**TTI 202 - Desenvolvimento Ágil**

**Atividade de Revisão e Estudo – P2**

**Profa. Ana Cláudia Tiessi**

**Profa. Ana Paula Serra**

**Atividade Individual**

***Material de apoio: Material das Aulas e Guia do Scrum***

1. Considerando o SCRUM, explique:
2. Framework do Scrum (teoria e valores).

Os pilares da teoria do Scrum são transparência(processo visível a todos), adaptação(capacidade de mudar e alcançar metas) e inspeção(inspecionar o código e encontrar pontos de melhoria).

Os valores do Scrum são comprometimento, coragem, foco, abertura e respeito, e é fundamental porque estimulam o sucesso da metodologia e bom entrosamento entre os membros da equipe

Os pilares do Scrum:

- Transparência

- Adaptação

- Inspeção

Os valores do Scrum:

- Comprometimento

- Coragem

- Foco

- Abertura

- Respeito

1. Equipe do Scrum.

Equipe composta pelo time de desenvolvimento, Scrum Master & Product Owner

PO, Scrum Master, Time Dev

1. Eventos Scrum.

Sprint, Planejamento da Sprint, Daily, Review da Sprint e Retrospectiva da Sprint

Sprint, Daily, Planning, Review, Retrospectiva

1. Artefatos do Scrum.

Product Backlog, Sprint Backlog e Incrementos

Product Backlog, Sprint Backlog e Incrementos (incrementos = passos em direção à meta do produto)

1. Considerando testes, explique:
2. TDD (*Test Driven Development*).

O TDD, em resumo, é uma metodologia de desenvolvimento onde os testes são realizados antes do código funcional, onde o desenvolvedor escreve um teste que define o comportamento esperado de uma funcionalidade

Metodologia de testes automatizados que definem o comportamento esperado e previnem erros antes do código ser desenvolvido (teste leva à criação do código)

1. BDD (*Behavion Driven Development*)

O BDD é uma metodologia de desenvolvimento que visa melhorar a comunicação entre desenvolvedores e stakeholders através da criação de testes com base em comportamento, usando uma linguagem natural e compreensível

Metodologia que facilita a comunicação entre os membros das diferentes equipes do projeto (devs, cliente, stakeholders) através de uma linguagem natural

1. Qual a importância de utilização de TDD e BDD no desenvolvimento ágil?

O BDD e o TDD oferecem ao projeto um controle dos processos e membros envolvidos, já que exige a comunicação entre os diferentes membros das equipes de desenvolvimento, stakeholders, testadores, etc. Além disso, o uso dessas metodologias permite a depuração do código, prevenção de erros e simplificação das entregas com bom planejamento e manejamento do tempo.

Ambos contribuem para o controle do projeto e dos membros na comunicação e desenvolvimento de um código coeso e que previne erros, além de contribuir para as entregas e plena aplicação dos conceitos de CI/CD.

1. Explique Integração Contínua, Entrega Contínua e Implantação Contínua. Qual sua importância no desenvolvimento ágil?

- CI:

Prática de **integrar frequentemente** o código desenvolvido por todos os membros da equipe em um repositório compartilhado (geralmente várias vezes ao dia). Cada integração é automaticamente **verificada por builds e testes** para detectar rapidamente erros

Integração constante, na qual mais de um membro da equipe trabalha para verificar e corrigir erros no código. As integrações ocorrem PELO MENOS 1x POR DIA

- CD:

Uma extensão da CI, onde o código integrado e testado automaticamente é **sempre mantido em um estado "pronto para ser liberado"** para produção. Após a CI, o código passa por estágios adicionais de teste (como testes de aceitação e de performance) e é empacotado para implantação.

O código integrado é entregue após os testes (Countinuous Delivery) e entra em produção após passar por estádios adicionais de teste, como aceitação e performance, estando pronto para ser liberado sempre (Continuous Deployment)

1. Leia com atenção o sistema a seguir:

O sistema tem como objetivo emitir passagens aéreas em um aeroporto. Sendo que seu escopo é realizar o cadastro e verificação de passageiros, consulta de voos e emissão de passagens.

Um passageiro deverá ter uma reserva, já efetuada em alguma agência de viagem ou realizar a compra da passagem diretamente no balcão de atendimento da companhia aérea. Com isso, a passagem poderá ser de dois tipos: passagem de reserva ou passagem de compra.

1. No caso de passagem de reserva, o passageiro já terá um ticket de reserva emitido anteriormente pela agência de viagem e deverá apresentá-lo no balcão da companhia aérea, onde um funcionário irá consultar no sistema externo de reservas sua validade, confirmada a reserva, o Qual é a proposta do DEVOPS e por que é um desafio para as empresas?

funcionário irá registrar na passagem de reserva o código da agência de viagem e a data e horário de solicitação da passagem.

No caso, do passageiro não possuir reserva este deverá comprá-la no balcão de atendimento (passagem de compra). Para isso, os voos são consultados e caso exista disponibilidade, o pagamento da passagem é realizado através de cartão de crédito ou cheque. No caso de cartão de crédito o número do cartão e a validade são registrados no próprio sistema e validados no sistema externo de operadora de crédito. No caso de cheque é registrado o banco, agência, conta e número do cheque. Na passagem de compra devem ser registrados o valor da passagem e descontos.

Após verificada a passagem de reserva ou solicitada a passagem de compra, o funcionário irá verificar se o passageiro já é cadastrado, consultando seus dados através do número do documento de identidade. Caso o passageiro não tenha cadastro, o funcionário deverá cadastrá-lo informando nome, data de nascimento, número do documento de identidade, número do passaporte (caso de viagens internacionais).

Os dados do voo (número do voo, destino, data e hora de partida e status (confirmado ou atrasado)) são verificados, o passageiro escolhe a poltrona que deseja sentar-se de acordo com a disponibilidade e a passagem é emitida.

Explique uma estratégia de desenvolvimento ágil. Responda:

1. Quais práticas ágeis podem ser utilizadas, liste no mínimo 4.
2. Scrum: para organizar o trabalho em sprints com reuniões regulares e papéis bem definidos.
3. TDD (Test Driven Development): para escrever testes antes da implementação do código.
4. Integração Contínua (CI): para validar automaticamente o código em cada commit.
5. Revisões de Sprint e Retrospectivas: para coleta de feedback e melhoria contínua do processo.
6. Identifiquem quem seria a equipe SCRUM.

* PO: gerente de projetos, em geral, responsável pela elaboração das histórias, transcrição das regras de negócio e critérios de aceitação, além de manter comunicação com o cliente
* Scrum Master: em geral, responsável por avaliar do desempenho da equipe, relatar suas queixas e passar ao PO e coordenar os eventos Scrum
* Devs:
  + Analistas de Requisitos
  + Designer
  + Front
  + Back
  + Q/A

1. Crie o backlog com os cartões de história, contendo prioridade.

|  |  |
| --- | --- |
| **Prioridade** | **História do Usuário** |
| Alta | Como funcionário, quero cadastrar um passageiro para que ele possa emitir passagens. |
| Alta | Como passageiro, quero comprar uma passagem no balcão para viajar. |
| Alta | Como funcionário, quero validar uma reserva com código da agência para emitir passagem de reserva. |
| Média | Como funcionário, quero consultar os voos disponíveis para informar o passageiro. |
| Média | Como passageiro, quero escolher minha poltrona de acordo com a disponibilidade. |
| Média | Como funcionário, quero registrar o pagamento por cartão e validar com operadora. |
| Baixa | Como funcionário, quero registrar pagamento com cheque, informando dados bancários. |
| Baixa | Como passageiro, quero saber se o voo está atrasado ou confirmado. |

1. Defina as sprints e os cartões de história de cada sprint.

Sprint 1:

- Como passageiro, quero comprar uma passagem no balcão para viajar.

- Como passageiro, quero escolher minha poltrona de acordo com a disponibilidade

- Como funcionário, quero consultar os voos disponíveis para informar o passageiro.

- Como funcionário, quero registrar o pagamento por cartão e validar com operadora.

- Como funcionário, quero validar uma reserva com código da agência para emitir passagem de reserva

Sprint 2:

- Como funcionário, quero cadastrar um passageiro para que ele possa emitir passagens.

- Como passageiro, quero saber se o voo está atrasado ou confirmado.

Sprint 3:

- Como funcionário, quero registrar pagamento com cheque, informando dados bancários

1. Qual seria o horário será a reunião diária e quem participará da reunião diária?

Das 10 às 10:15

Integrantes: Scrum, Devs (todos) e PO

1. Faça um esboço de como seria o Quadro de Tarefas para a 1ª. Sprint.

FODA-SE

1. Como seriam realizados os testes? BDD e TDD poderiam ser utilizados? Como?
2. **TDD:** seria usado na camada de lógica de negócio (ex: cadastro de passageiro, validação de reservas).
3. Escreve-se primeiro os **testes unitários**, depois o código.
4. **BDD:** usado para funcionalidades que envolvem regras de negócio compreensíveis por stakeholders.
5. Escreva um teste de BDD para a funcionalidade “Consultar Voo”. Considere: Funcionalidade, Contexto, 1 cenário válido (correto) e 1 cenário inválido (erro).
   * Funcionalidade: Consulta de Voos

Para verificar opções de voo disponíveis

Como funcionário

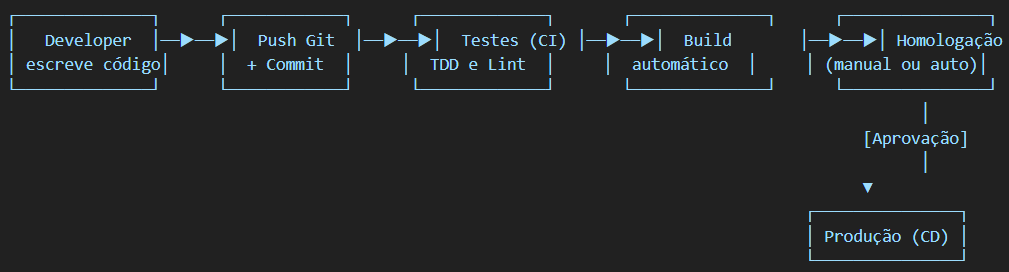
Quero consultar os voos no sistema

* + Cenário: Consulta válida de voos para um destino
    - Dado que estou na tela de consulta de voos
    - Quando eu informar o destino "São Paulo"
    - Então o sistema deve listar os voos disponíveis para "São Paulo"
  + Cenário: Consulta de voos com destino inexistente
    - Dado que estou na tela de consulta de voos
    - Quando eu informar o destino "Marte"
    - Então o sistema deve exibir a mensagem "Nenhum voo encontrado para esse destino"

1. Como a integração contínua poderia ser aplicada no desenvolvimento deste sistema.

A Integração Contínua pode ser aplicada a este sistema através dos testes automatizados, realizados diariamente após a conclusão de cada etapa de história trabalhada na sprint, ‘commits’ constantes, após cada teste, e integração desse trecho do código já testado e desenvolvido

1. Esboce um pipeline de como seria o desenvolvimento, testes unitários, integração contínua e implantação. No caso da implantação defina e justifique se será entrega ou implantação contínua.



**Para se divertir...**

1. Histórias do topo do backlog têm maior prioridade e maior tamanho (em story points).

( ) Verdadeiro ( X ) Falso

1. Times Scrum são multidisciplinares e auto-organizáveis (isto é, têm autonomia para decidir como as histórias serão implementadas).

( X ) Verdadeiro ( ) Falso

1. Reuniões diárias devem durar de 10 a 60 minutos, dependendo da duração do sprint.

( ) Verdadeiro ( X ) Falso

1. No Scrum, todos os eventos têm uma duração bem definida, chamada de time-box do evento.

( X ) Verdadeiro ( ) Falso

1. Uma das funções de um Scrum Master é organizar e facilitar os eventos que ocorrem em um sprint, tais como reunião de planejamento, reuniões diárias, revisão e retrospectiva.

( X ) Verdadeiro ( ) Falso

1. Um ciclo TDD possui dois passos: escrever um teste que falha e fazer o teste passar.

( X ) Verdadeiro ( ) Falso

1. Sempre que se usa TDD, a cobertura de testes alcança 100%.

( ) Verdadeiro ( X ) Falso

1. A pirâmide de testes tem três níveis: integração fica no topo e testes de unidade ficam na base.

( ) Verdadeiro ( X ) Falso

1. DevOps advoga que devem existir dois silos de engenheiros: desenvolvedores de software e administradores de sistemas.

( ) Verdadeiro ( X ) Falso

1. O movimento DevOps surgiu junto com os primeiros métodos ágeis, logo no início dos anos 2000.

( ) Verdadeiro ( X ) Falso

1. Deployment é o processo de liberar uma nova versão de um sistema para seus usuários finais.

( X ) Verdadeiro ( ) Falso

1. A respeito do TDD (*Test Driven Development*), é correto afirmar que:
   1. Consiste em um processo iterativo que se inicia escrevendo um código de uma funcionalidade do sistema e, logo em seguida, testa-o para saber se a implementação foi correta.
   2. Apesar de útil, não diminui o custo de implementação de testes de uma maneira geral no sistema.
   3. A ordem correta do processo TDD é primeiro escrever a funcionalidade e após, escrever os testes unitários para aquele código em específico e, por fim, refatorar o código implementado.
   4. Sua utilização elimina a necessidade de testes de validação do sistema por parte de um usuário, uma vez que ele já foi testado incrementalmente.
   5. Apesar de ter sido apresentado como parte das metodologias ágeis, também pode ser usado em outros processos de desenvolvimento de software.
2. CESPE - 2016 - TRT - 8ª Região (PA e AP) - Tecnologia da Informação - Acerca de DevOps, assinale a opção correta.
3. O DevOps concentra-se em reunir diferentes processos e executá-los mais rapidamente e com mais frequência, o que gera baixa colaboração entre equipes.
4. O DevOps tem como princípio produzir, a partir da avaliação dos times de desenvolvimento do serviço, grandes mudanças e farta documentação com valor agregado para os usuários, assemelhando-se, por isso, com objetivos dos métodos iterativos e em cascata.
5. A infraestrutura de nuvem de provedores internos e externos vem restringindo o uso de DevOps pelas organizações.
6. O DevOps parte da premissa de adoção de grandes equipes de especialistas, com a menor interação possível, visando à padronização de processos e à mínima automação de atividades.
7. Atividades típicas em DevOps compreendem teste do código automatizado, automação de fluxos de trabalho e da infraestrutura e requerem ambientes de desenvolvimento e produção idênticos.
8. Assinale a alternativa que apresenta o principal objetivo do *Behavior Driven Development* (BDD).
9. BDD foca exclusivamente na performance do código e na otimização de recursos do sistema.
10. BDD visa melhorar a comunicação entre desenvolvedores e stakeholders através da criação de testes com base em comportamento, usando uma linguagem natural e compreensível.
11. BDD é uma metodologia para o gerenciamento de projetos de software, priorizando o planejamento e a organização do trabalho.
12. BDD é uma técnica de desenvolvimento de software que se concentra apenas na refatoração do código e na eliminação de erros.
13. BDD é uma abordagem de desenvolvimento de software que ignora a necessidade de testes.