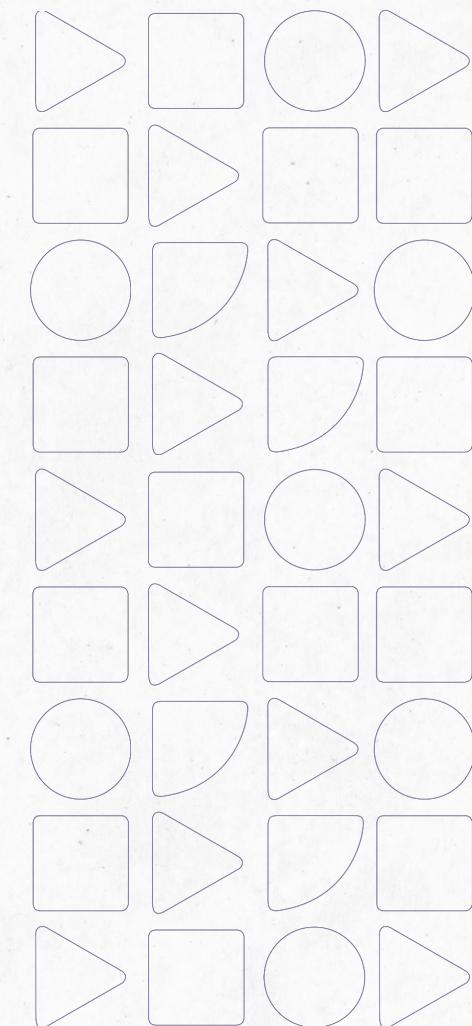


Diagnóstico

Disciplina: Qualidade de *Software*



Conteúdos:

Diagnóstico.

Habilidade(s):

Compreender como funcionam os testes estruturais em problemas aplicados.

Bloco 1

Modelo tradicional dos testes diagnósticos.

Com quem você mais se identifica?

Na hora de resolver um problema, quem é você?

1

Deito e durmo. Depois eu resolvo.

2

Sou ansioso. Até resolver, eu não consigo sossegar.

3

Tranquilo. Funciono bem sob pressão.

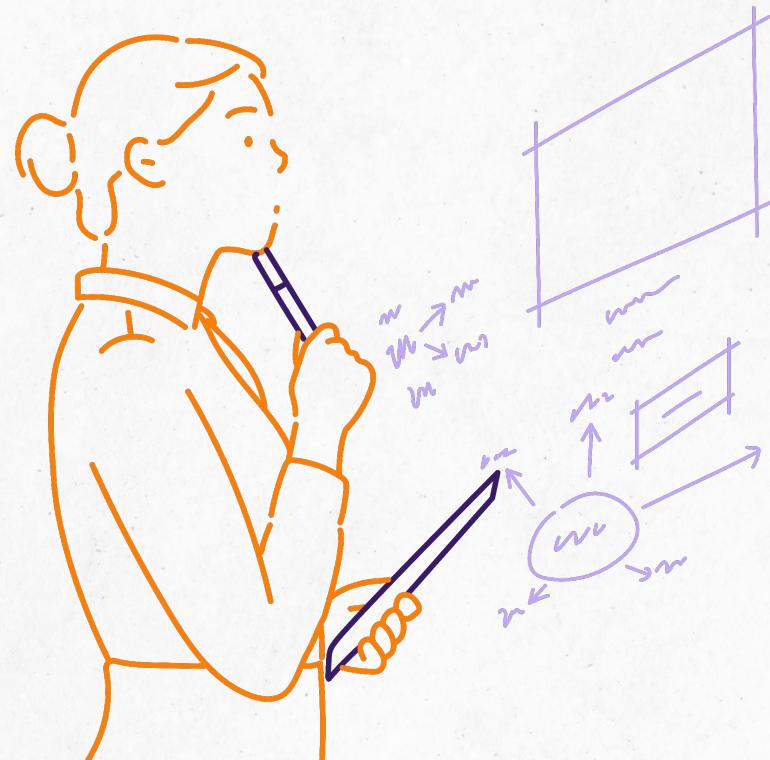
4

Sou criativo. Adoro resolver problemas!



Quando estamos gripados, é comum procurarmos um médico. Quando estamos passando por problemas pessoais, um psicólogo é uma boa saída.

Para vários problemas do dia a dia, sabemos qual solução adotar. No entanto, quando o assunto é teste de qualidade de software, isso pode ser mais difícil.



Testes em *software*

O teste é uma etapa essencial no desenvolvimento de *software*, mas nem sempre o seu processo é fácil.

Geralmente, esse tipo de teste é realizado em duas etapas essenciais.

1 Encontrar a causa raiz do *bug*;

2 Corrigir os componentes defeituosos, como as funções ou a classe.

Processo tradicional

Testador

Executa um conjunto de testes

Descobre o *bug*

Faz um relatório de *bug* de arquivo

Desenvolvedor

Identifica onde está o *bug*

Conserta o *bug*

Bloco 2

Modelos alternativos para o processo diagnóstico.

Modelo tradicional de diagnóstico na prática

Imagine que uma empresa depende de um sistema de gerenciamento de estoque para rastrear os seus produtos. No entanto, nas últimas semanas, os funcionários começaram a perceber problemas no sistema...

Eles observaram que as informações sobre os níveis de estoque estavam frequentemente imprecisas, levando a erros de previsão e falta de produtos.

A situação piorou quando o sistema começou a travar com frequência, impedindo os funcionários de acessar informações críticas.

Os funcionários também relataram lentidão no sistema e tempos de resposta muito longos ao tentar realizar tarefas simples.

Como esse *bug* seria identificado no modelo tradicional?



Desafios do teste de software

Encontrar a raiz da causa é desafiador e envolve o sistema de **tentativa e erro**. Aqui, o programador sugere diagnósticos possíveis e conduz testes para determinar o correto. Entretanto, existe uma dificuldade em reproduzir *bugs* identificados por testadores.

O que fazer, então?



Uma saída

A solução é que, durante a identificação de um *bug*, o testador realize **testes adicionais**, ajudando o programador a encontrar a causa do problema.



Porém, a maioria dos testadores é composta por profissionais de Garantia de Qualidade (QA) e eles não conhecem bem o código desenvolvido.

A boa notícia é que um método foi proposto
para lidar com esse problema.
Confira a seguir!



Processo proposto

Testador

Executa um conjunto de testes

Descobre um *bug*

Cria um relatório de *bug* de arquivo

IA

Executa um algoritmo de diagnóstico de IA

Produz um conjunto de diagnósticos possíveis

Planeja um teste para podar os diagnósticos falsos

Desenvolvedor

Conserta o *bug*

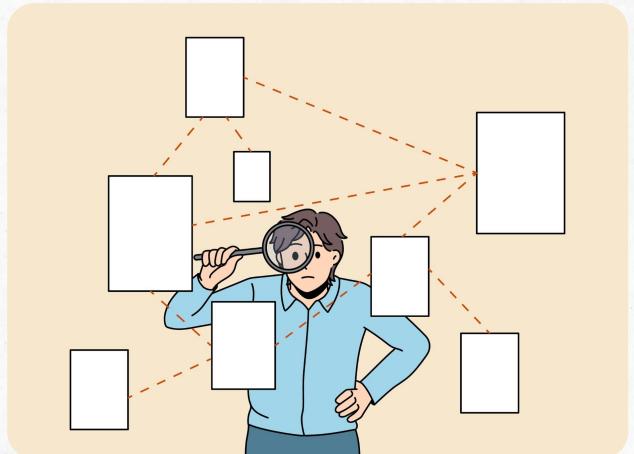
Bloco 3

Compreendendo o cálculo do nível de aproximação.

Vamos recapitular?

Analise a situação a seguir e diga quais seriam as principais contribuições do novo processo proposto.

Considere que um *software* de processamento de texto amplamente utilizado, como o Microsoft Word, possui um *bug* específico. Suponha que ele cause a perda ocasional de formatação de texto ao salvar documentos em determinados formatos, como em PDF.





Como seria a contribuição do novo modelo proposto?

Testador

IA

Desenvolvedor

Alguns colaboradores

Na década de 1990, Les Irwig e Paul Glasziou trabalharam na revisão sistemática da precisão de testes de diagnóstico.

Eles fundaram o **Grupo de Métodos de Teste de Rastreamento e Diagnóstico da Colaboração Cochrane** em 1996, com o objetivo de preparar revisões de precisão de testes diagnósticos.



Alguns colaboradores

Inicialmente, eles planejavam incluir essas revisões na Biblioteca Cochrane, mas, por limitações de recursos, a inclusão foi adiada até 2008.

O grupo desenvolveu métodos padronizados, além de softwares. Eles também colaboraram com especialistas para criar revisões e orientações.

Isso levou à publicação da primeira revisão de precisão de teste diagnóstico, datada de 2008.



Para você, o que não poderia faltar em um teste diagnóstico?



Compartilhe os seus
palpites!

Como era a metodologia inicial?

Nos anos finais da década de 1980 e no começo dos anos 1990, começaram a ser feitas análises que juntavam informações de vários estudos sobre os testes diagnósticos.

Isso foi feito de forma parecida com o que já se fazia com as análises de **intervenções médicas**: coletando e escolhendo estudos, avaliando a qualidade deles, resumindo os resultados em um só lugar (meta-análise) e tirando conclusões a partir disso



??



A análise da precisão dos testes diagnósticos costumava ser **complicada**. Isso acontecia porque essas análises envolviam várias medidas diferentes, como a sensibilidade e a especificidade ou os valores preditivos positivos e negativos.

Para entender o quanto bom era um teste, era preciso considerar várias informações.

Um método alternativo

Um método que ajudou a lidar com esse problema foi desenvolvido por Moses, Littenberg e outros colegas. Método esse que levou em conta as variações nos pontos de corte, tornando-se mais fácil de usar.



A pergunta certa a fazer

Fazer essas análises não envolve apenas a parte estatística.



Qual é o melhor jeito de fazer esse tipo
de estudo?

Não existe uma linguagem padrão

Isso torna mais difícil a busca por estudos relevantes em bancos de dados e a escolha de quais devem ser utilizados na análise.



Bloco 4

Investigando a formulação de perguntas e a interpretação dos resultados.

Hora do quiz

O teste muito preciso, por si só, não garante melhora nos resultados.



Verdadeiro.



Falso.

Hora do quiz

O teste muito preciso, por si só, não garante melhora nos resultados.



Verdadeiro.



Falso.

O teste por si só garante resultados?

Na verdade, o que realmente importa é como o profissional lida com os resultados do teste e como toma decisões baseadas neles.



O valor dos testes

Depende muito de como eles são utilizados em diferentes situações.

É válido se perguntar: essas descobertas valem para todos os casos ou as situações diferentes podem afetar os resultados do teste?



Vamos observar o exemplo a seguir

Um questionário usado para identificar se pessoas idosas estão desenvolvendo demência pode ser útil em geral.

No entanto, se esse questionário for usado em um lugar onde os pacientes já possuem vários sintomas em comum, pode não ser possível distinguir alguém com problemas cognitivos gerais de alguém com demência.



Os testes não são usados sozinhos

Por isso, as revisões do Cochrane focam em entender o valor relativo dos testes, ou seja, o quanto eles são melhores ou piores em comparação com as outras opções.

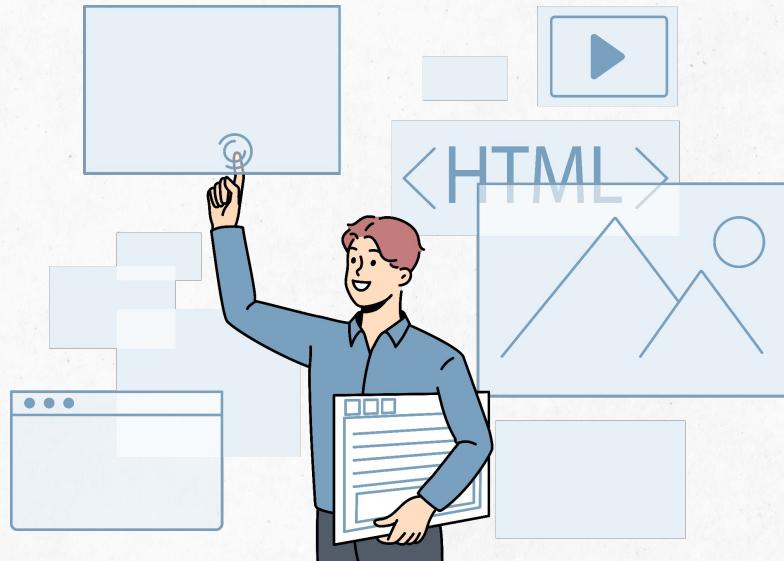


Bloco 5

Vamos explorar o Selenium?

Vamos conhecer o Selenium?

Selenium é uma suíte de ferramentas de código aberta, amplamente utilizada para a automação de testes em aplicativos da web.





Navegue pelo conhecimento!



Hora de compreender como o Selenium funciona.

Bloco 6

Vamos praticar o que vimos até aqui?

Hora de pôr a mão na massa!

Agora, vocês colocarão em prática tudo o que aprenderam!

- Criem um *script* de teste Selenium simples, que abra um navegador da *web*. Depois, accessem um *site* da *web* e executem uma ação básica, como clicar em um botão ou preencher um formulário.
- Escolham um *site* da *web* real ou uma página de teste e criem *scripts* Selenium para testar as diferentes funcionalidades do *site*.
- Introduzam *bugs* propositalmente no *site* ou na página de teste. A partir disso, criem um relatório.



Fechamento

Que bom!

Que pena...

Que tal?

Referências Bibliográficas

PROZ EDUCAÇÃO. *Apostila de Qualidade de software*. 2023.