Tópicos em Desenvolvimento de Sistemas I

Atividade Avaliativa

Questionário

Alunos: Ícaro Bicalho Quintão EC - 14.1.8083

Guilherme Ferreira Faioli Lima - 16.1.8243

João Monlevade

Novembro 2019

**1)** A escolha de produtos em paralelos com o código basicamente o que de fato buscamos e o que realmente é entregue.

**2)** É um conjunto de práticas originalmente desenvolvidas pela Motorola para melhorar sistematicamente os processos ao eliminar defeitos um defeito é definido como a não conformidade de um produto ou serviço com suas especificações. Seis Sigma também é definido como uma estratégia gerencial para promover mudanças nas organizações, fazendo com que se chegue a melhorias nos processos, produtos e serviços para a satisfação dos clientes Diferente de outras formas de gerenciamento de processos produtivos ou administrativos o Six Sigma tem como prioridade a obtenção de resultados de forma planejada e clara, tanto de qualidade como principalmente financeiros. Não é aplicável ao desenvolvimento de softwares porque o objetivo é controlar variáveis de processos para otimizar seus resultados

**3)** Um software eficiente cumpre com os requisitos de forma rápida. Já um software escalável significa aumentar as funcionalidades de um software, aumentar o consumo de memória, de processos, atender demandas de alta exigência de processamento e possivelmente concorrência de acesso. Desenvolver um software escalável significa também planejamento. Por fim, um software confiável cumpre com as funcionalidades desejadas e eficiente atende as normas e padrões

**Parte 2)**

**1. QUALIDADE FUNCIONAL**

**1.1)** Qualidade Funcional: sistema que assegura a execução consistente e a melhoria contínua das atividades funcionais de rotina, satisfazendo as necessidades e expectativas razoáveis dos clientes internos, por forma a conseguir deliciar o cliente externo.

**1.2)** Clientes externos

**1.3)**Netbeans, por se tratar de uma IDE ela prove todos os recursos para para programação e compilação a desvantagem é o alto consumo de memória ram tornando se a aplicação lenta se comparada com editores como vscode

**1.4)** Cumprir com os requisitos, ou seja, funcionar de acordo como foi projetado. Ter poucos bugs, estar com seu nível de confiabilidade de software aceitável. Ter boa performance, ou seja, ser eficiente, gastar menos recursos. Ter boa ux, de fácil entendimento e usabilidade.

**2. QUALIDADE ESTRUTURAL**

**2.1)** A qualidade do código escrito e o quão bem estruturado ele é. Pontos como modularização e re-usabilidade são essenciais nesse aspecto para garantir que o código é de fácil compreensão e manutenível — ou seja, possa ser incrementado com novas features sem que estas “quebrem” suas funções. Nesse parâmetro, os desenvolvedores são o “público alvo”, já que a legibilidade e a escalabilidade do código influenciam diretamente na ambientação de novos profissionais no projeto e, consequentemente, na velocidade de implementação.

**2.2)** Desenvolvedores

**2.3)**

* Testável: é a facilidade de se preparar rotinas para se verificar a conformidade do software com os seus requisitos.
* Sustentável: característica de um software que pode ser reutilizado ao todo ou em parte por outros softwares.
* Legível também denominada usabilidade, é a facilidade com que o software pode ser aprendido e utilizado
* Eficiente: refere-se ao bom uso que o software faz dos recursos de hardware, tais como memória e processadores.
* Seguro: é uma característica relacionada à segurança de dados, programas e documentos. Integridade é a habilidade de proteger tais componentes contra acessos não autorizados.
* Escalabilidade de softwares significa aumentar as funcionalidades de um software, aumentar o consumo de memória, de processos, atender demandas de alta exigência de processamento e possivelmente concorrência de acesso. Desenvolver um software escalável significa também planeja qualidade de processo

**3. QUALIDADE DE PROCESSO**

**3.1)** Conjuntos de melhores práticas no desenvolvimento dos projetos como também adoção de normas e padrões de projetos que asseguram o ciclo de vida e desenvolvimento de software

**3.2)** Clientes e participantes do projeto

**3.3)**

* Respeitar prazos de entrega: estabelecer e cumprir com os prazos nos quais foram acordados no projeto é essencial para que o cliente não se decepcione quanto ao desenvolvimento do projeto – cenário que pode gerar ruídos na comunicação entre as partes envolvidas
* Respeitar o orçamento original: atentar se aos requisitos preestabelecidos e como alcançá-los de maneira mais eficiente, assim como soluções eficazes sem que gaste mais tempo e consequentemente aumente o orçamento do projeto
* Alinhamento com a equipe: as equipes de produção precisam estar em plena sintonia e o gerente de projeto deve estar ciente das necessidades e do ritmo de trabalho de programadores e designers para que possa negociar um prazo adequado.
* Processo de entrega consistenteé muito frequente que haja uma mudança na metodologia de desenvolvimento no meio do projeto, seja por problemas na organização das demandas do cliente, ou por prazos mais apertados. Em todos os casos, o famoso *“*scrumterfall*”*(mistura do desenvolvimento *Scrum* com *Waterfall*) deve ser evitado: a mudança de paradigmas no curso da implementação do produto é extremamente nociva, podendo gerar estresses desnecessários para a equipe de desenvolvimento, que se vê tendo que mudar de estratégia de trabalho o tempo todo.