**ATIVIDADE DE REVISÃO**

**MATHEUS HENRIQUE BUTKOSKI SILVA**

**1)**

**SCRUM-**

O método Scrum é um framework e se baseia de um conjunto de práticas, conhecidas como metodologias ágeis, cujo objetivo é otimizar a entrega do trabalho e a qualidade do produto ou serviço que está sendo desenvolvido.

O desenvolvimento da ferramenta Scrum se baseia na definição de prioridades e em um acompanhamento periódico das etapas do projeto, sempre com foco nas necessidades do cliente.

Existem diversos papeis e etapas envolvidos no Scrum. Entre eles:

**Product Owner:** Define o cronograma e designa as funções da equipe.

**Scrum Master:** Certifica o andamento do projeto.

**Scrum Team:** Time responsável por desenvolver o projeto.

**Sprint:** Ciclo de trabalho.

**Product Backlog:** Lista de atividades que devem ser feitas durante o processo.

**Sprint Backlog:** Lista de tarefas que o Scrum Team deve cumprir em um Sprint.

No início de cada **Sprint**, faz-se um **Sprint Planning** em uma **Daily Scrum**, que como citado anteriormente é uma reunião de planejamento na qual o **Product Owner** prioriza os itens do **Product Backlog** e a equipe seleciona as atividades que ela será capaz de implementar durante o **Sprint** que se inicia.

A cada dia de uma **Sprint**, a equipe faz uma breve reunião, o objetivo é disseminar conhecimento sobre o que foi feito no dia anterior, identificar impedimentos e priorizar o trabalho do dia que se inicia. Ao final de um Sprint, a equipe apresenta as funcionalidades implementadas em uma nova reunião. Finalmente, faz-se uma retrospectiva e a equipe parte para o planejamento do próximo Sprint. Assim reinicia-se o ciclo.

**XP-**

O XP (Extreme Programming ou Programação Extrema) é uma metodologia focada no desenvolvimento de software que possui valores e princípios, onde são fundamentados por um conjunto de práticas.

É uma metodologia leve que pode facilmente ser adotada por diferentes níveis de desenvolvedores (experientes ou não) e em qualquer tamanho de equipe.

É uma excelente metodologia por se adaptar a requisitos que às vezes podem mudar rapidamente. O XP pode ser utilizado de forma complementar ao Scrum, pois ele acaba focando mais em processos de engenharia e desenvolvimento de software.

Como dito, o XP possui um conjunto de princípios e valores, onde os princípios tendem a ser mais concretos que os valores. O conjunto de valores servem como um critério que norteiam as pessoas envolvidas no desenvolvimento do software, além de se complementarem. São eles: comunicação, simplicidade, feedback, coragem e respeito. Além desses valores, existe um conjunto de princípios que deve ser seguido por equipes que forem usar XP em projetos, sendo o feedback rápido, presumir simplicidade, abraçar mudanças, trabalho de alta qualidade, pequenos passos, melhoria, diversidade, reflexão.

O SCRUM é uma forma de gestão ampla para projetos que não depende da área de conhecimento. Já o XP tem sua aplicação mais restrita, focada basicamente no mundo de desenvolvimento de sistemas de softwares. Entretanto, quando usamos o SCRUM como forma de gestão e para criação de sistemas de software, muitas das práticas contidas no XP são de grande competência, como por exemplo a criação de testes automatizados ou o uso de refatoração de código para que um trecho de código funcional seja alterado buscando um ganho de qualidade e garantindo que a manutenção futura seja simplificada.

**2)**

O TDD (Test-Driven Development) se baseia em pequenos ciclos de repetições, onde para cada funcionalidade do sistema um teste é criado antes. Este novo teste criado inicialmente falha, já que ainda não temos a implementação da funcionalidade em questão e, em seguida, implementamos a funcionalidade para fazer o teste passar

**3)**

O objetivo da Validação e da Verificação é assegurar que o SW seja adequado e se atende às necessidades, ou seja, a confirmação de que este cumpra suas especificações.

A Verificação é uma atividade, a qual envolve a análise de um sistema para certificar se este atende aos requisitos funcionais e não funcionais. Já a Validação, é a certificação de que o sistema atende as necessidades e expectativas do cliente. O processo de Validação e Verificação, não são processos separados e independentes.

**4)**

O Teste Caixa-Branca testa a codificação interna e infraestrutura de uma solução de software. Concentra-se principalmente no fortalecimento do salvaguardar, o fluxo de entradas e saídas através da aplicação e a melhoria do design e usabilidade.

O Teste Caixa-Preta por outro lado, baseia-se nos requisitos básicos do software, sendo o foco nos requisitos da aplicação, ou seja, nas ações que deve desempenhar. Os níveis de teste caixa preta são Integração, Sistema, Aceitação, Alfa e Beta. Possuem métodos e classes, comandos de repetição e condições. Se resumem em testes de entrada e saída.

A técnica de Teste Caixa-Cinza junta a Caixa-Branca e a Caixa-Preta. Dessa forma, são submetidos a análise tanto a estrutura do código-fonte quanto o cumprimento das funções do sistema.

A finalidade de utilizar a Caixa-Cinza é buscar por erros que o sistema pode apresentar. A forma com que é feita é denominada de engenharia reversa.

**5)**

**TESTE FUNCIONAL**

Os testes funcionais permitem que os testes ocorram de uma forma mais eficiente e rápida, possibilitando encontrar as não conformidades do software em relação aos requisitos do sistema.

Geralmente os testes funcionais são aplicados nas fases de teste de Integração, Sistema e Aceitação.

**TESTE UNITÁRIO**

O teste unitário consiste em verificar o comportamento das menores unidades em sua aplicação, que é focado em partes isoladas de um sistema.

No teste unitário, cada parte do sistema ganha uma atenção devida e detalhada, de modo a otimizar o processo de identificação de erros. O objetivo é ajudar a rastrear os bugs e impedir que eles retornem depois que alterações forem feitas no produto.

**TESTE DE USABILIDADE**

Teste de usabilidade é um teste do tipo caixa-preta, que verifica a facilidade que o software possui, de ser claramente compreendido e manipulado pelo usuário.

Seu objetivo é observar usuários reais usando o produto para descobrir problemas e pontos de melhorias.

Em um teste de usabilidade de software são analisados fatores como facilidade na utilização, de memorização dos atalhos e dos comandos, de localização das pastas e diretórios, capacidade autoexplicativa do sistema, entre outros. Tudo isso ajuda a prevenir erros e a indicar soluções para que a interface seja limpa, intuitiva e agradável de usar.

**6)**

|  |  |
| --- | --- |
| Caso de Teste | Caso de Teste 1 – Validar com valor 0 |
| Objetivo | Validar se o algoritmo aceita o valor 0 |
| Procedimento | 1 – Acessar Sistema  2 – Inserir o Número 0  3 - Executar |
| Dados de Entrada | 0 |
| Resultado Esperado | Erro indicando valor inválido |
| Ambiente | Windows 10 |
| Técnica | Manual |
|  |  |

|  |  |
| --- | --- |
| Caso de Teste | Caso de Teste 2 – Validar com valor não inteiro |
| Objetivo | Validar se o algoritmo aceita o valor 1,5 |
| Procedimento | 1 – Acessar Sistema  2 – Inserir o Número 1,5  3 - Executar |
| Dados de Entrada | 1,5 |
| Resultado Esperado | Erro indicando valor não inteiro |
| Ambiente | Windows 10 |
| Técnica | Manual |
|  |  |

|  |  |
| --- | --- |
| Caso de Teste | Caso de Teste 1 – Validar com valor inteiro |
| Objetivo | Validar se o algoritmo aceita o valor 2 |
| Procedimento | 1 – Acessar Sistema  2 – Inserir o Número 2  3 - Executar |
| Dados de Entrada | 2 |
| Resultado Esperado | Programa executado com sucesso |
| Ambiente | Windows 10 |
| Técnica | Manual |
|  |  |

**BIBLIOGRAFIA**

<https://www.treinaweb.com.br/blog/o-que-e-xp-extreme-programming>

<https://www.devmedia.com.br/test-driven-development-tdd-simples-e-pratico/18533>

<https://www.devmedia.com.br/a-importancia-da-validacao-e-da-verificacao/24559>