

## TESTE DE SOFTWARE

### Níveis de teste de sistema:

- Teste de unidade - 1
- teste de integração - 2
- teste de sistema - 3
- teste de aceitação - 4
- teste Alfa - 5
- teste Beta - 6
- Teste de Regressão - 7

Teste de unidade: Feito pelo programador, o teste de unidade tem como objetivo encontrar falhas em uma pequena parte do sistema funcionando (métodos, classes, I.E), independente do todo. Geralmente é automatizado e os principais programas usados são: Junit, PHPUnit, XXXUnit

Teste de Integração: Tem como objetivo, testar a comunicação entre os componentes de um sistema, é feito pelo programador. Geralmente é automatizado e os principais programas usados são: Junit, PHPUnit, XXXUnit. É feito antes do sistema estar concluído.

Teste de sistema: Tem como objetivo testar como se fosse o usuário final, procurando falhas. É planejado e executado pela equipe de teste de acordo com os requisitos especificados para o sistema. Feito após a conclusão da codificação do sistema.

Teste de aceitação: Tem como objetivo testar pelo usuário final (poucos), para saber qual a resposta dele, se ele aceita ou não o sistema.

Teste Alfa: Tem como objetivo liberar o sistema para uma pequena quantidade de usuários finais (entre 50 a 100) para receber feedback direto do usuário final. A equipe precisa estar presente em tempo real para resolver eventuais problemas reportados.

Teste Beta: É feito por muitos usuários finais, porém um subconjunto do usuário final. Por

exemplo: um jogo em teste beta liberado apenas para streamers, é um pré lançamento e os próprios usuarios reportam erros.

Teste de regressão: Pode ser feito a qualquer momento. Tem como objetivo refazer testes já feitos após atualização/alteração de alguma parte do sistema.

Teste estático: analisar o código sem executá-lo, teste "Caixa branca", o FindBugs é a ferramenta mais utilizada para automatizar esse teste.

Teste dinâmico: Executa o sistema para o teste

jmeter: programa para testar desempenho de sistemas. Da pra criar um script de como um usuário final usaria o sistema.

Ferramentas para gerenciar falhas encontradas durante o desenvolvimento de software:

- Proj4me, JIRA, Bugzilla, Trac