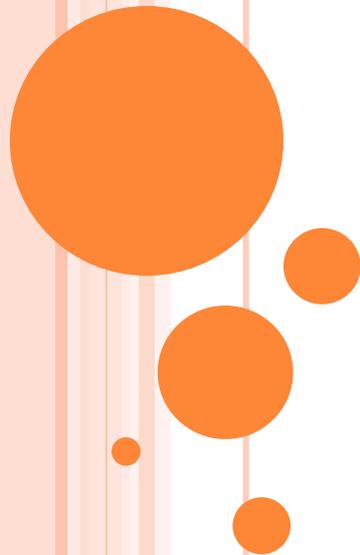


SSC5877 VALIDAÇÃO E TESTE DE SOFTWARE
SSC0721 INSPEÇÃO E TESTE DE SOFTWARE

Principais Desafios ao Teste de Software:

Simone Senger de Souza
srocio@icmc.usp.br

ICMC/USP
2020



O QUE JÁ SABEMOS SOBRE TESTE DE SOFTWARE?



MITO OU VERDADE?

1. Desenvolvedores (ou times) mais experientes produzem códigos melhores que os poucos experientes



MITO OU VERDADE?

1. Desenvolvedores (ou times) mais experientes produzem códigos melhores que os poucos experientes

- Priorização do que testar poderia levar em consideração a expertise da equipe de desenvolvimento.



MITO OU VERDADE?

2. Funções ou métodos mais complexos tem maior probabilidade de terem defeitos que os mais simples



MITO OU VERDADE?

2. Funções ou métodos mais complexos tem maior probabilidade de terem defeitos que os mais simples

- Priorização dos testes – utilizar métricas de complexidade para avaliar métodos e funções
 - Ex: McCabe



MITO OU VERDADE?

3. Aumentar a usabilidade de um sistema pode aumentar a complexidade do software



MITO OU VERDADE?

3. Aumentar a usabilidade de um sistema pode aumentar a complexidade do software

- O aumento de usabilidade aumenta o esforço da atividade de teste de software



MITO OU VERDADE?

4. Quanto mais um código é modificado, maior é a probabilidade de defeitos



MITO OU VERDADE?

4. Quanto mais um código é modificado, maior é a probabilidade de defeitos

- Mais mudanças significam menor entendimento do código e mais chances de enganos
- Priorizar a atividade de teste em partes que foram mais modificadas
 - Usar informações de número de commits do código



MITO OU VERDADE?

5. Tecnologias novas ou desconhecidas pela equipe levam a mais problemas de codificação



MITO OU VERDADE?

5. Tecnologias novas ou desconhecidas pela equipe levam a mais problemas de codificação

- Por não conhecer, a complexidade do software aumenta e, conseqüentemente, o esforço da atividade de teste também



MITO OU VERDADE?

6. Nem sempre é possível replicar o realismo do software em um ambiente de testes



MITO OU VERDADE?

6. Nem sempre é possível replicar o realismo do software em um ambiente de testes

- Priorizar testes em partes críticas, definindo um esforço maior pois é necessário fazer simulações do ambiente



MITO OU VERDADE?

7. Defeitos não são peixes



MITO OU VERDADE?

7. Defeitos não são peixes

- Defeitos se aglutinam em partes problemáticas do código
 - Identificar padrões de defeitos
- Pareto funciona: 80% dos defeitos estão em 20% do código
- Geração de dados de teste aleatória pode não ser a melhor saída



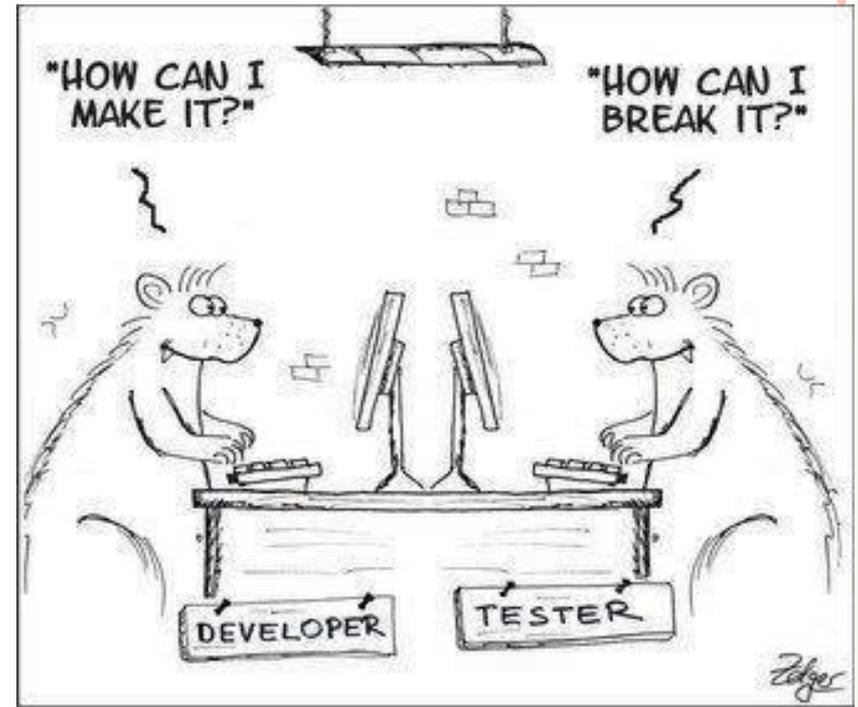
DESAFIOS PRINCIPAIS

- Automatização x criação
 - Aplicação de IA para resolver problemas de esforço na atividade de teste de software (geração de dados de teste)
 - Diferentes ferramentas de apoio ao teste de software
 - Tecnologias evoluem – teste de software precisa acompanhar
 - A criatividade e intuição do testador ainda prevalece
 - Uso de boas técnicas e critérios podem colaborar



DESAFIOS PRINCIPAIS

- Teste de software no processo de desenvolvimento
 - A atividade de teste é um processo por si só
 - Custos envolvidos
 - Comprometimento



They are not so much different,
but they have different path for the same goal,
to improve quality!!

DESAFIOS PRINCIPAIS

- Problema do oráculo
 - Especificação está certa? O modelo está certo?
 - Para alguns domínios não há um oráculo simples:
 - Ex: software de processamento de imagens



DESAFIOS PRINCIPAIS

- Testes para domínios específicos
 - Mobile applications, sistemas Web, software concorrente, distribuído, sistemas adaptativos ...
 - Defeitos específicos
 - Desafios novos a atividade de teste
 - Ex: observability error



DESAFIOS PRINCIPAIS



- Propriedade emergente: Segurança

- Funções de segurança x comportamento seguro

- Ex: função “desligamento de emergência”

- E se a função não funcionar quando deveria funcionar?

- Análise de “Suposições” sobre Segurança

- Ex: Ariane4

- Suposições foram feitas sobre velocidade e aceleração que o foguete iria experimentar, as quais foram usadas no código de manipulação de exceções

- Essas suposições foram “aproveitadas” para o Ariane5, porém não foram adequadas, levando a falhas e a explosão do Ariane5.

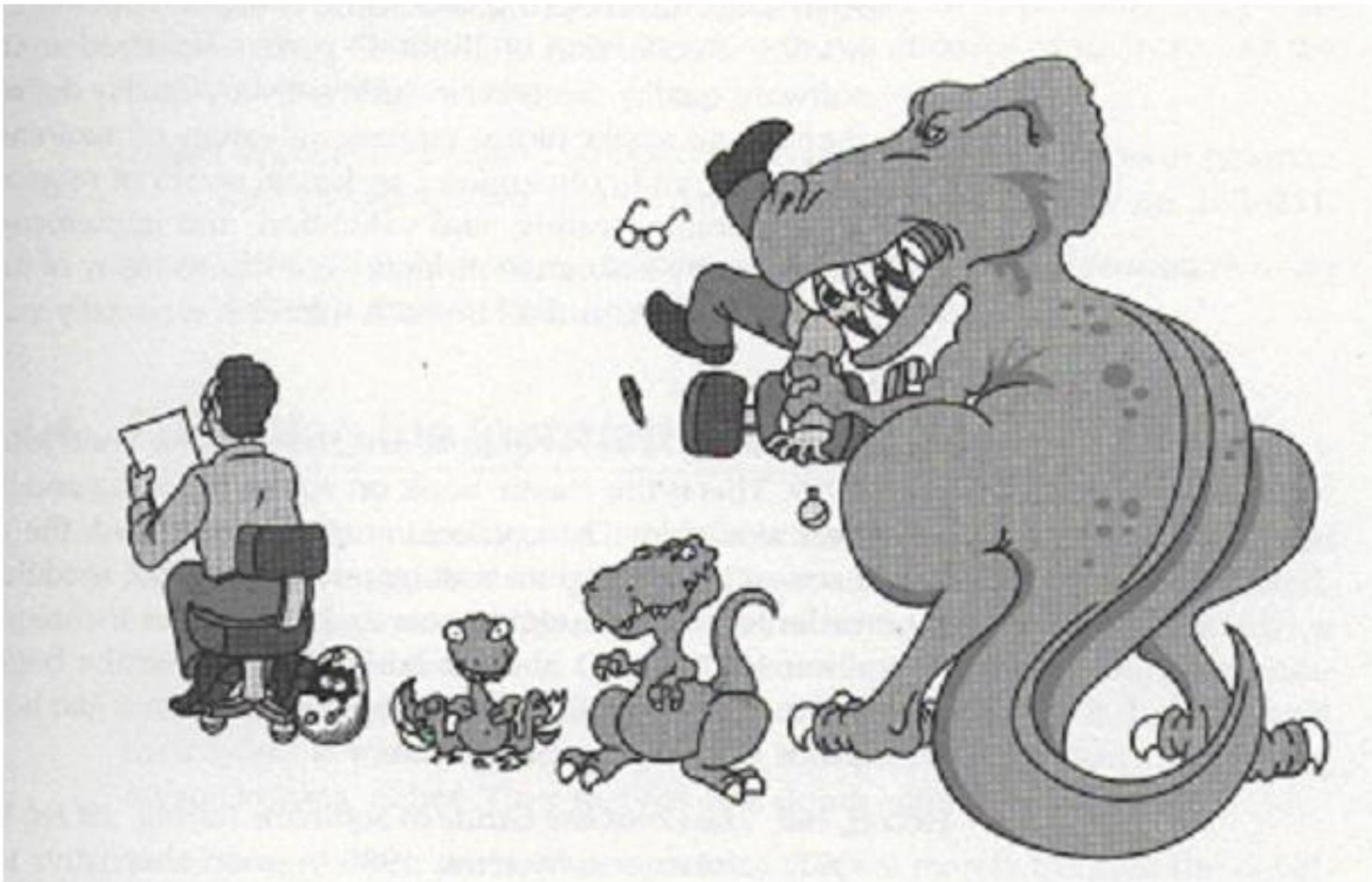
- **Conclusão:** as “suposições” são importantes na fase de teste de segurança, porém devem estar corretas!



ÚLTIMAS DICAS SOBRE SELEÇÃO DE BONS CASOS DE TESTE

1. Desenvolver casos de teste para usos não “desejados”: considerar “*Misuses Cases*”
2. Identificar suposições e criar casos de teste que violam tais suposições
3. Identificar questões sobre “configurações de componentes” e projetar testes para elas
Caso do Ariane5
4. Desenvolver entradas de teste inválidas
Use e abuse de critérios de teste funcionais





The longer defects remain undetected, the longer they take to fix . Correct defects when they are young and easy to control!

