QUIZ 1

1. O que é engenharia de software?
A) Codificação de programas
B) Aplicação de engenharia ao hardware
C) Aplicação sistemática ao desenvolvimento de software
D) Criação de sistemas operacionais

2. Qual característica distingue o software do hardware?
A) É físico
B) Não apresenta defeitos
C) Não se desgasta fisicamente
D) Não requer manutenção

3. Qual é uma das principais causas da deterioração do software?
A) Poeira e impacto
B) Código mal formatado
C) Alterações sucessivas
D) Falta de eletricidade

4. O que é um software legado?
A) Um software de jogos
B) Um sistema recém-criado
C) Um sistema antigo, crítico e difícil de manter
D) Um software gratuito

5. Qual das opções NÃO é um domínio de aplicação de software?
A) Software de linha de produtos
B) Software agrícola
C) Aplicações Web
D) Software embarcado

6. Qual a primeira atividade em um processo de software?
A) Modelagem
B) Construção
C) Comunicação
D) Entrega

7. O que envolve a atividade de modelagem?
A) Escrever código
B) Testar o sistema
C) Criar representações do sistema
D) Controlar mudanças

8. Qual princípio destaca o valor ao usuário?
A) KISS
B) A razão de existir
C) Pense
D) Esteja aberto para o futuro

9. O princípio KISS indica que o projeto deve ser:
A) Inovador
B) Minimalista e eficaz
C) Extensivo

D) Complicado para evitar falhas

10. O que caracteriza um bom planejamento?
A) Fazer testes
B) Iniciar o projeto sem requisitos
C) Definir tarefas, riscos e cronograma
D) Escrever documentação apenas no fim

11. Qual é a última fase do processo de software?
A) Construção
B) Planejamento
C) Comunicação
D) Entrega

12. O que representa a curva de defeitos idealizada para software?
A) Aumento contínuo de defeitos
B) Desgaste físico
C) Redução após correção de falhas
D) Falta de bugs

13. Que prática está relacionada à reutilização de software?
A) Refatoração contínua
B) Projeto pensando em futuro uso
C) Reescrever o código do zero
D) Testes automatizados

14. Qual atividade lida com controle de versões?
A) Engenharia de requisitos
B) Gerência de configuração
C) Teste de software
D) Qualidade externa

15. Quem são os envolvidos na fase de comunicação?
A) Somente programadores
B) Apenas gerentes
C) Cliente e equipe de software
D) Designers

16. Qual tipo de processo é incremental?
A) Cascata
B) Espiral
C) Ágil
D) Funcional

17. Qual é o objetivo da modelagem?
A) Codificar com mais rapidez
B) Avaliar visualmente o sistema
C) Evitar testes
D) Fazer manutenção

18. Qual princípio alerta contra “colchas de retalhos” no sistema?
A) Pense
B) Mantenha a visão
C) Planeje para reutilização

D) KISS

19. O que os testes de software revelam?
A) A ausência de bugs
B) Apenas falhas críticas
C) A presença de bugs
D) Erros de compilação

20. O que é considerado na qualidade externa?
A) Código-fonte
B) Arquitetura
C) Documentação interna
D) Fatores percebidos pelo usuário

QUIZ 2

1. Quais são as quatro dificuldades essenciais descritas por Fred Brooks no desenvolvimento de software?

2. Por que a metáfora "não existe bala de prata" é importante para a engenharia de software?

3. O que diferencia uma dificuldade essencial de uma dificuldade acidental no desenvolvimento de software?

4. Qual é a definição de um processo de software?

5. Por que é importante seguir um processo de software em projetos modernos?

6. Quais são os quatro tipos de fluxo de processo descritos no conteúdo?

7. Como o processo de software contribui para a organização e avaliação do trabalho em equipe?

8. Qual é a principal característica do modelo de processo linear?

9. Em que situações o modelo iterativo é mais apropriado?

10. Quais são as seis ações envolvidas na comunicação em projetos complexos?

11. Qual modelo de processo foi inspirado nos métodos tradicionais de engenharia?

12. Quais são as principais vantagens do modelo Waterfall?

13. Quais são as maiores limitações do modelo Waterfall em projetos reais?

14. O que é prototipação e quando ela deve ser usada?

15. Quais são os riscos associados ao uso excessivo de prototipação?

16. Como funciona o modelo espiral e quais suas vantagens?

17. Qual a principal diferença entre o modelo espiral e os demais modelos?

18. O que caracteriza o Processo Unificado em relação a outros modelos?

19. Quais são as quatro fases do Processo Unificado?

20. Quais as principais vantagens e desvantagens do modelo de Processo Unificado?

QUIZ 3

1. Qual era a principal limitação dos modelos tradicionais de desenvolvimento de software como o modelo cascata?

2. O que motivou o surgimento dos métodos ágeis?

3. O que diz o Manifesto Ágil sobre indivíduos e processos?

4. Quais são os cinco princípios fundamentais dos métodos ágeis?

5. O que significa “acolher mudanças” no contexto ágil?

6. Por que a entrega incremental é importante nos métodos ágeis?

7. O que é valorizado em relação às pessoas nos métodos ágeis?

8. Quais são os principais valores do XP?

9. O que significa "baby steps" no contexto do XP?

10. Como o princípio da responsabilidade pessoal se aplica no XP?

11. O que é programação em pares e por que é utilizada no XP?

12. Qual o objetivo dos testes automatizados no XP?

13. O que é TDD e como funciona?

14. O que é um build automatizado?

15. Como funciona a integração contínua no XP?

16. Quais são os papéis principais do Scrum?

17. O que é a Sprint e qual seu papel dentro do Scrum?

18. Quais são os três principais artefatos do Scrum?

19. O que é o Quadro Kanban e como ele auxilia na gestão de projetos?

20. Como o DevOps integra desenvolvimento e operações no ciclo de vida do software?

Explicações

Quiz 1

1. O que é engenharia de software?
☑ C) Aplicação sistemática ao desenvolvimento de software
🔎 Explicação: A engenharia de software é definida como a aplicação de uma abordagem sistemática, disciplinada e quantificável ao desenvolvimento, operação e manutenção de software. Não se trata apenas de programar, mas de aplicar processos e práticas da engenharia para garantir qualidade e eficiência.

2. Qual característica distingue o software do hardware?
☑ C) Não se desgasta fisicamente
🔎 Explicação: O software não sofre desgaste físico como o hardware. Ele não se deteriora por poeira ou impacto, mas sim por alterações acumuladas, o que caracteriza deterioração lógica e não física.

3. Qual é uma das principais causas da deterioração do software?
☑ C) Alterações sucessivas
🔎 Explicação: A cada modificação feita no software (para corrigir erros ou adicionar funcionalidades), há o risco de introduzir novos defeitos, o que eleva sua taxa de erros e contribui para a deterioração ao longo do tempo.

4. O que é um software legado?
☑ C) Um sistema antigo, crítico e difícil de manter
🔎 Explicação: Sistemas legados são antigos, muitas vezes mal documentados e difíceis de manter, mas continuam sendo fundamentais para o funcionamento de operações críticas de negócios.

5. Qual das opções NÃO é um domínio de aplicação de software?
☑ B) Software agrícola
🔎 Explicação: “Software agrícola” não é um domínio formal reconhecido. Os domínios canônicos incluem embarcados, aplicativos, IA, etc. Embora existam softwares usados na agricultura, eles se enquadram dentro de outras categorias como “software de aplicação”.

6. Qual a primeira atividade em um processo de software?
☑ C) Comunicação
🔎 Explicação: Antes de planejar ou modelar, a equipe precisa entender o problema e os requisitos do cliente. Isso acontece na fase de comunicação, onde há coleta de informações, entendimento dos objetivos e requisitos do projeto.

7. O que envolve a atividade de modelagem?
☑ C) Criar representações do sistema
🔎 Explicação: Modelagem envolve a criação de representações visuais e conceituais do sistema para facilitar o entendimento, planejamento e implementação. É como um esboço técnico que orienta o projeto.

8. Qual princípio destaca o valor ao usuário?
☑ B) A razão de existir
🔎 Explicação: Esse princípio afirma que todo sistema deve agregar valor ao

usuário. Nenhuma funcionalidade deve ser adicionada se não trouxer benefícios reais e mensuráveis.

9. O princípio KISS indica que o projeto deve ser:
☑ B) Minimalista e eficaz
🔎 Explicação: KISS significa “Keep It Simple, Stupid”. A ideia é manter o projeto o mais simples possível para garantir fácil entendimento, manutenção e menor propensão a erros — sem ser simplista ou mal feito.

10. O que caracteriza um bom planejamento?
☑ C) Definir tarefas, riscos e cronograma
🔎 Explicação: Planejamento cria um “mapa” do projeto que inclui tarefas a serem feitas, riscos esperados, prazos, recursos e artefatos a serem produzidos.

11. Qual é a última fase do processo de software?
☑ D) Entrega
🔎 Explicação: Depois de planejar, modelar, construir e testar, o software (ou uma parte funcional dele) é entregue ao cliente para avaliação e feedback.

12. O que representa a curva de defeitos idealizada para software?
☑ C) Redução após correção de falhas
🔎 Explicação: A curva idealizada mostra que os defeitos são altos no início, mas vão diminuindo com as correções, e a curva se estabiliza. Isso difere da curva real, onde novas alterações elevam novamente a taxa de erros.

13. Que prática está relacionada à reutilização de software?
☑ B) Projeto pensando em futuro uso
🔎 Explicação: Reutilização exige planejamento. Componentes reutilizáveis só são eficientes se foram criados com essa intenção desde o início, prevendo modularidade, documentação e padronização.

14. Qual atividade lida com controle de versões?
☑ B) Gerência de configuração
🔎 Explicação: A gerência de configuração gerencia versões do software, garantindo que mudanças sejam rastreáveis e controladas. Envolve muito mais que apenas usar Git; inclui políticas e processos de controle.

15. Quem são os envolvidos na fase de comunicação?
☑ C) Cliente e equipe de software
🔎 Explicação: A comunicação precisa ser feita com todos os envolvidos no projeto, especialmente os stakeholders (clientes, usuários, patrocinadores), para garantir alinhamento e entendimento claro das necessidades.

16. Qual tipo de processo é incremental?
☑ C) Ágil
🔎 Explicação: Os métodos ágeis trabalham com entregas incrementais e iterativas, permitindo adaptações constantes. Em contraste, o modelo cascata é linear e sequencial.

17. Qual é o objetivo da modelagem?
☑ B) Avaliar visualmente o sistema
🔎 Explicação: Modelos permitem abstrair a complexidade do código e representar graficamente o sistema, facilitando a análise e planejamento sem mergulhar nos detalhes técnicos.

18. Qual princípio alerta contra “colchas de retalhos” no sistema?
☑ B) Mantenha a visão
🔎 Explicação: Um sistema sem visão clara pode se tornar inconsistente. Esse princípio reforça a importância de uma arquitetura sólida e coesa durante todo o projeto.

19. O que os testes de software revelam?
☑ C) A presença de bugs
🔎 Explicação: Como disse Dijkstra, testes mostram a presença de bugs, mas nunca podem provar a ausência total deles. Logo, testes são meios de encontrar erros, não de garantir perfeição.

20. O que é considerado na qualidade externa?
☑ D) Fatores percebidos pelo usuário
🔎 Explicação: A qualidade externa é percebida por quem usa o sistema (ex: usabilidade, desempenho, confiabilidade), sem necessidade de entender o código-fonte.

QUIZ 2

1. Quais são as quatro dificuldades essenciais descritas por Fred Brooks no desenvolvimento de software?
◇ Complexidade: o software é altamente interconectado e difícil de entender como um todo.
◇ Conformidade: precisa se adaptar rapidamente a mudanças externas, como leis e regras.
◇ Facilidade de mudanças: sistemas bem-sucedidos precisam ser constantemente modificados.
◇ Invisibilidade: por ser abstrato, é difícil visualizar o software e estimar esforço e progresso.

2. Por que a metáfora "não existe bala de prata" é importante para a engenharia de software?
◇ Porque ela mostra que não existe uma solução mágica que resolva todos os desafios da engenharia de software. Os problemas essenciais são inerentes à natureza do software e não podem ser completamente eliminados com novas ferramentas ou métodos.

3. O que diferencia uma dificuldade essencial de uma dificuldade acidental no desenvolvimento de software?
◇ Essenciais: fazem parte da própria natureza do desenvolvimento de software (ex: complexidade, invisibilidade).
◇ Acidentais: resultam de ferramentas, tecnologias ou práticas ruins, e podem ser resolvidas com conhecimento e boas decisões (ex: framework sem documentação).

4. Qual é a definição de um processo de software?
◇ É uma metodologia estruturada que define atividades, ações e tarefas necessárias para desenvolver software de alta qualidade, organizando as etapas desde o planejamento até a entrega.

5. Por que é importante seguir um processo de software em projetos modernos?

◇ Porque projetos modernos são realizados em equipe e são grandes demais para serem feitos por uma só pessoa. O processo garante organização, qualidade, coordenação e produtividade, além de alinhar o produto aos objetivos da empresa.

6. Quais são os quatro tipos de fluxo de processo descritos no conteúdo?
Linear: etapas em sequência, sem volta (como no modelo cascata).
Iterativo: etapas podem ser revisitadas (ex: revisões constantes).
Evolucionário: o sistema é entregue em versões crescentes e funcionais.
Paralelo: várias atividades ocorrem ao mesmo tempo (ex: desenvolvimento e testes simultâneos).

7. Como o processo de software contribui para a organização e avaliação do trabalho em equipe?
◇ Ele define funções, responsabilidades, atividades e prazos, facilitando o controle de qualidade, a colaboração e a avaliação de desempenho. Isso aumenta a produtividade e previsibilidade do desenvolvimento.

8. Qual é a principal característica do modelo de processo linear?
◇ A sequência rígida de etapas. Cada fase (requisitos, projeto, implementação, etc.) deve ser concluída antes da próxima começar, o que permite planejamento detalhado, mas dificulta mudanças posteriores.

9. Em que situações o modelo iterativo é mais apropriado?
◇ Quando há incerteza ou mudanças frequentes nos requisitos. Permite revisitar etapas conforme o projeto avança, ajustando-o com base em feedbacks ou novas informações.

10. Quais são as seis ações envolvidas na comunicação em projetos complexos?
Concepção
Levantamento
Elaboração
Negociação
Especificação
Validação
◇ Essas ações garantem que todas as necessidades dos diversos envolvidos sejam consideradas.

11. Qual modelo de processo foi inspirado nos métodos tradicionais de engenharia?
◇ O modelo Waterfall (Cascata), criado na década de 1970, foi baseado nos processos sequenciais usados na engenharia civil e mecânica, com foco em documentação e planejamento detalhado.

12. Quais são as principais vantagens do modelo Waterfall?
◇ Facilidade de entendimento e planejamento, previsibilidade de prazos e custos, e foco na documentação e qualidade em todas as etapas.

13. Quais são as maiores limitações do modelo Waterfall em projetos reais?
◇ Inflexibilidade a mudanças, dificuldade de os clientes preverem tudo no início, e feedback tardio, o que pode fazer com que erros só sejam detectados no final do projeto.

14. O que é prototipação e quando ela deve ser usada?
◇ É o desenvolvimento de uma versão preliminar do software (protótipo), usada

para testar ideias e obter feedback do cliente. É ideal quando há incertezas nos requisitos, interfaces complexas ou necessidade de envolvimento do usuário.

15. Quais são os riscos associados ao uso excessivo de prototipação?
◇ Pode levar a atrasos, pois o cliente muda constantemente de ideia; o desenvolvedor pode "embalar" o protótipo como produto final, mesmo sem qualidade; e há risco de desperdício de trabalho em protótipos que serão descartados.

16. Como funciona o modelo espiral e quais suas vantagens?
◇ Desenvolve o software em versões incrementais, com análise de riscos a cada ciclo. Une o controle do modelo cascata com a flexibilidade da prototipação. É ideal para sistemas grandes e complexos com risco elevado.

17. Qual a principal diferença entre o modelo espiral e os demais modelos?
◇ O modelo espiral incorpora o gerenciamento de riscos como parte central do processo e permite evolução contínua do software, ao contrário dos outros modelos que terminam com a entrega final.

18. O que caracteriza o Processo Unificado em relação a outros modelos?
◇ O Processo Unificado é incremental, orientado a objetos e usa UML. Tenta equilibrar o rigor de processos clássicos com a flexibilidade das metodologias ágeis. Possui fases bem definidas, mas com sobreposição.

19. Quais são as quatro fases do Processo Unificado?
Concepção: define escopo, viabilidade e riscos iniciais.
Elaboração: detalha os requisitos e arquitetura.
Construção: desenvolvimento do sistema funcional.
Transição: testes finais e entrega ao cliente.

20. Quais as principais vantagens e desvantagens do modelo de Processo Unificado?
◇ Vantagens:
Alta documentação
Adaptação a mudanças
Envolvimento contínuo do cliente

◇ Desvantagens:
Fases sobrepostas podem ser difíceis de gerenciar
Integração de incrementos é complexa
Requer equipe experiente

QUIZ 3

1. Qual era a principal limitação dos modelos tradicionais de desenvolvimento de software como o modelo cascata?
◇ Resposta: Exigiam que todos os requisitos fossem definidos antes de começar o projeto, o que torna difícil lidar com mudanças posteriores.
🔎 Isso criava retrabalho quando os requisitos mudavam, tornando o processo inflexível para projetos dinâmicos.

2. O que motivou o surgimento dos métodos ágeis?

◇ Resposta: A insatisfação com os altos índices de fracasso em projetos, identificados pelo CHAOS Report de 1994.
🔎 Os métodos tradicionais não atendiam bem à realidade de mudanças frequentes nos requisitos, levando ao surgimento de abordagens mais adaptativas.

3. O que diz o Manifesto Ágil sobre indivíduos e processos?
◇ Resposta: Que indivíduos e interações são mais importantes do que processos e ferramentas.
🔎 Isso valoriza a colaboração entre pessoas acima da rigidez de regras, promovendo um ambiente mais flexível e eficaz.

4. Quais são os cinco princípios fundamentais dos métodos ágeis?
◇ Resposta:
Envolvimento do cliente
Acolher mudanças
Entrega incremental
Simplicidade
Valorização das pessoas sobre os processos
🔎 Esses princípios orientam a criação de software mais adaptável e alinhado às necessidades do cliente.

5. O que significa “acolher mudanças” no contexto ágil?
◇ Resposta: Aceitar que mudanças nos requisitos são normais e devem ser bem-vindas, mesmo no final do desenvolvimento.
🔎 Isso permite que o software continue relevante e útil conforme o ambiente de negócios evolui.

6. Por que a entrega incremental é importante nos métodos ágeis?
◇ Resposta: Porque permite entregar partes funcionais do software com rapidez, gerando valor constante ao cliente.
🔎 Também permite feedback rápido e ajustes contínuos no produto.

7. O que é valorizado em relação às pessoas nos métodos ágeis?
◇ Resposta: O talento, a autonomia e a colaboração das pessoas, que devem ter liberdade para definir como vão trabalhar.
🔎 Equipes auto-organizadas e motivadas tendem a produzir melhores resultados.

8. Quais são os principais valores do XP (eXtreme Programming)?
◇ Resposta: Comunicação, simplicidade, feedback, coragem, respeito e qualidade de vida.
🔎 Esses valores sustentam um ambiente colaborativo, sustentável e focado na entrega contínua de valor.

9. O que significa "baby steps" no contexto do XP?
◇ Resposta: Dar passos pequenos, seguros e contínuos no desenvolvimento.
🔎 Em vez de grandes entregas arriscadas, o XP valoriza pequenas melhorias constantes e testadas.

10. Como o princípio da responsabilidade pessoal se aplica no XP?
◇ Resposta: Cada desenvolvedor é responsável por tudo que desenvolve, incluindo teste e manutenção.
🔎 Isso evita que a responsabilidade seja repassada e aumenta a qualidade do código entregue.

11. O que é programação em pares e por que é utilizada no XP?
◇ Resposta: É quando duas pessoas trabalham juntas em uma mesma tarefa no mesmo computador.
🎾 Isso aumenta a qualidade do código, favorece o aprendizado e reduz a incidência de erros.

12. Qual o objetivo dos testes automatizados no XP?
◇ Resposta: Automatizar a verificação de que funções individuais do sistema estão se comportando como esperado.
🎾 Eles reduzem o esforço manual, facilitam a manutenção e aumentam a confiança na qualidade do sistema.

13. O que é TDD e como funciona?
◇ Resposta: Desenvolvimento guiado por testes (Test-Driven Development). Primeiro escreve-se o teste, depois o código.
🎾 Isso garante que cada parte do sistema esteja coberta por testes desde o início, prevenindo falhas.

14. O que é um build automatizado?
◇ Resposta: Um processo automático para gerar uma versão executável do software pronta para ser testada ou entregue.
🎾 Automatização evita erros humanos e acelera o ciclo de desenvolvimento.

15. Como funciona a integração contínua no XP?
◇ Resposta: Os desenvolvedores integram (compartilham) seu código diariamente no repositório comum.
🎾 Isso detecta rapidamente conflitos e garante que o sistema continue funcionando à medida que evolui.

16. Quais são os papéis principais do Scrum?
◇ Resposta:
Product Owner: responsável pelo valor do produto e gerenciamento do backlog.
Scrum Master: guia a equipe na aplicação do Scrum.
Developers: constroem o produto.
🎾 Esses papéis formam o Scrum Team, que é auto-organizado e multifuncional.

17. O que é a Sprint e qual seu papel dentro do Scrum?
◇ Resposta: É um ciclo curto (até 1 mês) em que uma parte funcional do software é planejada, desenvolvida e entregue.
🎾 Sprints promovem previsibilidade, inspeção contínua e adaptação rápida.

18. Quais são os três principais artefatos do Scrum?
◇ Resposta:
Product Backlog: lista geral de tudo que será desenvolvido.
Sprint Backlog: o que será feito na sprint atual.
Incremento: resultado funcional da sprint.
🎾 Eles organizam o trabalho e fornecem transparência ao time e aos stakeholders.

19. O que é o Quadro Kanban e como ele auxilia na gestão de projetos?
◇ Resposta: Um quadro visual com colunas (ex: A Fazer, Em Andamento, Concluído) que ajuda a controlar o fluxo de trabalho.
🎾 Torna visível o progresso das tarefas e ajuda a identificar gargalos e sobrecarga.

20. Como o DevOps integra desenvolvimento e operações no ciclo de vida do software?
◇ Resposta: Unifica as equipes de desenvolvimento e operações para automatizar e acelerar todo o ciclo:
Desenvolvimento
Testes
Entrega
Monitoramento
🎾 Essa integração reduz erros, melhora a qualidade e entrega valor com mais frequência.