**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 no desenvolvimento de software 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 aplicativo de gestão de saúde pessoal, com funcionalidades como monitoramento de atividades físicas, rastreamento de saúde mental e integração com dispositivos de saúde conectados.
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?

O Scrum pode ser aplicado ao desenvolvimento do aplicativo de gestão de saúde pessoal por meio de ciclos iterativos e incrementais que garantem flexibilidade para lidar com mudanças frequentes. O Product Owner (PO) é responsável por priorizar e gerenciar o Product Backlog, garantindo que as funcionalidades mais importantes, como monitoramento de atividades físicas e integração com dispositivos de saúde conectados, sejam desenvolvidas primeiro. O PO atua como a ponte entre os stakeholders e a equipe, ajustando continuamente o backlog com base no feedback e nas necessidades do mercado. O Scrum Master facilita os eventos do Scrum (como Sprint Planning e Daily Scrum), ajuda a remover impedimentos que possam comprometer o progresso da equipe e assegura que os princípios ágeis sejam seguidos, promovendo a inspeção e adaptação contínuas. O Time de Desenvolvimento, por sua vez, é multifuncional e responsável por transformar os itens do backlog em incrementos funcionais e testáveis ao final de cada Sprint. Esses papéis colaboram no gerenciamento da lista de funcionalidades prioritárias durante o Sprint Planning, onde o PO apresenta os itens mais importantes e a equipe avalia o que pode ser entregue dentro da Sprint, mantendo um fluxo de comunicação constante para ajustar prioridades e garantir alinhamento.

1. Durante uma Sprint, o Time de Desenvolvimento descobre que uma integração com dispositivos de saúde conectados requer uma atualização de hardware que não estava prevista no Backlog. Analise como o Scrum lidaria com essa descoberta, considerando os artefatos e eventos principais da metodologia.

Quando o Time de Desenvolvimento descobre durante a Sprint que a integração com dispositivos de saúde conectados exige uma atualização de hardware não prevista no Backlog, o Scrum fornece ferramentas para lidar com essa situação de maneira ágil. A necessidade seria registrada como um novo item no Product Backlog, priorizado pelo Product Owner conforme seu impacto no produto. O time avaliaria o efeito dessa descoberta no escopo da Sprint durante eventos como o Daily Scrum, permitindo discutir ajustes e replanejar as tarefas. Se a nova necessidade comprometer o objetivo da Sprint atual, o PO e o Time de Desenvolvimento podem decidir, em conjunto, pausar a Sprint, reavaliar prioridades ou ajustar o Sprint Backlog de forma controlada, respeitando a transparência e a comunicação contínua. Essa descoberta seria apresentada durante a Sprint Review, onde o progresso e os desafios são compartilhados com os stakeholders. Posteriormente, na Sprint Retrospective, a equipe refletiria sobre como prever melhor situações semelhantes no futuro, fortalecendo processos de inspeção e adaptação para evitar surpresas de grande impacto em Sprints futuras.

1. Discuta como o uso de Scrum impacta a qualidade do produto final neste cenário, comparando-o a uma abordagem tradicional. Considere fatores como inspeção, adaptação e a entrega incremental de valor.

O uso do Scrum impacta positivamente a qualidade do produto final ao promover ciclos curtos de desenvolvimento e entregas incrementais, permitindo que problemas sejam identificados e corrigidos precocemente, diferentemente de uma abordagem tradicional como o modelo cascata. No Scrum, a inspeção contínua ocorre por meio de eventos como o Daily Scrum e a Sprint Review, onde o progresso e os incrementos são avaliados com base em critérios claros, garantindo que o produto atenda às expectativas do cliente a cada iteração. A adaptação é outro fator crucial, pois o backlog é constantemente revisado e ajustado pelo Product Owner, permitindo que mudanças nas prioridades ou descobertas técnicas, como a necessidade de uma atualização de hardware, sejam incorporadas sem comprometer o fluxo do projeto.

Já na abordagem tradicional, o planejamento detalhado e sequencial limita a capacidade de resposta a mudanças, levando frequentemente à descoberta tardia de problemas, aumentando custos e retrabalho. No Scrum, a entrega incremental de valor permite que o cliente receba e avalie partes funcionais do produto ao final de cada Sprint, contribuindo com feedback contínuo para ajustar o desenvolvimento. Isso reduz o risco de entregar um produto desalinhado com as necessidades do mercado e melhora a qualidade final ao incorporar melhorias ao longo do projeto, enquanto o modelo tradicional tende a entregar todo o produto apenas no final, dificultando ajustes baseados em feedback. Assim, o Scrum proporciona maior alinhamento com as expectativas, além de um produto mais funcional e adaptado às necessidades do usuário.

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

Para reduzir a ocorrência de funcionalidades que não atendem às expectativas do cliente dentro do framework Scrum, várias estratégias podem ser implementadas:

* Melhorar o Refinamento do Product Backlog: Realizar sessões regulares de refinamento com o Product Owner, o Time de Desenvolvimento e os stakeholders para detalhar os requisitos, esclarecer ambiguidades e alinhar expectativas antes de começar o desenvolvimento.
* Definição de Pronto (Definition of Done): Estabelecer critérios claros e objetivos para considerar uma funcionalidade como "pronta", incluindo validação com o cliente, testes completos e documentação necessária.
* Feedback Constante nas Sprint Reviews: Garantir que os stakeholders participem ativamente das Sprint Reviews, oferecendo feedback detalhado sobre os incrementos entregues. Isso ajuda a identificar e ajustar problemas mais cedo.
* Adoção de Prototipagem e Testes Intermediários: Sempre que possível, desenvolver protótipos ou versões mínimas viáveis (MVPs) para validar funcionalidades com o cliente antes de investir no desenvolvimento completo.
* Melhorar a Comunicação com o Cliente: Ampliar o envolvimento do cliente no processo, utilizando ferramentas como demonstrações intermediárias ou reuniões regulares, para alinhar expectativas e ajustar requisitos.
* Revisão da Retrospective: Utilizar a Sprint Retrospective para discutir as razões por trás das falhas em atender às expectativas e implementar melhorias nos processos internos, como revisão dos critérios de aceitação ou maior atenção ao feedback inicial.