Texto)

O sistema deve exibir imediatamente o nome da empresa "Confeitaria Imperia" ao cliente ao entrar no site.

O sistema deve apresentar uma interface intuitiva onde o cliente pode inserir o valor desejado para o orçamento.

O sistema deve aceitar valores numéricos como representação dos valores dos bolos que o cliente está disposto a gastar.

O sistema deve calcular e apresentar bolos disponíveis com base no valor fornecido pelo cliente.

Os bolos apresentados devem ser os mais próximos possível do valor inserido pelo cliente.

Valores dos bolos:

1Kg = R$300

3Kg = R$700

5Kg = R$1500

O sistema deve calcular com precisão o valor restante após a escolha do bolo.

Se o cliente não possui o dinheiro suficiente para o bolo de R$300, o sistema deve proceder com o banimento do cliente.

Após a apresentação dos bolos, o sistema deve permitir ao cliente adicionar modificações à escolha.

A cada R$100 que ultrapassar o valor inicial, o sistema deve habilitar o cliente a escolher entre modificações adicionais.

O sistema deve listar claramente as modificações disponíveis.

Caso o cliente tenha valor restante o cliente pode solicitar qualquer modificação ao bolo, e a Confeitaria Imperia deve fazer o máximo possível para realizar essas solicitações.

Caso o cliente nao queria

Acessibilidade e Compatibilidade:

O sistema deve ser acessível através de diferentes dispositivos e navegadores.

Deve garantir compatibilidade com uma variedade de sistemas operacionais.

Usabilidade e Confidencialidade:

O sistema deve ser intuitivo e de fácil navegação.

As informações do cliente devem ser tratadas com confidencialidade.

1) Uma abordagem flexível e adaptativa para o desenvolvimento de software.

As metodologias ágeis são conhecidas por serem uma abordagem flexível e adaptativa para o desenvolvimento de software. Elas promovem a entrega incremental, a colaboração constante com os clientes e a capacidade de responder a mudanças nos requisitos do projeto de maneira eficiente.

As demais opções estão incorretas porque:

Um modelo estático para lidar com mudanças no escopo do projeto é oposto à flexibilidade das metodologias ágeis.

Um método tradicional de gerenciamento de projetos geralmente implica em abordagens mais sequenciais, o oposto da natureza iterativa das metodologias ágeis.

Um conjunto de práticas rígidas e inflexíveis está em oposição à filosofia ágil, que favorece a adaptabilidade.

A afirmação de que as metodologias ágeis são uma técnica específica para aumentar a complexidade nos processos é incorreta; pelo contrário, elas visam simplificar o desenvolvimento e melhorar a eficiência.

2)

Responder às mudanças do cliente ao longo do desenvolvimento.

R:Correto: Este é o principal valor das metodologias ágeis. Elas se baseiam na adaptabilidade e flexibilidade para atender às mudanças contínuas nas necessidades do cliente.

Minimizar a interação entre os membros da equipe e entregar de uma maneira ágil.

R:Incorreto: A interação entre os membros da equipe é uma parte crucial das metodologias ágeis, promovendo colaboração e comunicação constante.

Entregar software conforme o planejado, sem alterações.

R:Incorreto: As metodologias ágeis valorizam a capacidade de resposta a mudanças, buscando a entrega incremental e adaptativa. O desenvolvimento rígido conforme o planejado vai contra esses princípios.

Priorizar a documentação extensiva sobre o produto.

R:Incorreto: Embora a documentação seja importante, as metodologias ágeis priorizam mais a comunicação cara a cara e o software funcionando do que a documentação extensiva.

Seguir um plano detalhado desde o início do projeto.

R:Incorreto: As metodologias ágeis promovem a flexibilidade e se ajustam às mudanças, o que contrasta com a ideia de seguir um plano detalhado desde o início.

3)

A criação de um plano de marketing para promover o produto após o lançamento.

1.Incorreto: A criação de um plano de marketing está relacionada à estratégia de comercialização e promoção do produto, não à análise de requisitos, que se concentra na identificação e documentação das necessidades.

Apenas a coleta de requisitos do cliente.

2.Incorreto: Embora a coleta de requisitos seja uma parte crucial da análise de requisitos, ela não abrange todas as etapas, que incluem também a identificação, documentação e validação das necessidades de todas as partes interessadas.

A identificação, documentação e validação das necessidades e expectativas dos stakeholders.

3.Correto: A análise de requisitos abrange a identificação, documentação e validação das necessidades e expectativas de todos os stakeholders, garantindo uma compreensão abrangente do que o sistema deve realizar.

A definição dos prazos de entrega para cada fase do projeto.

4.Incorreto: A definição dos prazos de entrega está mais relacionada ao planejamento do projeto do que à análise de requisitos, que foca na compreensão das necessidades antes de determinar a execução.

A implementação de testes de software para garantir a qualidade do produto final, seguindo as metodologias ágeis.

5.Incorreto: A implementação de testes de software está relacionada à garantia de qualidade e ao ciclo de vida do desenvolvimento, não diretamente à análise de requisitos, embora ambas estejam inter-relacionadas durante o desenvolvimento do software.

4)

1. Correta. No modelo cascata, todas as funcionalidades são desenvolvidas e entregues de uma só vez, seguindo uma abordagem sequencial, onde cada fase é concluída antes de passar para a próxima. Por outro lado, na entrega incremental, as funcionalidades são entregues em várias iterações, permitindo entregas parciais ao longo do tempo, o que proporciona maior flexibilidade e adaptabilidade às mudanças de requisitos e feedback do cliente.

2. Errado: No modelo cascata, embora seja mais difícil e custoso realizar mudanças nos requisitos em fases posteriores do desenvolvimento devido à natureza sequencial do processo, as mudanças ainda podem ser incorporadas, embora possam ser mais complicadas. Na entrega incremental, as mudanças nos requisitos são mais facilmente incorporadas em qualquer fase do desenvolvimento devido à natureza iterativa e flexível do processo.

3. Errado: O modelo cascata não é uma abordagem ágil. Ele é uma metodologia tradicional que segue uma abordagem sequencial, onde cada fase do desenvolvimento é concluída antes de passar para a próxima. Por outro lado, a entrega incremental é mais alinhada com abordagens ágeis, como o Scrum e o Kanban, onde o desenvolvimento é iterativo e incremental.

4. Errado: Não há uma regra estrita sobre o tamanho do projeto que determine a adequação da entrega incremental ou do modelo cascata. Ambos os modelos podem ser aplicados a projetos de diferentes tamanhos, dependendo da natureza do projeto e dos requisitos do cliente. A escolha entre os dois modelos geralmente depende da preferência da equipe de desenvolvimento, das necessidades do cliente e do contexto do projeto.

5. Errado: No modelo cascata, cada fase do desenvolvimento é concluída antes de passar para a próxima, o que significa que as funcionalidades são adicionadas e entregues em uma única iteração, em vez de serem entregues em incrementos sucessivos. Na entrega incremental, as funcionalidades são entregues em várias iterações, onde cada iteração adiciona uma parte do escopo do projeto, resultando em entregas incrementais ao longo do tempo. Portanto, a afirmação está invertida em relação às características de cada modelo.

5)

2. Correta. O Ciclo de Vida de Software engloba todas as fases do projeto, desde a concepção até a manutenção, enquanto a Engenharia de Requisitos é uma parte específica do ciclo relacionada à identificação, análise, documentação e validação dos requisitos do software.

1. Errado: Embora a Engenharia de Requisitos seja de fato um método específico dentro do Ciclo de Vida de Software para a documentação de requisitos, o Ciclo de Vida de Software não se limita apenas à documentação de requisitos, mas abrange todas as fases do projeto de software.

3. Errado: O Ciclo de Vida de Software e a Engenharia de Requisitos não são termos intercambiáveis. O Ciclo de Vida de Software refere-se à gestão de todo o processo de desenvolvimento de software, enquanto a Engenharia de Requisitos é uma das atividades realizadas dentro desse ciclo.

4. Errado O Ciclo de Vida de Software não é uma metodologia ágil. Ele é um conceito que abrange diversas metodologias e abordagens para o desenvolvimento de software, tanto ágeis quanto tradicionais. Já a Engenharia de Requisitos não é uma abordagem para o desenvolvimento de software, mas sim uma disciplina que trata da elicitação, análise, documentação e validação dos requisitos do software.

5. Errado: O Ciclo de Vida de Software não trata apenas das etapas de desenvolvimento do software. Ele engloba todas as fases do projeto, desde a concepção até a manutenção, enquanto a Engenharia de Requisitos não se limita apenas à identificação dos requisitos, mas também inclui atividades como análise, documentação e validação dos requisitos.

6)

5.Correta. O termo "Engenharia de Software" refere-se à aplicação de princípios de engenharia para o desenvolvimento, operação e manutenção de sistemas de software. Isso inclui não apenas o processo de desenvolvimento de software, mas também aspectos como planejamento, análise, design, implementação, teste, manutenção e gestão de projetos de software.

1. Errado: A Engenharia de Software não se refere apenas à aplicação de métodos para criar a interface de usuário de um software. Ela abrange todo o processo de desenvolvimento de software, desde a concepção até a manutenção, e não se limita apenas à interface de usuário.

2. Errado\* Embora a gestão de recursos humanos seja uma parte importante das empresas de desenvolvimento de software, a Engenharia de Software vai além do gerenciamento de recursos humanos e inclui todos os aspectos técnicos relacionados ao desenvolvimento de software.

3. Errado: A Engenharia de Software não se refere exclusivamente à codificação de programas de computador. Ela inclui todas as atividades envolvidas no desenvolvimento de software, desde a análise de requisitos até a manutenção do sistema.

4. Errado: A Engenharia de Software não exclui a necessidade de pensar em como o cliente pode ajudar o funcionário a escolher as ferramentas. Na verdade, a interação com o cliente e a compreensão de suas necessidades são aspectos fundamentais da Engenharia de Software, e a escolha das ferramentas é uma decisão que deve ser tomada com base nos requisitos do cliente e nas melhores práticas de engenharia de software.

7)

3.Correta. A Engenharia de Software é uma disciplina que existe desde os primórdios da computação. Ela surgiu da necessidade de aplicar princípios de engenharia na criação e no desenvolvimento de software, garantindo sua qualidade, confiabilidade e eficiência. Desde o surgimento dos primeiros computadores, a Engenharia de Software tem sido uma preocupação dos profissionais da área, buscando constantemente melhorar as práticas de desenvolvimento de software.

1. Errado: A afirmação de que a Engenharia de Software sempre foi alvo dos profissionais da área, mas apenas em empresas de grande porte, é falsa. A Engenharia de Software é aplicável em empresas de todos os tamanhos e sempre foi uma preocupação dos profissionais da área, independentemente do porte da empresa.

2. Errado: A Engenharia de Software não é uma abordagem obsoleta no desenvolvimento de software moderno. Pelo contrário, ela continua sendo uma disciplina fundamental para garantir a qualidade, confiabilidade e eficiência dos sistemas de software em um ambiente cada vez mais complexo e dinâmico.

4. Errado: A Engenharia de Software não é uma disciplina recente e só ganhou destaque nas últimas décadas. Ela existe desde os primórdios da computação e tem sido uma preocupação dos profissionais da área desde então.

5. Errado: Embora a Engenharia de Software inclua atividades como criação de planos e blueprints de softwares, seu foco não é exclusivamente nisso. Ela abrange todo o ciclo de vida do desenvolvimento de software, desde a concepção até a manutenção, e envolve uma variedade de atividades, incluindo análise de requisitos, design, implementação, teste e gestão de projetos.

8)

1.Correta. O SCRUM é um conjunto de práticas de gerenciamento de projetos ágeis, centradas na colaboração, transparência e adaptação. Ele é uma metodologia ágil que visa maximizar a produtividade da equipe e a qualidade do produto por meio de iterações curtas, chamadas de Sprints, onde são entregues incrementos do produto funcional.

2. Errado: O SCRUM não é uma técnica de teste de software, mas sim uma metodologia de gerenciamento de projetos.

3. Errado: O SCRUM não é um protocolo de comunicação, mas sim uma metodologia de desenvolvimento de software.

4. Errado: Embora o SCRUM enfatize a entrega contínua de software funcional, ele não é um modelo de desenvolvimento de software por si só, mas sim uma metodologia de gerenciamento de projetos.

5. Errado: O SCRUM não é uma linguagem de programação, mas sim uma metodologia de gerenciamento de projetos ágeis.

9)

4.Correta. Backlog da Sprint (Sprint Backlog). Ele contém todas as tarefas que a equipe se comprometeu a realizar durante o Sprint atual, sendo um documento dinâmico que é atualizado ao longo do Sprint conforme o progresso é feito e novas tarefas são adicionadas ou removidas.

1.Errado: O Gráfico de Velocidade (Velocity Chart) mostra a quantidade de trabalho concluída em cada Sprint ao longo do tempo, não rastreia o trabalho a ser realizado durante um Sprint.

2.Errado: O Backlog do Produto (Product Backlog) contém todas as funcionalidades desejadas para o produto, priorizadas pelo Product Owner. Ele não é utilizado para rastrear o trabalho durante um Sprint específico, mas sim para fornecer uma visão geral das necessidades do produto.

3.Errado: O Incremento é o resultado do trabalho realizado durante o Sprint, ou seja, é o produto funcional e potencialmente entregável ao final de cada Sprint. Ele não é utilizado para rastrear o trabalho a ser realizado durante um Sprint específico.

5.Errado: O Gráfico de Burndown mostra o trabalho restante ao longo do tempo durante um Sprint. Ele é baseado no Backlog da Sprint, mas não é o artefato principal utilizado para rastrear o trabalho durante o Sprint, como o Backlog da Sprint.