**Marque as respectivas respostas no gabarito abaixo. Cada questão vale 0,5 pontos.**

| **QUESTÃO:** | **Q1** | **Q2** | **Q3** | **Q4** | **Q5** | **Q6** | **Q7** | **Q8** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **RESPOSTA**: | B | D | A | E | B | C | C | D |

1. **(Ano: 2011 Banca: FCC)**. FDD (Feature Driven Development) é uma metodologia muito objetiva, possuindo apenas duas fases:
2. Decomposição Funcional e Construção.
3. Concepção & Planejamento e Construção.
4. Análise dos Requisitos e Desenvolvimento.
5. Planejamento Incremental e Desenvolvimento por Funcionalidade.
6. Planejamento por Funcionalidade e Construção por Funcionalidade.
7. **(Ano: 2010 Banca: FCC).** A Feature Driven Development (FDD) é uma metodologia ágil de desenvolvimento de software, sobre a qual é correto afirmar:
8. Não pode ser combinada a outras técnicas para a produção de sistemas.
9. Divide os papéis em dois grupos: papéis chave e papéis de apoio. Dentro de cada categoria, os papéis são atribuídos a um único participante que assume a responsabilidade pelo papel.
10. Mantém seu foco apenas na fase de modelagem.
11. Possui cinco processos: Desenvolver um Modelo Abrangente, Construir a Lista de Funcionalidades, Planejar por Funcionalidade, Detalhar por Funcionalidade e Implementar por Funcionalidade.
12. Mantém seu foco apenas na fase de implementação.
13. **(Ano: 2023 Banca: CPCON)**. O Rational Unified Process (RUP), um dos mais conhecidos processos unificados, é um processo de desenvolvimento iterativo e incremental de software que utiliza a UML como linguagem de modelagem. Ele foi desenvolvido para ser uma alternativa a alguns modelos preditivos, como o modelo de desenvolvimento em cascata, visando superar um conjunto de limitações destes modelos. Seu objetivo é garantir o desenvolvimento de software de alta qualidade e atender às necessidades de seus clientes, sendo obediente ao cronograma e orçamento previsíveis. Sobre as fases do RUP, assinale a alternativa CORRETA:
14. A fase de Elaboração procura o aprimoramento da análise e do projeto do sistema. Nessa fase, é elaborada a arquitetura do sistema e planejadas as iterações futuras do projeto.
15. É na fase de manutenção que a equipe de desenvolvimento deve focar na correção preventiva ou evolutiva do sistema.
16. O processo unificado RUP (Rational Unified Process) contém quatro fases, especificamente denominadas de Concepção, Elaboração, Construção e Verificação.
17. É durante a fase de Concepção que todos os casos de uso precisam estar definidos com bom nível de detalhes para que o planejamento do desenvolvimento do sistema seja o mais preciso possível.
18. A fase de Construção é responsável pelo desenvolvimento de diversos artefatos essenciais do projeto, dentre os quais se destaca o Diagrama de Classes do projeto.
19. **(Ano: 2005 Banca: FAPEU)**. A XP (Extreme Programming) é baseada em um conjunto de regras, também conhecidas como best practices. Nesse aspecto, analise:

I. The Customer is Always Available: Na XP todas as decisões sobre o rumo do projeto devem ser tomadas pelo cliente. Ele deve priorizar as tarefas, ser responsável pelos testes de aceitação, e, acima de tudo, orientar e tirar dúvidas dos desenvolvedores durante o processo de programação.

II. A técnica de Continuous Integration diz que o código desenvolvido por cada par de desenvolvedores deve ser integrado ao código base constantemente. Quanto menor o intervalo entre cada integração, menor a diferença entre os códigos desenvolvidos e maior a probabilidade de identificação de erros, pois cada vez que o código é integrado, todos os unit tests devem ser executados, e, se algum deles falhar, é porque o código recém integrado foi o responsável por inserir erro no sistema.

III. Pair Programming – na XP todo o código deve ser produzido por duas pessoas utilizando o mesmo computador. Enquanto um dos parceiros se preocupa com detalhes da implementação, ficando responsável pela digitação do código, o outro deve tentar ter uma visão mais ampla da rotina, imaginando as suas peculiaridades.

IV. Collective Code Ownership – O código deve ser de propriedade de todos e todos devem ter permissão para alterar o que for necessário para que seu trabalho possa ser desenvolvido e, também, para se evitar atrasos no desenvolvimento, devido à necessidade de alterações nas rotinas.

É correto o que consta em:

1. I e II, apenas.
2. II e III, apenas.
3. II, III e IV, apenas.
4. III e IV, apenas.
5. I, II, III e IV.
6. **(Ano: 2024 Banca: Instituto Consulplan).** Dany trabalha em uma startup que desenvolve jogos para a web, sua equipe faz uso de diversos métodos, guia de melhores práticas, conceitos, metodologias, ferramentas, tecnologias e frameworks para a execução dos trabalhos. Diante do exposto marque a alternativa INCORRETA.
7. Com o objetivo de garantir que o trabalho seja focado e eficiente o Scrum é composto por diversos eventos, alguns são: sprint, sprint planning, daily scrum, sprint review, sprint retrospective, product backlog, release, release planning. Eles possuem duração definida que são chamadas de timebox.
8. Scrum é baseado em um processo iterativo, incremental e linear, assim cada evento dentro da sprint é realizado na sequência definida pelo método, semelhante a uma cascata.
9. Algumas das vantagens na adoção de métodos ágeis são: redução de custos, melhoria na gestão de mudanças, melhor gerenciamento de prioridades, maior retorno sobre investimento, redução de riscos, aproximação da equipe técnica com a área de negócio melhorando a comunicação.
10. O Extreme Programming - XP utiliza o conceito de planning game, que pode ser definida como uma reunião a cada iteração para decidir funcionalidades que serão adicionadas ao ciclo de desenvolvimento atual.
11. O Extreme Programming – XP ressalta que realizar testes é de extrema importância para garantir a qualidade dos artefatos de softwares produzidos, todo código deve ser testado antes das entregas. Testes são um pilar fundamental do XP, com práticas como Test-Driven Development (TDD) sendo centrais para garantir a qualidade do código.
12. **(Ano: 2024 Banca: Instituto Consulplan).** Considerando o Guia do Scrum, versão 2020 (scrumguides.org), em português, analise as assertivas abaixo sobre os Scrum Teams da metodologia ágil Scrum:

I. São multifuncionais, o que significa que os membros têm todas as habilidades necessárias para criar valor a cada Sprint.

II. São compostos de até quatro pessoas: um Product Owner, um Scrum Master e um ou dois Developers.

III. São autogerenciáveis, o que significa que decidem internamente quem faz o que, quando e como.

Quais estão corretas?

1. Apenas I.
2. Apenas II.
3. Apenas I e III.
4. Apenas II e III.
5. I, II e III.
6. **(Ano: 2024 Banca: IV - UFG).** A metodologia SCRUM segue os princípios do manifesto ágil. O SCRUM baseia-se em seis características, que são:
7. flexibilidade dos resultados, flexibilidade dos prazos, grandes times, revisões frequentes, colaboração e orientação a objetos.
8. flexibilidade dos resultados, flexibilidade dos prazos, grandes times, revisões esporádicas, colaboração e orientação a objetos.
9. flexibilidade dos resultados, flexibilidade dos prazos, times pequenos, revisões frequentes, colaboração e orientação a objetos.
10. flexibilidade dos resultados, flexibilidade dos prazos, times pequenos, revisões esporádicas, colaboração e orientação a objetos.
11. **(Ano: 2024 Banca: COPESE - UFPI).** O Scrum é um framework ágil, amplamente utilizado para gerenciar projetos de desenvolvimento de software, com foco na entrega iterativa e incremental de valor. Equipes Scrum trabalham em ciclos curtos chamados Sprints, com papéis e responsabilidades bem definidos para cada membro da equipe. Considerando o contexto acima, assinale a opção que melhor descreve o papel do Scrum Master:
12. Gerenciar o orçamento e o cronograma do projeto, alinhando-o com os objetivos estratégicos da organização.
13. Tomar decisões técnicas sobre o desenvolvimento do software, incluindo escolhas de arquitetura e ferramentas.
14. Escrever o código do sistema e garantir a qualidade técnica das entregas.
15. Facilitar a comunicação entre os membros da equipe e remover impedimentos que estejam bloqueando o progresso da equipe.
16. Definir os requisitos do sistema, priorizando-os de acordo com o valor para o negócio e o retorno sobre o investimento.
17. **(Valor 2,0)**. O Scrum é uma metodologia ágil amplamente utilizada em diversos setores devido à sua capacidade de lidar com mudanças frequentes e promover a colaboração contínua entre os membros da equipe. Considere um projeto para desenvolver um sistema de gestão de colheita, com funcionalidades como planejamento de safras, monitoramento em tempo real de colheitas, e integração com sensores agrícolas para prever condições climáticas e otimizar recursos.
18. Explique como o Scrum poderia ser aplicado neste projeto, detalhando o papel do Product Owner, Scrum Master e do Time de Desenvolvimento. Como esses papéis colaborariam para gerenciar a lista de funcionalidades prioritárias no contexto da gestão de colheita?

O Product Owner (PO) é responsável por maximizar o valor do produto, gerenciando o Backlog do Produto, que contém todas as funcionalidades do sistema, como o planejamento de safras, o monitoramento em tempo real e a integração com sensores agrícolas. O PO define as prioridades com base no valor que cada funcionalidade traz para os usuários, como os agricultores, e garante que o sistema atenda às suas necessidades ao longo do projeto. Ele mantém um diálogo contínuo com os stakeholders, como técnicos e gestores agrícolas, para alinhar as expectativas e adaptar o Backlog conforme necessário. O Scrum Master tem o papel de facilitar o processo Scrum, removendo obstáculos e garantindo que o time siga as práticas ágeis. Ele assegura que o Time de Desenvolvimento tenha o suporte necessário para entregar os incrementos de forma eficiente. O Time de Desenvolvimento, por sua vez, é composto pelos profissionais que implementam as funcionalidades priorizadas. Eles trabalham de forma colaborativa, entregando incrementos do sistema a cada Sprint. O PO, o Scrum Master e o Time de Desenvolvimento colaboram de forma contínua, com o PO priorizando as funcionalidades no Backlog do Produto, o Scrum Master garantindo que o time tenha as condições ideais para trabalhar e o Time de Desenvolvimento entregando as funcionalidades de forma iterativa e incremental, sempre ajustando-se ao feedback dos stakeholders para atender às demandas do setor agrícola de forma eficiente.

1. Durante uma Sprint, o Time de Desenvolvimento descobre que a integração com sensores agrícolas requer novos dispositivos que não estavam previstos no Backlog. Analise como o Scrum lidaria com essa descoberta, considerando os artefatos e eventos principais da metodologia, como a Reunião de Planejamento, Backlog do Produto, Backlog da Sprint e a Reunião Diária.

Quando o Time de Desenvolvimento descobre que a integração com sensores agrícolas requer novos dispositivos não previstos no Backlog, o Scrum oferece uma abordagem flexível para lidar com a situação. Durante a Reunião de Planejamento da Sprint, a equipe revisaria o Backlog do Produto com o Product Owner (PO) para avaliar a prioridade da nova demanda, ajustando o planejamento conforme necessário. Se a alteração for essencial, o item pode ser adicionado ao Backlog da Sprint ou, se não for urgente, pode ser postergado para uma Sprint futura. A Reunião Diária serviria para discutir o progresso e os obstáculos relacionados à descoberta, permitindo ajustes no plano de trabalho. O Scrum Master auxiliaria na remoção de impedimentos, como a aquisição dos novos dispositivos, garantindo que o time tenha os recursos necessários. Ao final da Sprint, na Revisão da Sprint, o time e os stakeholders poderiam revisar o impacto da mudança e discutir como ela afeta o produto final, permitindo uma adaptação contínua ao longo do desenvolvimento.

1. Discuta como o uso do Scrum impacta a qualidade do sistema de gestão de colheita, comparando-o a uma abordagem tradicional. Considere fatores como inspeção, adaptação e a entrega incremental de valor no contexto da agricultura.

O uso do Scrum impacta positivamente a qualidade do sistema de gestão de colheita ao promover inspeção contínua, adaptação rápida e entrega incremental de valor. Através das Reuniões de Revisão e Reuniões Diárias, a equipe pode receber feedback constante dos agricultores e ajustar as funcionalidades de acordo com as necessidades emergentes, como mudanças nas condições climáticas ou tecnológicas. Isso difere de abordagens tradicionais, como o modelo em cascata, que geralmente entregam o produto final de uma vez, sem a flexibilidade para ajustes durante o desenvolvimento. O Scrum permite que melhorias sejam entregues de forma incremental, garantindo que o sistema evolua constantemente e esteja alinhado com as necessidades dos usuários, resultando em um produto de maior qualidade. Já em uma abordagem tradicional, a falta de interação contínua e de flexibilidade pode levar a um sistema que não atende completamente às expectativas do cliente, gerando retrabalho e atrasos.

1. A equipe percebe que, ao final de algumas Sprints, certas funcionalidades entregues não atendem completamente às expectativas dos agricultores. Proponha estratégias dentro do framework Scrum para reduzir a ocorrência de tais problemas, garantindo maior alinhamento entre os requisitos dos clientes e os entregáveis do sistema.

.

Para reduzir a ocorrência de problemas relacionados ao não atendimento completo das expectativas dos agricultores e garantir maior alinhamento entre os requisitos dos clientes e os entregáveis do sistema, o Scrum oferece várias estratégias que podem ser implementadas de forma contínua e iterativa:

* 1. Refinamento Contínuo do Backlog do Produto:
  + O Product Owner (PO) deve garantir que o Backlog do Produto seja regularmente refinado com a colaboração do Time de Desenvolvimento e dos stakeholders, incluindo os agricultores. Esse processo de refinamento garante que os itens do backlog estejam sempre atualizados, com requisitos bem definidos e prioridades ajustadas. A revisão constante das funcionalidades e prioridades permite que o time alinhe melhor os entregáveis às expectativas dos clientes.
* 2. Engajamento Frequente com os Stakeholders:
  + Para garantir que o sistema atenda às reais necessidades dos agricultores, é essencial que o PO e o Time de Desenvolvimento mantenham um diálogo constante com os stakeholders. Isso pode ser feito por meio de demonstrações regulares, como as Revisões de Sprint, onde os agricultores e outros usuários-chave podem dar feedback imediato sobre as funcionalidades entregues. Esse feedback é fundamental para ajustar os requisitos e corrigir possíveis falhas antes que se tornem problemas maiores.
* 3. Definição Clara de "Pronto" (Definition of Done - DoD):
  + Ter uma Definição Clara de Pronto é crucial para garantir que cada funcionalidade entregue esteja completa e atenda às expectativas. A DoD deve ser revista e ajustada conforme o desenvolvimento avança, garantindo que os critérios de aceitação estejam bem definidos e alinhados com as necessidades do cliente. Isso inclui tanto aspectos técnicos quanto funcionais, como a integração com sensores agrícolas, o desempenho do sistema e a usabilidade.
* 4. Backlog de Sprint Ajustado:
  + Durante o Planejamento da Sprint, o Time de Desenvolvimento e o PO devem garantir que os itens selecionados para a Sprint sejam bem compreendidos e alinhados com as expectativas dos agricultores. Caso a equipe perceba que algum item do Backlog não será suficiente para atender a uma necessidade específica, o PO pode ajustar o Backlog da Sprint ou incluir detalhes adicionais nas tarefas para melhor atendê-las.
* 5. Incorporação de Testes e Validação Contínua:
  + Durante cada Sprint, os testes devem ser incorporados como parte do processo de qualidade contínua. Os agricultores podem ser envolvidos em sessões de testes de aceitação para validar as funcionalidades entregues, assegurando que atendem às suas expectativas. Isso ajuda a detectar problemas mais cedo e ajustar as funcionalidades antes de sua liberação definitiva.
* 6. Revisões Pós-Sprint e Retrospectivas:
  + Realizar Retrospectivas de Sprint eficazes permite que a equipe reflita sobre os problemas enfrentados durante o ciclo de desenvolvimento e identifique maneiras de melhorar a comunicação e o alinhamento com os stakeholders. A equipe pode discutir como melhorar a coleta de requisitos e o entendimento das expectativas dos agricultores, ajustando o processo para evitar a repetição de problemas nas Sprints seguintes.
* 7. Prototipagem e MVP (Produto Mínimo Viável):
  + Em vez de desenvolver grandes funcionalidades de uma vez, o time pode trabalhar na criação de protótipos ou MVPs (Produtos Mínimos Viáveis) que permitem validar funcionalidades rapidamente com os agricultores. Essa abordagem permite ajustes rápidos e evita a entrega de funcionalidades que não atendem às suas expectativas. A entrega incremental de pequenas funcionalidades com valor tangível ajuda a garantir que o sistema evolua de forma mais alinhada com as necessidades reais.
* 8. Acompanhamento de Métricas e Indicadores:
  + Para garantir que o sistema atenda continuamente às necessidades dos agricultores, o time deve definir e monitorar métricas de sucesso que estejam diretamente relacionadas ao impacto das funcionalidades no campo. Isso pode incluir métricas de desempenho do sistema, precisão na previsão de safras, eficácia da integração com sensores agrícolas e satisfação do usuário final. Essas métricas podem ser discutidas nas Revisões de Sprint para identificar áreas de melhoria.

.