|  |  |  |
| --- | --- | --- |
| https://www.fiap.com.br/wp-content/themes/fiap/_img/logo-fiap-preto-300x100.png NAC 1 – Processo de Software1TINR **( ) Sala ( x ) Laboratório** | | |
| Design de Software **Prof. Ms. Allen Fernando Oberleitner Lima** | | |
| Nome: Yuri Breion | RM:79935 | Turma: 1TINR |

**ATENÇÃO!** Ao colocar o seu nome e RM no cabeçalho, você confirmará que leu as observações abaixo:

1. Esta prova é individual e sem consulta;
2. Não é permitido o uso de celulares, ou qualquer outra mídia de acesso a informações;
3. Não é permitido o uso de folha de rascunho, use o verso das folhas para isso;
4. Só serão tiradas dúvidas a respeito da impressão da prova. A interpretação de cada item faz parte da questão;
5. Idas ao toalete estão **proibidas** durante a prova;
6. Por prática ou tentativa de fraude (cola, identificação falsa, etc.), a prova será recolhida e sujeita a avaliação zero.
7. Preencha as questões 5, 6 e 7 no gabarito abaixo.
8. Boa prova a todos!

|  |  |  |
| --- | --- | --- |
| **5** | **6** | **7** |
| D | A | B |

1. Embora existam muitos processos de software diferentes, há atividades fundamentais comuns a todos eles. Identifique essas atividades e explique o que cada uma faz.

Obs.: Coloque as atividades em ordem lógica de acontecimento. (1,0 ponto)

***Planejamento, levantamento e especificação de requisitos, modelagem ou prototipação, testes e manutenção.***

* ***Planejamento onde tem escopo e cronograma, onde prazos e valores são definidos.***
* ***Elicitação de requisitos que contém o levantamento de requisitos, técnicas de obtenção de dados dos stakeholders e por fim os requisitos de usuário.   
  Analise de requisitos para definir requisitos de sistema (RF, RNF, RN)***
* ***Projeto, modelagem (diagrama caso de uso, o que?) (Diagrama estrutural, como?)***
* ***Implementação ou desenvolvimento, onde de fato são feitas as validações, as logicas e rotinas.***
* ***Testes/validação***
* ***Implantação, MTP, documentação e treinamento***
* ***Manutenção, suporte***

1. Os modelos de processos de software representam, cada um, um processo de software sob uma perspectiva arquitetural, ou seja, vemos a estrutura do processo, mas não os detalhes de atividades específicas. Os modelos estudados foram: CASCATA, EVOLUCIONÁRIO e INCREMENTAL.

Explique cada um dos modelos e os compare entre si, citando suas vantagens e desvantagens, além das características de cada um. (2,0 pontos)

***Modelo em cascata como o próprio nome diz, trata somente uma etapa por vez do projeto. Isso quer dizer que, para fazer a modelagem UML por exemplo, é necessário que seja feito toda etapa de elicitação de requisitos antes. Sua vantagem é ser mais controlado e seguro que os demais modelos, trata todos os itens de cada etapa antes de seguir, já sua desvantagem é o tempo que toma, por padrão, projetos neste modelo são mais extensos e demorados, sem contar que há possibilidades de gaps surgirem em etapas que já foram realizadas. Modelo não usa o feedback do cliente.***

***Já o modelo evolucionário exige uma mão de obra muito especializada. Trata de forma cíclica as demandas do projeto, onde parte do produto é trabalhado até que seja entregue com algum valor ao cliente. Sua vantagem é a ser mais rápido que o cascata por exemplo e permite trazer mais entregáveis ao cliente, em contrapartida, permite que detalhes não coletados a fundo na elicitação de requisitos passe batido, entregando parte do produto que pode não atender completamente as necessidades do cliente.***

***Modelo ágil não é mais rápido, ele entrega ao cliente pequenas partes porem no mesmo tempo que se fosse feito em cascata.***

***O modelo incremental é considerado ágil também, porem parte de um produto criado inicialmente por abstração simplesmente. Ai de acordo com os feedbacks dos stakeholders/clientes o produto vai sendo alterado em ciclos de trabalho incrementais, até que todas as partes do produto atendam as especificações do cliente. Sua vantagem é a capacidade de prototipar algo rapidamente ao cliente, o que por outro lado não prioriza o produto como um todo, gerando a necessidade de adaptações grandes em seu desenvolvimento. Partindo da premissa que o cliente não tem conhecimento técnico para ainda ver o produto como um todo. O modelo incremental é como se fosse cascata, em menor escala, onde o entregável da iteração é parte da cascata.***

1. A empresa bancária Itaó contratou você para desenvolver seu app com a finalidade de oferecer a clientes jurídicos funções para aplicações financeiras e empréstimos para negócios. A Itaó quer, com uma certa frequência, que você entregue partes do sistema para eles para que sejam testadas e validadas. Respondam as seguintes questões:
2. Identifique os stakeholders e digam quais informações eles podem nos fornecer (informe, também, qual (is) técnica (s) vai utilizar com cada stakeholder). (1,0 ponto)  
     
   ***Gerente de contas PJ – Brainstorm   
   Qual é o valor mínimo para realizar uma aplicação?  
   Todos os clientes jurídicos são elegíveis a empréstimos?   
   Existe algum segmento com taxas diferenciadas a serem definidas?  
     
   Analistas de crédito – Questionário  
   Qual o limite para um empréstimo?   
   Existe uma porcentagem por categoria?***
3. Em qual etapa do processo de construção de software estamos? (0,5 ponto)  
   ***Estamos na etapa de elicitação de requisitos***
4. Escreva os requisitos de usuário para a empresa Itaó. (2,0 pontos)  
   **O app vai permitir que os clientes jurídicos possam solicitar empréstimos de acordo com o seu segmento, onde pode haver ou não incentivos. Bem como consultar o status da sua solicitação.   
   Já para aplicações, poderão efetuar tanto em mercado de derivativos como em ativos do governo, renda fixa, a possiblidade de consulta de status também estará ativa.   
   Independente seja aplicação ou empréstimo, o cliente poderá solicitar um horário em forma de agendamento para consultoria do seu gerente.**
5. Levantar Requisitos é uma das etapas mais importantes do processo de construção de software. Essa etapa consiste em entender as necessidades do cliente para especificar o sistema na qual precisa ser feito. Existem muitas técnicas de levantamento de requisitos, dentre elas Entrevista, BrainStorm, Questionário e Observação. Explique cada uma dessas técnicas, identificando suas características, quando são usadas e suas principais vantagens. (2,0 pontos)  
     
   ***Entrevista – Encontro mais próximo com o stakeholder que permite obter informações mais precisas através de perguntas abertas ou fechadas. Dando mais liberdade ao entrevistador a saber como abordar e obter a informação. Comumente usados para obter informações mais especificas, ou até sigilosas da empresa.   
     
   Brainstorm – Encontro com vários stakeholder onde o objeto e coletar o maior número de informações possível.   
     
   Questionário – Usado para obter informações de um grupo de stakeholder de forma automatizada e com perguntas mais diretas. Usada mais com usuários finais para fins estatísticos.   
     
   Observação – Técnica usada para de fato observar as atividades dos stakeholders e definir como partes do sistema podem ser desenvolvidas.***
6. Na engenharia de requisitos trata-se de uma técnica de elicitação que ocorre em ambiente mais informal em que toda a ideia deve ser levada em consideração para a solução de um problema, sendo proibida a crítica a qualquer sugestão dada, e encorajada, inclusive, a criação de ideias que pareçam estranhas ou exóticas: (0,5 ponto)
7. Prototipação.
8. Entrevista.
9. Questionário.
10. **Brainstorming.**
11. Análise de protocolos.
12. No processo de engenharia de requisitos, os tipos de requisitos de usuário e requisitos de sistema podem ser, respectivamente: (0,5 ponto)
13. **apenas funcionais;**
14. apenas não funcionais e apenas não funcionais;
15. apenas funcionais e apenas não funcionais;
16. funcionais e não funcionais e funcionais e não funcionais;
17. apenas não funcionais e funcionais e não funcionais;
18. A gestão de requisitos é um conjunto de atividades que tem como principal objetivo ajudar a equipe de projeto a: (0,5 ponto)
19. utilizar ferramentas de engenharia de software para modelar os requisitos do sistema, através da UML.
20. **identificar, controlar e rastrear requisitos e modificações de requisitos em qualquer época, à medida que o projeto prossegue.**
21. construir um modelo técnico refinado de funções, características e restrições do software.
22. negociar com os clientes os conflitos de prioridade de requisitos e identificar e analisar os riscos associados a cada requisito.
23. avaliar os requisitos quanto à qualidade, garantindo que ambiguidades, inconsistências, omissões e erros tenham sido detectados e corrigidos.