização.

d. IBM Watson: O IBM Watson é uma plataforma de IA da IBM que oferece processamento de linguagem natural, análise de sentimentos, e aprendizado de máquina. É utilizado em setores como saúde, finanças e atendimento ao cliente para análise de dados e automação inteligente.

e. Microsoft Azure AI: O Microsoft Azure AI é um conjunto de serviços de IA na nuvem oferecido pela Microsoft, incluindo reconhecimento de fala, visão computacional e machine learning. É utilizado por empresas para criar soluções de IA escaláveis, como análise de dados e automação de processos.

2 - Neste exercício, você irá explorar e descrever o processo de software, bem como suas principais etapas.

Por favor, siga as instruções abaixo:

a) Definição do Processo de Software:
R: Segundo Ian Sommerville, no livro Engenharia de Software, “um processo de software é um conjunto de atividades e resultados associados que produzem um produto de software.", ou seja são atividades organizadas que visam todo a criação, desenvolvimento, teste e suporte do software, focando em cumprir a qualidade e os requisitos solicitados pelo cliente.

Explique o que é o processo de software e por que é importante no desenvolvimento de sistemas de software.

b) Especifique cada uma das quatro principais etapas do processo de software:

- Especificação de Software;
  R: Definição dos requisitos do software, incluindo funcionalidades, características e restrições, junto ao cliente.

- Projeto e Implementação de Software;
  R: Definição de tecnologia a ser utilizada, codificação e construção dos componentes do software segundo os requisitos estabelecidos.

- Validação de Software;
  R: Teste do software desenvolvido para assegurar que foram cumpridos os requisitos iniciais bem como a qualidade do software, antes de disponibilizar ao cliente.

- Evolução de Software.
  R: Manutenção do software ao longo do tempo, para correção de erros, atualização ou mudança no software, seja para adicionar novas funcionalidades ou em razão de adaptar o produto às novas regras de negócio.

Para cada etapa, forneça uma breve descrição do que ela envolve e seu propósito dentro do processo de desenvolvimento de software.

c) Exemplos ou Cenários:
R:
Cenário: Uma faculdade deseja um sistema para registro de presença dos alunos.

Especificação de Software
A equipe de desenvolvimento se reúne com os reitores da faculdade para entender o que o sistema deve fazer: registrar a presença por aula, gerar relatórios por turma, e permitir acesso por professores.Uma vez entendida a regra de negócio, o documento de requisitos é criado com todas as funcionalidades esperadas.

Implementação de Software
A equipe de desenvolvimento define a arquitetura do projeto, tecnologias empregadas, modelos utilizados e etc. Os desenvolvedores criam a interface do sistema, definem o banco de dados com as turmas e alunos, e escrevem o código que registra a presença. O sistema começa a tomar forma com base no que foi especificado.

Validação de Software
Com o sistema desenvolvido é necessário realizar testes antes de entregar o software à faculdade. A equipe de testes verifica se os professores conseguem registrar presença corretamente e se os relatórios estão corretos, testam o que acontece quando há falhas, como conexão ruim ou erros de digitação. Erros são reportados à equipe de desenvolvimento e as correções são feitas para garantir que tudo funcione conforme esperado.

Evolução de Software
Após alguns meses de uso, a faculdade quer uma nova função para que os alunos possam ver seu registro de presença online. A equipe de desenvolvimento retorna ao projeto e adiciona uma área no sistema onde os alunos podem verificar suas informações.

Dê exemplos ou cenários hipotéticos que ilustrem cada etapa do processo de software. Isso pode ajudar a entender melhor como cada etapa funciona na prática.

d) Importância das Etapas:
R: A especificação de software é a base do projeto, se os requisitos não forem bem documentados e esclarecidos, o sistema será desenvolvido com funcionalidades

erradas ou incompletas. Na implementação de software, é onde o sistema ganha corpo e toma forma, um bom desenvolvimento assegura que o software seja funcional, eficiente e bem estruturado, dentro dos requisitos e tecnologias definidas. Na validação do software é onde o desenvolvimento apresentado é colocado à prova, se garante a qualidade do software, se funciona corretamente antes de ser entregue, um bom teste de software evita que erros passem despercebidos e prejudiquem os usuários. A evolução do software vai garantir que o software se mantenha útil, corrigindo falhas, se adaptando a tecnologias novas, e atendendo a mudanças nos requisitos, ou criação de novos.

Discuta a importância de cada etapa do processo de software e como ela contribui para o sucesso geral do desenvolvimento de software. Considere como cada etapa ajuda a garantir a qualidade, eficiência e adequação do sistema de software.

e) Desafios e Soluções:
R: Na fase de especificação de software os principais desafios apresentados são a dificuldade em entender exatamente o que o cliente quer, requisitos mal definidos, incompletos ou que mudam com frequência, as principais soluções para estes problemas são realizar entrevistas detalhadas com os envolvidos, utilizar-se protótipos e desenhos variados para validar ideias, aplicar técnicas como histórias de usuário ou casos de uso para capturar requisitos com clareza, e gravar todas as reuniões e discussões formais sobre o projeto, descartando a informalidade nesta etapa. Já na etapa de implementação de software os principais desafios apresentados serão mais técnicos, na parte do desenvolvimento em si, como código mal-escrito gerando re-trabalho, falhas de comunicação entre a equipe gerando perda de tempo e etc, ou o uso de uma tecnologia em detrimento de outra que pode atrasar o processo de desenvolvimento, para estes problemas as soluções são adotar boas práticas de programação e utilizar padrões de projeto, usar alguma ferramenta de versionamento de código, revisar o código e manter uma documentação técnica clara e metodologias ágeis para facilitar a colaboração, além de uma comunicação precisa e oportuna entre a equipe. Na etapa da validação do software os principais desafios encontrados podem ser a falta de testes suficientes para cobrir todos os cenários, sejam por limitações diversas, e a descoberta de erros tarde demais no processo, para solucionar esses desafios é necessário disponibilizar o tempo necessário para criar um plano de testes desde o início do projeto, utilizar testes automatizados e testes manuais, e realizar testes de usabilidade com usuários reais sempre que possível. Por último, na etapa de evolução de software os principais desafios encontrados seriam a dificuldade para modificar o sistema sem causar novos erros, os ajustes provisórios, que deixam mais difícil a modificação e atualização de certos componentes do sistema, para solucionar esses desafios é necessário aplicar princípios de código limpo e modularidade desde o início do projeto,manter testes automatizados atualizados para evitar regressões, registrar e priorizar mudanças por meio de uma gestão de backlog eficiente, desta forma evita-se o “código desconhecido”.

Analise os desafios comuns enfrentados em cada etapa do processo de software e sugira possíveis soluções para superá-los. Isso pode incluir problemas de

comunicação, mudanças nos requisitos ou dificuldades técnicas, entre outros.

3 - Neste exercício, você irá investigar e compreender os modelos de processo de software, que são estruturas fundamentais para organizar e gerenciar o desenvolvimento de software.

Instruções:

a) Definição dos Modelos de Processo de Software:
R: Segundo Ian Sommerville, um modelo de processo de software é uma representação simplificada e abstrata de um processo de desenvolvimento de software, que descreve as principais atividades envolvidas e a ordem em que elas devem ser executadas. Esses modelos são essenciais porque fornecem uma estrutura organizada para planejar e controlar o desenvolvimento, permitindo melhor gerenciamento de tempo, recursos e qualidade. Sommerville destaca que diferentes modelos — como o cascata, incremental, integração contínua e ágil — são mais adequados para diferentes tipos de projetos, e a escolha correta do modelo pode aumentar significativamente as chances de sucesso no desenvolvimento do software.

Em um parágrafo, explique o que são modelos de processo de software e por que são essenciais no contexto do desenvolvimento de software.

b) Identificação de Exemplos:
R:
1. Modelo em Cascata (Waterfall) — Uma abordagem tradicional

Principais características:
Processo sequencial, passa por etapas bem definidas: requisitos, projeto, implementação, testes e manutenção. Cada fase precisa ser concluída totalmente antes de começar a próxima, a documentação é detalhada: há uma forte ênfase na documentação formal em cada etapa.
Como isso influencia o ciclo de vida do projeto:
É uma ótima opção para projetos bem estruturados e com requisitos claros desde o começo, pois facilita o gerenciamento, por outro lado, é mais difícil fazer mudanças nos requisitos ao longo do caminho, devido à sua estrutura rígida. Portanto, funciona melhor quando os requisitos estão estáveis e bem compreendidos desde o início.

2. Modelo Incremental

Principais características:
O desenvolvimento acontece em etapas, chamadas de incrementos, a cada etapa uma nova funcionalidade é entregue,essa abordagem combina elementos do modelo cascata com entregas mais frequentes e parciais, após cada incremento, o cliente dá seu feedback, ajudando a direcionar os próximos passos.
Como isso influencia o ciclo de vida do projeto:
É possível validar partes do sistema com os usuários antes de completar tudo, o que facilita ajustes ao longo do caminho,mudanças podem ser feitas com mais facilidade durante o processo, o risco é menor em comparação com o modelo cascata já que

problemas podem ser identificados e corrigidos nos incrementos seguintes.

3. Desenvolvimento Ágil com Scrum — Modelo Ágil

Principais características:
Trabalha com ciclos de desenvolvimento curtos, chamados sprints, garante a entrega contínua de softwares que realmente funcionam, valoriza bastante a colaboração com o cliente e a capacidade de se adaptar às mudanças que surgem, inclui testes e integrações contínuas, para manter tudo funcionando direitinho.
Como isso influencia o ciclo de vida do projeto:
É um método flexível, que consegue se ajustar a mudanças nos requisitos, mesmo nas fases mais avançadas,foca na entrega constante de valor e no recebimento de feedbacks frequentes, para dar certo, exige um bom envolvimento do cliente ao longo de todo o processo.

Liste e descreva brevemente três exemplos de modelos de processo de software distintos. Certifique-se de incluir pelo menos um modelo tradicional e um modelo ágil.

Para cada modelo identificado, destaque suas características principais e como elas influenciam o ciclo de vida do desenvolvimento de software.

c) Comparação entre Modelos:
R:
1. Modelo Cascata (Waterfall)

Vantagens:
O modelo é linear e fácil de entender, com etapas bem definidas e sequenciais,há uma documentação detalhada em cada fase, o que facilita a rastreabilidade e o controle do projeto,como o progresso é feito de forma sequencial, ele pode ser bem monitorado e controlado.

Desvantagens:
O modelo é rígido e inflexível, é difícil de voltar atrás e ajustar fases anteriores após a conclusão, mudanças nos requisitos durante o desenvolvimento são caras e complicadas, a entrega de um produto funcional só ocorre no final, o que significa que o feedback do cliente só pode ser obtido tardiamente, não é ideal para projetos com requisitos mal definidos ou que podem mudar ao longo do tempo.

Aplicação:
Projetos com requisitos bem definidos e estáveis, sistemas críticos, como software para indústrias de defesa ou aeroespaciais(safety e security), onde mudanças no projeto podem ser caras e arriscadas, sistemas regulamentados onde a documentação e o controle rigoroso são essenciais.

2. Modelo Incremental

Vantagens:

A entrega de funcionalidades é mais rápida, cada incremento entrega uma parte funcional do sistema, o que permite que os usuários testem e forneçam feedback mais cedo, mudanças podem ser feitas em incrementos futuros, permitindo que o o sistema evolua com base nas necessidades reais, como o sistema é desenvolvido em partes, falhas podem ser detectadas e corrigidas mais cedo, evitando grandes erros no final do projeto.

Desvantagens:

Exige um planejamento complexo, embora as entregas parciais sejam vantajosas, o planejamento de múltiplos incrementos pode ser desafiador, especialmente em grandes sistemas, a integração contínua necessária requer uma boa coordenação e testes constantes para garantir que os incrementos se integrem bem no sistema global, falta de coesão no design como cada incremento pode ser desenvolvido independentemente, pode haver problemas de consistência entre diferentes partes do sistema.

Aplicação:

Projetos com requisitos parciais ou incrementais, quando o cliente não precisa de um sistema completo de imediato, mas valoriza entregas regulares, sistemas que podem evoluir ao longo do tempo, como softwares de produtos ou aplicações web, onde novas funcionalidades podem ser lançadas em partes, projetos com incertezas moderadas, onde mudanças podem ser esperadas, mas dentro de um escopo gerenciável.

3. Modelo Ágil: Scrum

Vantagens:

O modelo ágil é altamente adaptável a mudanças rápidas de requisitos, o que é ideal para ambientes dinâmicos, entrega contínua de valor, foco em ciclos curtos (sprints) e entregas regulares permite que o cliente veja resultados tangíveis rapidamente e faça ajustes conforme necessário, o trabalho em equipe é enfatizado, assim como a colaboração constante com o cliente, promovendo um ambiente de feedback contínuo, como o desenvolvimento é guiado por feedback frequente, o produto final tende a estar mais alinhado com as expectativas do cliente.

Desvantagens:

Em algumas abordagens ágeis, a documentação pode ser minimizada, o que pode ser um problema em ambientes que exigem rastreabilidade rigorosa, embora funcione bem para pequenas equipes, escalar essas práticas para grandes projetos pode ser desafiador, especialmente sem uma boa governança, exige maturidade da equipe, times ágeis precisam ser autônomos, colaborativos e bem treinados, em equipes inexperientes, pode haver dificuldades em cumprir os ciclos de sprint dentro dos prazos e entregar valor consistentemente.

Aplicação:

Projetos com requisitos dinâmicos e em constante mudança, como em startups

ou produtos inovadores, desenvolvimento de software com foco em valor imediato para o cliente, como no desenvolvimento de aplicativos móveis, sistemas web ou novos produtos de software.

Compare e contraste os modelos de processo de software identificados.

Discuta suas vantagens, desvantagens e contextos de aplicação adequados.

d) Aplicação Prática:
R:

Cenário 01: Sistema de software para uma instituição de controle de tráfego aéreo, onde os requisitos são bem definidos e a arquitetura precisa ser robusta e segura.

O modelo Cascata seria ideal para este cenário, a natureza linear e sequencial do modelo permite que o sistema seja desenvolvido de maneira estruturada, o que é necessário para sistemas de alta complexidade e criticidade, como os de uma instituição de controle de tráfego aéreo. O sistema possui requisitos claros e documentados, e mudanças no escopo são improváveis após o início do desenvolvimento, a abordagem formal e a documentação detalhada são essenciais para garantir que cada fase do desenvolvimento esteja correta antes de seguir para a próxima, o modelo cascata funciona bem com equipes grandes e bem estruturadas, onde diferentes sub equipes podem se concentrar em diferentes fases (como design, implementação, testes, etc.).

Cenário 02: Uma startup desenvolvendo um MVP (Produto Mínimo Viável) para testar uma ideia de mercado, onde os requisitos podem mudar com frequência.

O modelo ágil Scrum, seria a escolha mais eficaz para este projeto, pois o foco em ciclos curtos de desenvolvimento (sprints) e feedback constante se alinha perfeitamente com a necessidade de flexibilidade e evolução rápida de requisitos, startups frequentemente experimentam mudanças rápidas nos requisitos à medida que testam o mercado e o feedback dos usuários, o que é ideal para um MVP, o modelo ágil funciona bem com equipes pequenas e dinâmicas,
que podem colaborar de forma intensiva e adaptativa.

Cenário 03: Software de gestão empresarial(ERP) para uma empresa de médio porte, onde os requisitos são bem definidos, mas algumas mudanças podem ocorrer à medida que o projeto avança.

O modelo Incremental seria uma boa escolha para este tipo de projeto, pois permite o desenvolvimento de módulos ou funcionalidades de forma incremental e iterativa, começando com um conjunto básico de funcionalidades, mas à medida que o projeto avança, novos requisitos podem surgir ou ajustes podem ser feitos, permitindo que diferentes partes do sistema sejam desenvolvidas em ciclos curtos e permitindo uma validação contínua do sistema o que ajuda a mitigar riscos de falhas graves no final

do projeto.

Considere diferentes cenários de desenvolvimento de software e discuta como cada modelo de processo de software poderia ser aplicado de maneira eficaz em tais situações.

Leve em consideração fatores como tamanho da equipe, complexidade do projeto e requisitos de flexibilidade.

e) Reflexão:
R:
A escolha do modelo de processo deve ser cuidadosamente ajustada ao contexto e às necessidades do projeto para otimizar o ciclo de desenvolvimento e maximizar o sucesso do software , logo o a escolha do modelo certo ajuda a entregar o software no prazo e dentro do orçamento, considerando o perfil do projeto, da equipe e as necessidades do cliente. Modelos ágeis, e Incremental oferecem flexibilidade e adaptabilidade, enquanto o modelo Cascata é adequado para projetos com requisitos bem definidos e estabilidade, ter isto em mente é fundamental para poder mensurar a produtividade da equipe, cumprir os prazos estabelecidas e a expectativa do cliente quanto à solução a ser entregue.

Finalize o exercício refletindo sobre a importância de selecionar o modelo de processo de software adequado para um projeto específico.

Considere como a escolha do modelo certo pode impactar o sucesso do projeto, a eficiência da equipe e a satisfação do cliente.

f) Exemplos de Modelos de Processo de Software:

Modelo Cascata (Waterfall)

Descrição: O modelo Cascata é um processo sequencial de desenvolvimento em que cada fase do projeto(requisitos, design, implementação, testes, manutenção) deve ser concluída antes de iniciar a próxima. Não há espaço para revisões após o início de uma fase.
Exemplo: Software para sistemas críticos e altamente regulamentados, como em indústrias de defesa, aeroespaciais.

Modelo incremental

Descrição: O modelo Incremental divide o projeto em pequenas partes, incrementos que são desenvolvidos e entregues em ciclos curtos. Cada incremento adiciona funcionalidades ao sistema já em uso, permitindo ajustes baseados no feedback.
Exemplo: Software de gestão empresarial (ERP), onde o sistema é liberado por módulos, e cada módulo pode ser testado e ajustado de acordo com a necessidade do cliente.

É perceptível que o modelo Cascata é ideal para para projetos com requisitos fixos e

bem definidos, o que dificulta ajustes durante desenvolvimento e visualização do software, já o modelo incremental é mais flexível, com entregas regulares de funcionalidades, permitindo ajustes e feedback contínuos. Logo, podemos considerar que para grandes sistemas, que não podem falhar, que devem assegurar safety e security, como exemplo um sistema de controle de tráfego aéreo, ou sistemas embarcados , é mais vantajoso optar pelo modelo Cascata, para sistemas menores, que mudam constantemente e necessitam se atualizar para continuar relevante, como sistemas para e-commerce, ou sistema para pdv, é mais interessante optar pelo modelo Incremental. Portanto, a escolha do modelo de processo de software pode ser decisiva no sucesso de um projeto, um exemplo positivo é o uso do modelo ágil no desenvolvimento do Spotify, onde entregas rápidas e feedback contínuo permitiram adaptações ágeis ao mercado, resultando em grande sucesso, por outro lado, um exemplo negativo ocorreu no projeto SPDRF da Receita Federal, que utilizou o modelo Cascata, a inflexibilidade para lidar com mudanças nos requisitos causou atrasos e custos elevados, resultando em um sistema desatualizado e mal integrado, gerando insatisfação entre os usuários. A escolha correta do modelo é crucial para o alinhamento entre complexidade, exigências do cliente e capacidade de adaptação.

Pesquise cada modelo de processo de software e forneça uma descrição clara e concisa para cada um.

Ao comparar os modelos, destaque as diferenças significativas em suas abordagens e como elas podem afetar o desenvolvimento de software.

Considere os exemplos fornecidos e pense em outras situações nas quais esses modelos poderiam ser aplicados.

Ao refletir sobre a importância da seleção do modelo adequado, pense em casos práticos nos quais a escolha do modelo de processo de software teve um impacto positivo ou negativo em um projeto de desenvolvimento de software.