

**INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**
RIO GRANDE DO SUL
Campus Feliz

Teste de Software

Verificação e Validação

- Processo de verificar e validar que o software/programa atende aos **requisitos de negócio** e **técnicos** que guiaram sua arquitetura e desenvolvimento e trabalha como **esperado pelo cliente**.
- Verificação X Validação
 - **Verificação:** “Estamos construindo certo o produto?”
 - **Validação:** “Estamos construindo o produto certo?”
- Duas Abordagens complementares
 - **Inspeção de software (ou teste estático)**
 - **Teste de software**



Teste de Software

- O teste de software tem duas metas distintas:
 - Demonstrar ao desenvolvedor e ao cliente que o software atende aos requisitos;
 - Descobrir falhas/defeitos no software que apresentam comportamento incorreto, não desejável ou em não conformidade com sua especificação.



Erros de Software Recentes

Bank of Queensland – Austrália (2010)

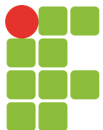


Uma rotina incorreta na conversão de um número hexadecimal, que deveria considerar o ano de 2010 pulava 6 anos até 2016, fazendo com que os terminais recusassem os cartões por estarem expirados.

Amazon (2011)



Erro de software dificultou a recuperação de um Data Center da Amazon após uma queda de energia. O sistema considerou que alguns blocos de dados não estavam sendo utilizados e os deletou. Na época, a Amazon se comprometeu em avisar aos afetados assim que uma nova cópia dos dados estivesse disponível.



Erros de Software Recentes

TRENDnet (2012)



Milhares de webcams se tornaram públicas por erro de codificação, o qual acredita-se que começou a partir de uma atualização no *firmware* em 2010. Para que a situação fosse corrigida, foi necessária uma nova atualização no *firmware*.

Gol (2015)



Falha no sistema causou a queda do valor da cotação do dólar americano, diminuindo muito o preço das passagens. Com isso foi possível comprar passagens aéreas internacionais (Orlando-EUA; Caracas-Venezuela; entre outros) por menos de R\$100. Muitos clientes realizaram a compra e a GOL arcou com o prejuízo.



Erros de Software Recentes

YouTube (2014)



Foi criado com 32 bits para armazenar a contagem de visualizações de vídeo.

Vídeo Gangnam Style (do coreano Psy) ultrapassou o limite dos 32 bits. As visualizações passaram para valores negativos.

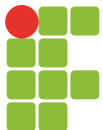


YouTube

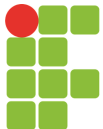
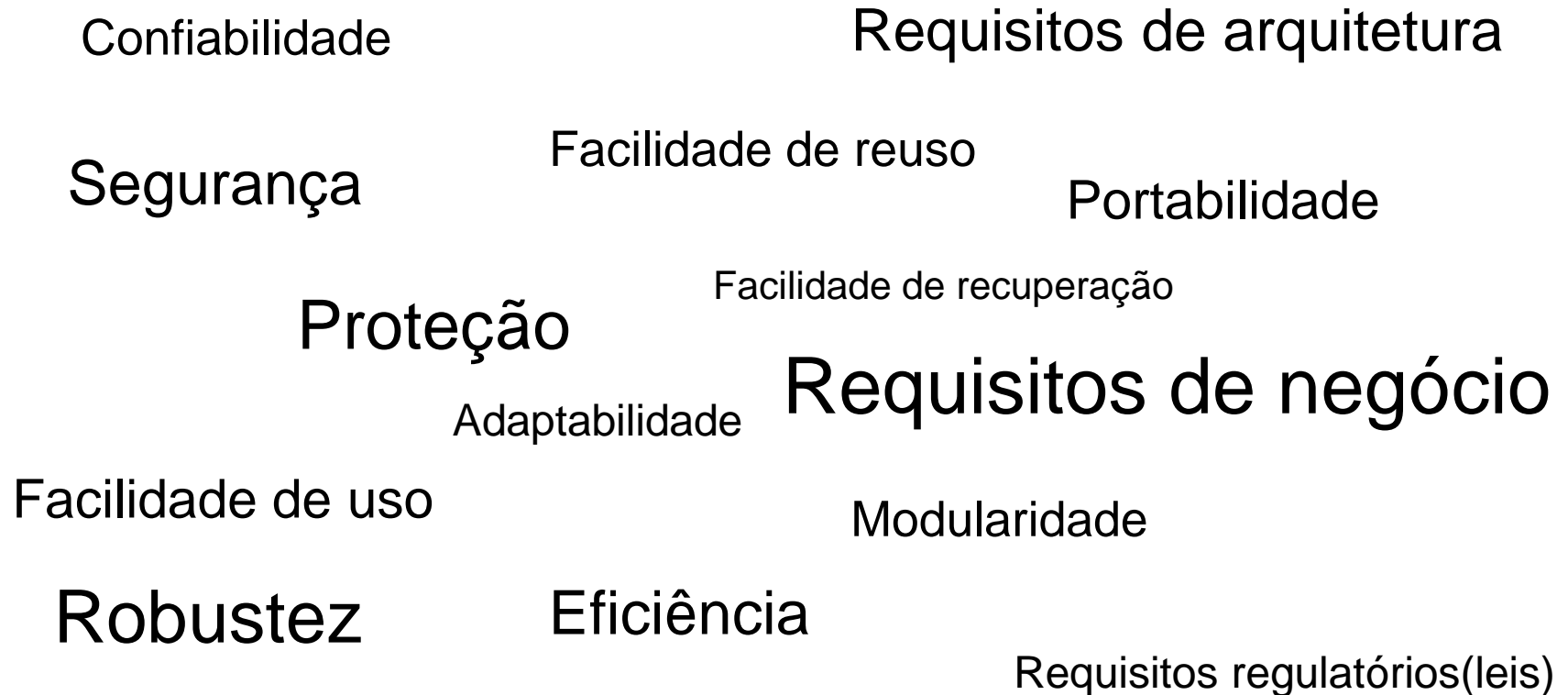
Shared publicly - Dec 1, 2014

We never thought a video would be watched in numbers greater than a 32-bit integer (=2,147,483,647 views), but that was before we met PSY. "Gangnam Style" has been viewed so many times we had to upgrade to a 64-bit integer (9,223,372,036,854,775,808)!

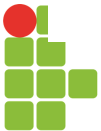
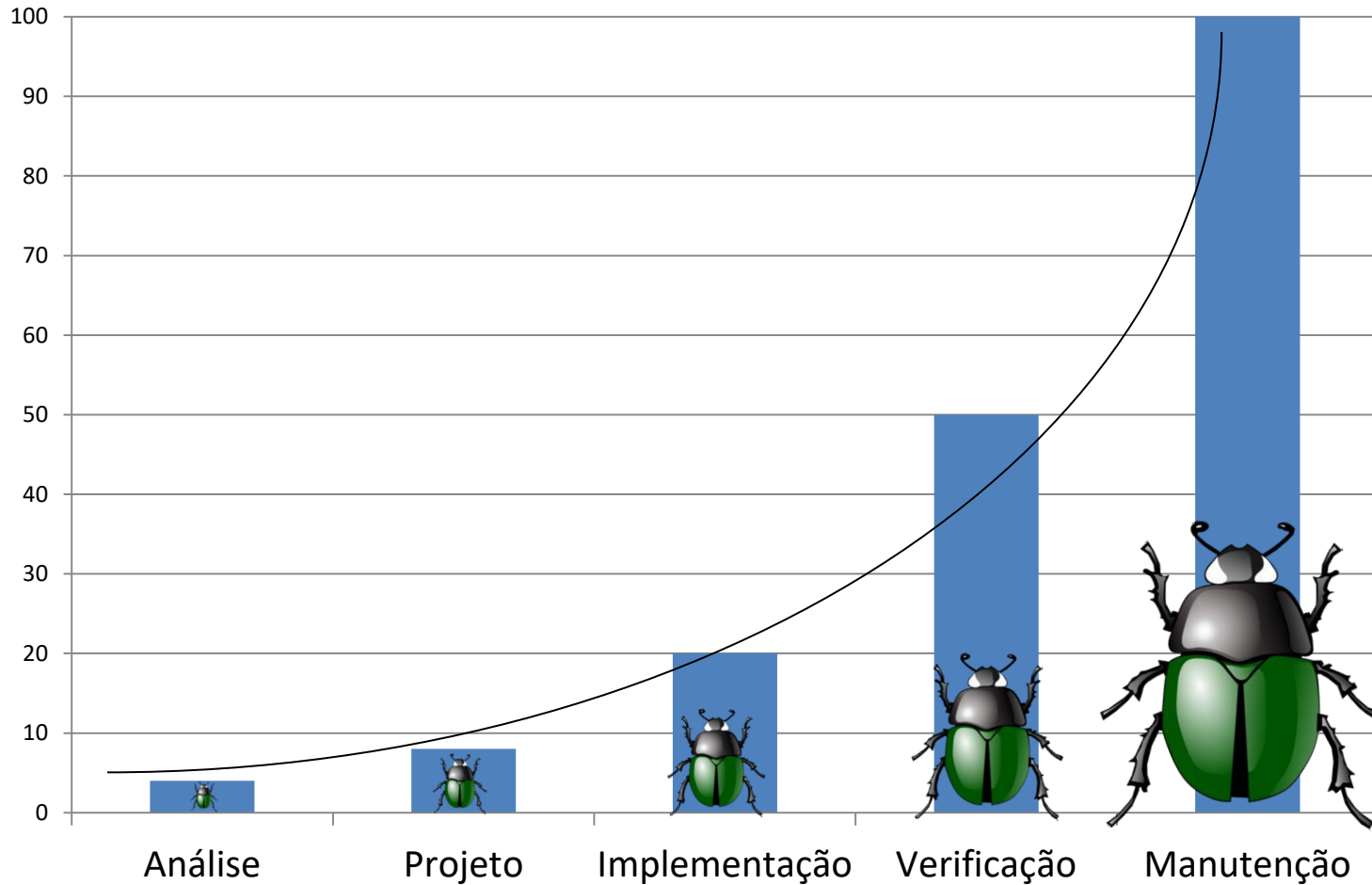
Hover over the counter in PSY's video to see a little math magic and stay tuned for bigger and bigger numbers on YouTube.



O quê testar?

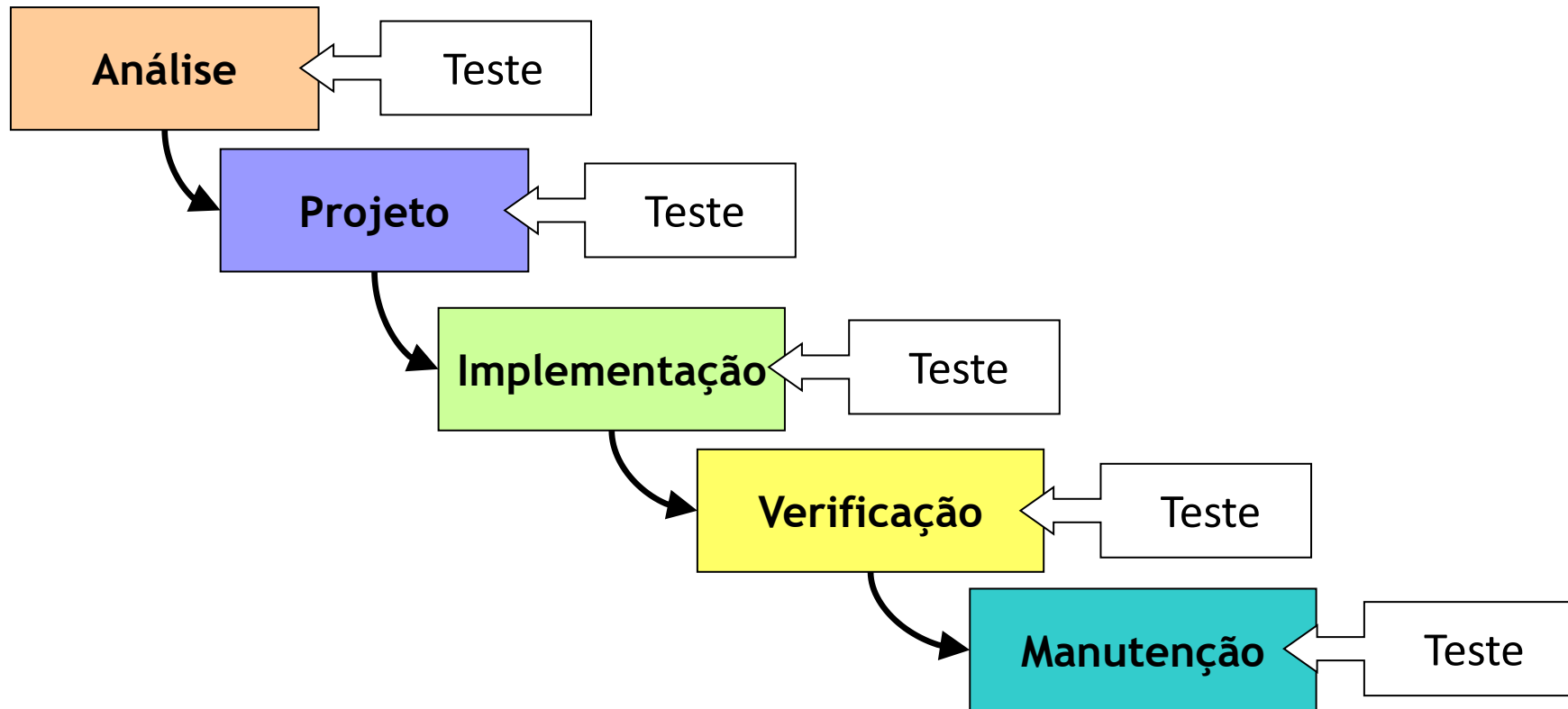


Custo dos defeitos por etapa do desenvolvimento



Quando testar?

- Modelo de desenvolvimento Cascata tradicional



Níveis de Teste de Software

- Teste de unidade
 - Módulos, funções
- Teste de integração
 - Interfaces entre módulos
- Teste de sistema
 - Busca falhas simulando o usuário final
- Teste de aceitação
 - Testes realizados por um grupo de usuários do sistema



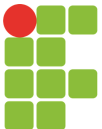
Tipos de Teste de Software

- **Funcional** (caixa-preta)
 - Objetivo: garantir que o sistema atende aos requisitos solicitados.
- **Estrutural** (caixa-branca)
 - Objetivo: avaliar o comportamento interno do componente de software. Teste de condição, fluxo de dados, ciclos e teste de caminhos lógicos.
- **Performance**
 - Objetivo: verificar tempo de resposta, carga, estabilidade, utilização de recursos (memória, CPU). Utiliza ferramentas específicas.
- **Segurança**
 - Objetivo: verificar acesso, segurança de dados, *firewall*, criptografia.
- **Usabilidade**
 - Objetivo: verificar se o sistema é fácil de acessar e se tem boa usabilidade.



Testabilidade de Requisitos

- Um requisito é testável se for possível criar um teste objetivo que possa determinar se o sistema atende ao requisito.
 - Claro, conciso, finito, mensurável, consistente, não ambíguo.
 - “deve processar os pedidos rapidamente”
 - “deve processar 500 pedidos a cada 20 segundos”
- O teste deve demonstrar ao seu final que o sistema atende aos requisitos.



Testabilidade de Requisitos

- Palavras vagas
 - *Ex.: gerenciar, processar*
- Palavras envolvendo tempo
 - *Ex.: depois, antes, quando, simultaneamente*
- Suposições e comparações
 - *Ex.: o sistema Z deve processar 15% a mais de volume do que o sistema Y*
- Conectivos
 - *Ex.: “ou”*



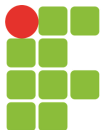
Testabilidade de Requisitos

- Adjetivos e Advérbios
 - *Ex.: robusto, efetivo, flexível, amigável*
 - *Ex: sempre, nunca*
- Palavras e acrônimos não específicos
 - *Evitar etc., e/ou*
- Pronomes indefinidos
 - *Ex.: alguns, muitos, poucos, maioria*
- Na revisão procure também por: ambiguidades, inconsistências, omissões, requisitos negativos, acrônimos não especificados



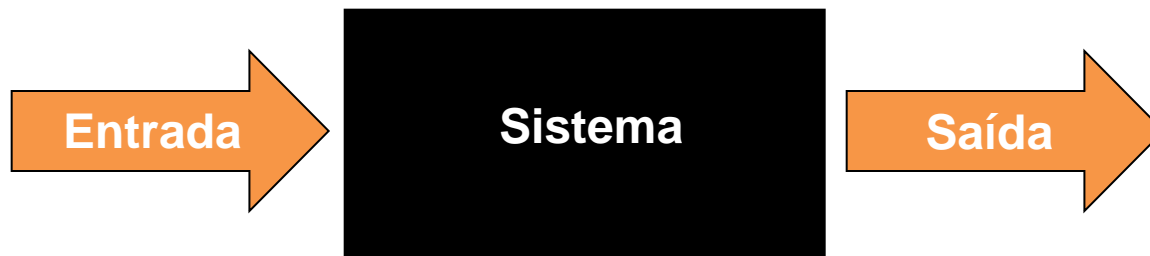
Tipos de Teste de Software

- **Funcional** (caixa-preta)
 - Objetivo: garantir que o sistema atende aos requisitos solicitados.
- **Estrutural** (caixa-branca)
 - Objetivo: avaliar o comportamento interno do componente de software. Teste de condição, fluxo de dados, ciclos e teste de caminhos lógicos.
- **Performance**
 - Objetivo: verificar tempo de resposta, carga, estabilidade, utilização de recursos (memória, CPU). Utiliza ferramentas específicas.
- **Segurança**
 - Objetivo: verificar acesso, segurança de dados, *firewall*, criptografia.
- **Usabilidade**
 - Objetivo: verificar se o sistema é fácil de acessar e se tem boa usabilidade.



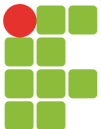
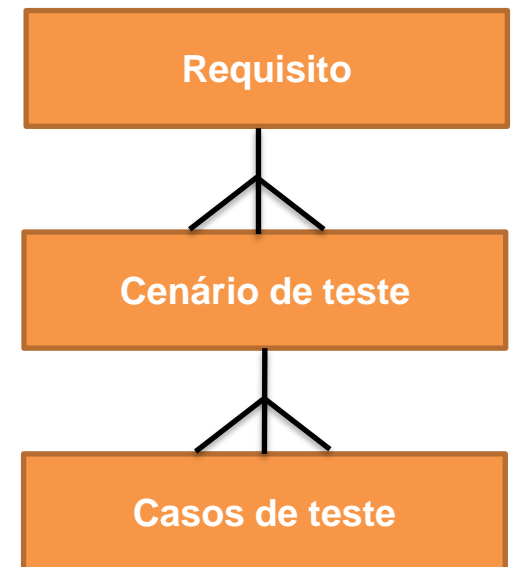
Teste Funcional

- É baseado na especificação funcional do sistema.
 - O sistema é testado como se fosse uma caixa-preta.
- Visa garantir que o sistema faz o que deveria fazer.
- Requer requisitos completos, detalhados, sem ambiguidade.



Teste Funcional - Cenários e Casos de Teste

- Um **cenário de teste** representa uma situação que poderia acontecer quando o sistema estiver em produção.
- **Caso de teste** é um passo a passo detalhado para simular determinado cenário.



Cenários e Casos de Teste - Exemplo

Domínio: Loja Virtual

- **Requisito:** O sistema deve permitir que o usuário verifique o *status* de seus pedidos.
- Possíveis cenários de teste:
 - Verificar o *status* de uma ordem que não existe
 - Verificar o *status* de uma ordem paga
 - Verificar o *status* de uma ordem cancelada

Exemplo 1 de Caso de teste: Verificar o *status* de uma ordem paga.

1. Abra o Navegador
2. Navegue para "<http://www.lojavirtual.com>"
3. Clique no link "Logar"
4. Preencha os campos "Usuário e Senha"
5. Clique no botão "Entrar"
6. Clique no link "Meus Pedidos"
7. Digite o número do pedido no campo "Pedido"
8. Clique no botão "Visualizar"
9. Verifique se o pedido está com status: "Pago"



Cenários e Casos de Teste - Exemplo

Exemplo 2 de Caso de teste: Verificar o *status* de uma ordem paga.

Passos	Resultado esperado
1. Abra o Navegador	Navegador aberto
2. Navegue para http://www.lojavirtual.com	Site da loja visualizado
3. Clique no link “Logar”	Tela de login visualizada
4. Preencha os campos “Usuário e Senha”	Campos preenchidos
5. Clique no botão “Entrar”	Logado no sistema
6. Clique no link “Meus Pedidos”	Tela de pedidos visualizada
7. Digite o número do pedido no campo “Pedido”	Número digitado
8. Clique no botão “Visualizar”	Tela com informações do pedido visualizada
9. Verifique o status do pedido	Status do pedido está pago



Cenários e Casos de Teste - Exemplo

Exemplo 2 de Caso de teste: Verificar o *status* de uma ordem paga.

Passos	Resultado esperado	Resultado da execução
1. Abra o Navegador	Navegador aberto	Ok
2. Navegue para http://www.lojavirtual.com	Site da loja visualizado	Ok
3. Clique no link “Logar”	Tela de login visualizada	Ok
4. Preencha os campos “Usuário e Senha”	Campos preenchidos	Preenchido com user: admin senha: 123
5. Clique no botão “Entrar”	Logado no sistema	Ok
6. Clique no link “Meus Pedidos”	Tela de pedidos visualizada	Ok
7. Digite o número do pedido no campo “Pedido”	Número digitado	Pedido utilizado: 21354
8. Clique no botão “Visualizar”	Tela com informações do pedido visualizada	Erro: sistema apresentou mensagem informando que pedido não existe.
9. Verifique o status do pedido	Status do pedido está pago	

Como Relatar um Defeito

- **Título:** deve fornecer uma ideia concisa e clara do problema encontrado;
- **Descrição:** deve conter uma explicação detalhada do problema;
- **Passos para reprodução do defeito:**
 - devem ser bem detalhados e sem ambiguidades e redundância;
 - não se deve assumir que determinados passos são óbvios;
 - se existem pré-condições para a reprodução da ocorrência, elas devem ser especificadas;
 - esse campo é utilizado para convencer de que o problema existe e é reproduzível.



Como Relatar um Defeito

Passos para reprodução:

1. Acessar a “Tela de Cadastro de Usuários”;
2. Não informar os campos obrigatórios;
3. Clicar em [Salvar].



Passos para reprodução:

“Acesse a tela de cadastro de usuário sem informar os campos obrigatórios e clique em salvar.”



Como Relatar um Defeito

- **Resultados esperados:** descrição do que é esperado após a execução dos passos;
- **Resultados obtidos:** descrição do que é obtido após a execução dos passos;
- **Anexos:** arquivos que possam auxiliar no entendimento do problema reportado (como *print de telas*);
- **Informações Adicionais:** detalhar informações adicionais que possam ajudar no entendimento do problema. Por exemplo, informar navegador, dispositivo móvel e sistema operacional utilizados.

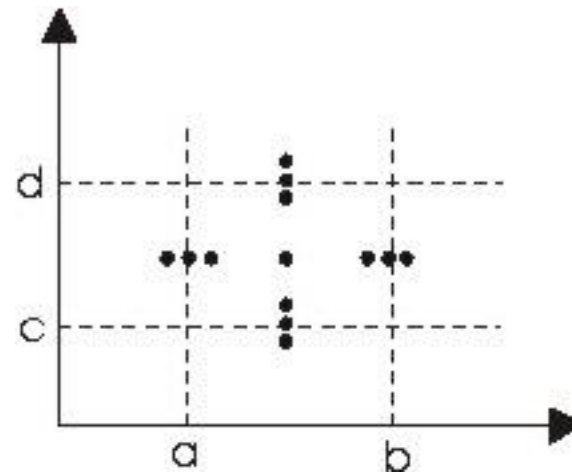
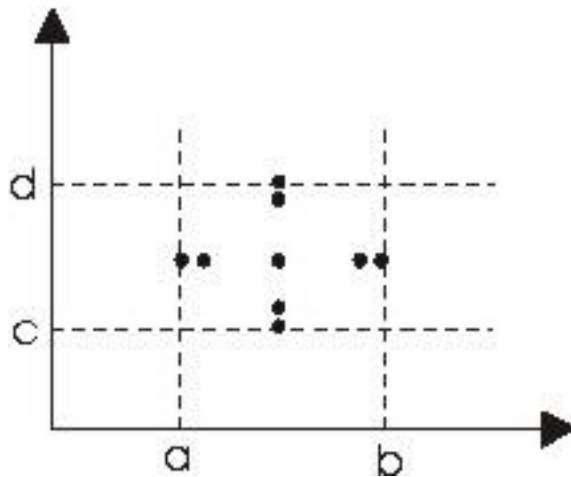


Técnicas de Teste - Análise do Valor Limite

- Parte da observação de que os erros costumam ocorrer próximos dos valores limites das variáveis de entrada.
- Idéia básica: usar os valores de entrada nos limites máximo e mínimo;

$$a \leq x1 \leq b$$

$$c \leq x2 \leq d$$



Técnicas de Teste - Classes de equivalência

- Pense em todos os possíveis valores para cada entrada;
- Crie grupos (classes de equivalência) e coloque neles valores que serão tratados da mesma forma pelo sistema;
- Se espera-se o mesmo resultado para duas entradas então usar apenas uma delas;
- Algumas classes serão válidas e outras inválidas

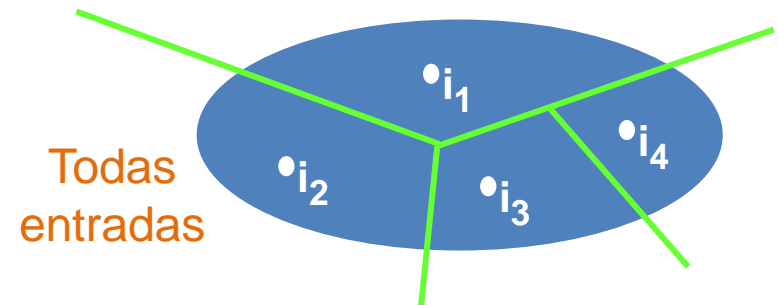
- Exemplo: Empréstimo bancário

- Entrada

- Avaliação do crédito b (entre 1 e 5)
 - Salário familiar

- Saída

- Se avaliação do crédito é 4 ou mais o empréstimo é aprovado
 - Senão se salário da família é R\$5.000 ou mais aprova o crédito
 - Senão rejeita empréstimo.



Técnicas de Teste - Classes de equivalência

Sistema para uma loja online para compra de jogos

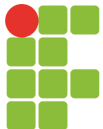
- **Cadastro de clientes:**
 - O sistema deve permitir o cadastro apenas de clientes com idade superior a 10 anos e inferior a 120 anos.
 - Os campos obrigatórios são: Usuário, Senha, CPF, Nome, Telefone e E-mail.
 - Os campos opcionais são: Endereço e Telefone comercial.
- **Compra de jogos:**
 - Alguns jogos possuem idade mínima para que o cliente possa comprar. Por exemplo:
 - O jogo "GOD OF WAR ASCENSION" tem idade mínima de 18 anos.
 - O jogo Mario Kart tem idade mínima de 10 anos.



Técnicas de Teste

- Objeto de aprendizagem sobre as técnicas de teste citadas:

http://www.inf.ufsc.br/~fabiane.benitti/byebug/objetos/OA25/presentation_html5.html



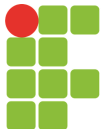
Automação de Testes

- Princípio básico: automatizar a execução dos testes manuais através de uma ferramenta automática, cobrindo o máximo de cenários possíveis, através da mudança de parâmetros
 - Funcional, regressão, performance, geração de dados
- Razões para automatizar: melhor qualidade do produto com aumento de cobertura, menos tempo de execução, defeitos encontrados mais cedo, reuso.
- Problemas:
 - Automatizar leva tempo
 - Custo inicial alto (pode consumir até 10 vezes mais tempo): adquirir ferramenta, aprender, desenvolver os scripts, dar manutenção
 - Automação é desenvolvimento/codificação
 - Falta de experiência
 - Falsas expectativas



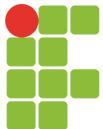
Automação de Testes – Quando fazer?

- Tarefas repetitivas como teste de regressão, teste superficial, funcionalidades mais usadas
- Carregamento/geração de dados, configuração do sistema
- Aplicação tem muitos *installs, patches*
- Aplicação precisa ser testada em diferentes hardware ou configurações de software



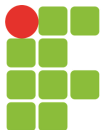
Automação de Testes – Quando **NÃO** fazer?

- Teste de usabilidade
- Funções não estáveis
- Processos indefinidos
- Testes que serão executados apenas 1 vez
- Testes complexos



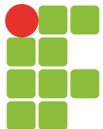
Ferramentas

- **Teste Unitário**
 - **Junit** - <http://www.junit.org/>
- **Gerenciamento de Teste**
 - **TestLink** - <http://www.teamst.org/>
 - **QAManager** - <http://qamanager.sourceforge.net/site/en/>
- **Execução de Testes Funcionais**
 - **Selenium** - <http://selenium.openqa.org/>
- **Gestão de Defeitos**
 - **Mantis** - <http://www.mantisbt.org/>
 - **BugZilla** - <http://www.bugzilla.org/>



Resumo – Teste de Software

- Requisitos completos e precisos são essenciais para um teste efetivo;
- Procure se envolver o mais cedo possível no processo;
- Descreva formalmente seus testes;
- Separe seus dados de teste dos casos de teste;
- Tenha certeza de que fez teste negativo suficiente;
- Acompanhe a execução dos testes para ter um relatório de status efetivo;
- Entenda como seus testes irão afetar os seus dados;
- Inclua análise de impacto em sua estratégia de testes de regressão;
- Defina teste como um processo e não como uma fase do ciclo de vida;
- Selecione ferramentas para suportar o seu processo;
- Esteja pronto para atender as reuniões de revisão do projeto da aplicação;
- Planeje ser um testador profissional e não apenas um testador.



Referências

- Bibliografia básica:

Sommerville, Ian. **Engenharia de Software**. São Paulo: Pearson Addison-Wesley. 2007, 8ª edição.

- Bibliografia complementar:

Pressman, Roger. **Engenharia de Software**. São Paulo: McGraw-Hill, 2006, 6ª edição.

Forouzan, B.; Mosharraf, F. **Fundamentos da Ciência da Computação**. São Paulo: Cengage Learning, 2011.

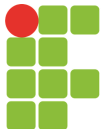
<http://www.softex.br/mpsbr/>

<http://www.devmedia.com.br/maturidade-no-desenvolvimento-de-software-cmmi-e-mps-br/27010>



Crowdsourced Testing

- *Crowdsourcing testing, is the practice of sending out prototype software and products to the expert testers instead of testing internally. It is also called crowd testing. Nowadays, it has become an alternative for traditional testing.*
- 13 Best Crowdsourced Testing Companies in 2018:
<https://www.guru99.com/crowdsourced-testing-companies.html>



Crowdsourced Testing - BugFinders

The screenshot displays the BugFinders website. The top navigation bar includes links for SIGN IN, BECOME A TESTER, ABOUT, PARTNERS, CONTACT US, SEE A DEMO, and a phone number. The main header features the BugFinders logo and a menu with SOLUTIONS, PRICING, SERVICES, TESTIMONIALS, CASE STUDIES, and RESOURCES. The main content area has a large section titled 'BUGFINDERS' with the tagline 'Digital testing at a speed and scale no other supplier can match'. It includes a video player showing various brand logos (VISA, Electron, Audi, Money Super Market, Harrods, ck Calvin Klein, claire's, Superdry, LVE) and a button to 'VIEW ALL OUR CUSTOMER SUCCESS STORIES'. Below this is a section for 'AUDI' with the text 'BugFinders' web and app testing guarantees conversion success' and a video player showing a man in a suit. A chat bubble on the right asks 'Have any questions about our website and app testing? We're happy to help!'.

BUGFINDERS

Digital testing at a speed and scale no other supplier can match

Ensure your applications work on all the devices your visitors use with BugFinders' rapid community testing service. Discover how we can deliver 24 days' worth of testing for you in just 24 hours.

[VIEW ALL OUR CUSTOMER SUCCESS STORIES](#)

AUDI

BugFinders' web and app testing guarantees conversion success

Have any questions about our website and app testing? We're happy to help!



Crowdsourced Testing - crowdTest

← → ↻ https://app.crowdtest.me

Cientes Planos Blog Contato Projetos aplemke

crowdtest

Crowdtest - Crowdsourcing de T... Watch later Share

O Crowdtest encontra centenas de bugs em poucas horas por menos da metade do custo.

Teste seu software agora!

Se você quer ser um testador, [clique aqui](#) para se cadastrar.

257 projetos testados | 9644 testadores participando | 7540 ocorrências encontradas | R\$ 300 valor médio pago aos melhores testadores

PROJETOS RECENTES As chamadas de novos projetos são divulgadas no Facebook

livelo	Projeto PaperX	ISVOR
Nívelo Nívelo-2 Este projeto é para encontrar BUGs em um novo Market Place que será lançado no mercado. O testador é livre para reali...	Projeto PaperX 10 segundos 1.11 Este projeto é para testar o Aplicativo 10 segundos da PaperX em seu ambiente de produção. O App 10 segundos é u...	Projeto Isvor Isvor Este projeto é para testar o game de agendamento da Isvor, em seu ambiente de homologação. Este projeto é na modal...
Inscrições encerradas PAUSADA	Inscrições encerradas PAUSADA	Inscrições encerradas PAUSADA

Komark uol Jogos info TNW InformationWeek

