Certified Tech Developer
The Ultimate Degree

Testing I

Exercício de Fixação

Objetivo

Consolidar os conteúdos teóricos vistos na Aula 1.

 Micro desafio

Com base no conteúdo do Playground e no que foi visto nas aulas, debata na mesa de trabalho e responda às seguintes perguntas:

1. Disserte, com as suas palavras, sobre as principais subatividades que são realizadas dentro de cada atividade no ciclo de vida do teste de software.

Etapa 1: Análise de requisitos
1. Coletar o máximo de informação possível sobre o software - ferramentas a serem utilizadas, pessoas que farão parte do time, orçamento disponível.

Etapa 2: Fase de planejamento
1. Elaboramos o plano geral de testes
2. Recomendação das ferramentas a serem utilizadas e o tipo de teste (segurança, performance)
3. Estimar o tempo de trabalho para cada tipo de teste
4. Estimar o custo detalhado para cada parte do projeto/teste

Etapa 3: Integração do caso de teste
1. Elaborar os casos de testes e scripts (passo-a-passo)
2. Fase para programar um robô para que este realize o teste automaticamente (caso seja essa a opção)

Etapa 4: Configuração do ambiente de teste

1. QA ou desenvolvedor verifica os requisitos do sistema para que o teste possa ser realizado e prepara o ambiente

Etapa 5: Fase de implementação

1. Documentação dos resultados obtidos de cada teste
2. Registrar os erros/desvios em relação ao resultado esperado para cada etapa, informando o horário de realização e tester que acompanhou o teste
3. Reportar os problemas encontrados

Etapa 6: Encerramento

1. Discussão dos resultados obtidos durante o ciclo de vida do teste com o time de QA, e compartilhamento das experiências

2. Relacione cada princípio de teste com sua definição em uma planilha.

Princípio	Definição
Teste demonstra a presença de defeitos	"Mas não prova que não há nenhum bug ali..." Teste que prova a existência de bugs, porém não prova a sua não existência
Teste exaustivo é impossível	"Em alguns casos, você poderia levar meses para testar todos os cenários..." É impossível testar todos os cenários dentro do tempo disponibilizado. A questão é priorizar os testes a serem feitos dentro do prazo das sprints
Teste antecipado	"Prevenir é melhor que remediar..." Quanto antes testarmos, melhor. Se o bug for crítico, a confiabilidade do software cai e o investidor tira seu dinheiro da empresa

Agrupamento de defeitos	"Um número pequeno de módulos (20%) contém a maioria dos defeitos descobertos (80%)" Temos tentar priorizar nosso tempo com locais em que já encontramos um bug/erro anteriormente. A raiz do defeito pode ser a mesma
Paradoxo do Pesticida	"Revise os seus testes para garantir que você não está sendo enganado pelos bugs..." Se criarmos uma lista de testes a serem feitos num sistema e os desenvolvedores mudarem algo no código nesse meio tempo, esse teste não terá mais o mesmo efeito do que quando foi feito. Ele deve ser revisado periodicamente a cada alteração do time do desenvolvimento
Teste depende do contexto	"Na maioria das vezes, não é possível aplicar o mesmo teste de um sistema X em um sistema Y" O contexto para o qual foi elaborado um teste é específico. Temos que dar uma atenção especial para cada caso
A ilusão da ausência de erros	"Um sistema sem bugs, porém que não atende às necessidades dos usuários, não serve de muita coisa..." Podemos até entregar um software sem bugs para o mercado, contudo um concorrente pode entregar outro com mais funcionalidades, mesmo com bugs.

Não deixe de discutir/dialogar sobre as perguntas com os demais alunos da sua mesa de trabalho.

Bom trabalho!