
DOCUMENTANDO TESTES

RECAPITULANDO

- Para realizar de forma assertiva o desenvolvimento de software, além de codificar, é necessário testar o software
- Os testes podem ser classificados em algumas categorias
 - Unitários
 - De sistema
 - Aceitação

ESTRUTURANDO OS TESTES

- Realizar os testes tendo que criar muitas vezes, em conjunto com o programa pode ser exaustivo.
 - Desta forma, geralmente os times de desenvolvimento são divididos,
 - Uma equipe codifica.
 - Uma equipe elabora roteiros e testa o software
 - Assim, o tempo é otimizado.
-

ESTRUTURANDO OS TESTES

- Para otimizar o procedimento de teste podemos criar roteiros
 - Estes roteiros podem conter sugestão de entrada
 - Saída esperada
 - Sequência de passos a serem executados
- Desta forma, o teste é facilmente reproduzido por qualquer pessoa.

CRIANDO PLANO DE TESTES

- Não há um modelo de plano predeterminado
 - Depende da metodologia usada pela empresa.
 - Ou ainda da sua metodologia
- Um bom ponto de partida é o modelo enviado no grupo.
 - Um plano de teste para especificar como o ambiente de teste deve ser conduzido
- Um modelo que pode ser adotado é o exemplificado abaixo:
 - [Modelo de plano de teste.pdf](#)

CASOS DE TESTE

- Os casos de teste provêm uma especificação detalhada dos testes a serem realizados
 - Entradas, saídas, comportamentos esperados
- Dessa forma, busca-se mapear o comportamento do software, como ele deve se comportar ao ser realizada cada ação.
- Um modelo que pode ser adotado é o abaixo:
 - [Modelo de caso de teste.pdf](#)

RELATÓRIO DE TESTE

- Após a execução dos testes, se faz necessário criar uma forma de registrar os resultados, e repassar para a equipe de desenvolvimento efetuar as correções
- O instrumento utilizado pode ser o relatório de teste.
 - Pode ser digital(issues no github, bugzilla,etc)
 - Pode ser formalizado em estrutura própria.
- Um modo de registrar é o conforme o modelo de relatório do teste abaixo:
 - [Modelo de relatório de teste.pdf](#)

ABORDAGENS DE TESTES

- Dependendo o objetivo, os testes podem ter diferentes abordagens
 - Caixa branca
 - Caixa preta
 - Caixa cinza

TESTE CAIXA BRANCA

- Este tipo de teste foca na parte estrutural do sistema
 - Lógica, arquitetura do sistema
- O principal objetivo é encontrar problemas ocultos no código
 - Fluxos, condicionais
 - Segurança

TESTE CAIXA BRANCA

- Exige um conhecimento mais aprofundado de implementação
- Pode ajudar a tornar o código mais eficiente.
- Utilizando em fases de teste unitários e testes de integração.

TESTE CAIXA PRETA

- Estes testes, por sua vez, focam no aspecto comportamental do software
- O teste é encarado a partir do POV do usuário final
 - Use como um usuário faria
- Não é necessário conhecimento de programação, arquitetura de sistemas ou congêneres.

TESTE CAIXA PRETA

- É necessário conhecimento dos requisitos do sistema
 - Funcionais e não funcionais
- Foca em encontrar erros nas partes de:
 - Funcionalidade
 - Interface
 - Fluxo de uso do programa

TESTE CAIXA CINZA

- Uma abordagem hibrida entre o testes de caixa preta e branca
 - Nele, o testador tem conhecimento parcial da arquitetura do sistema
 - Foca em encontrar defeitos devido a um uso inadequado das estrutura/recursos
-

SITUAÇÃO PROBLEMA

- Uma equipe de desenvolvimento de software, com o objetivo de entender melhor as preferências de sua comunidade de usuários, decidiu criar uma enquete online simples. A ideia é descobrir qual a linguagem de programação favorita entre três opções populares: PHP, Python e JavaScript.