FUNDAMENTOS DE TESTE DE SOFTWARE

Prof. Me. Rober Marcone Rosi

Introdução

- Definitivamente, a construção do software não é uma tarefa simples. Pelo contrário, pode se tornar bastante complexa, dependendo das características e dimensões do sistema a ser criado.
- Por isso, está sujeita a diversos tipos de problemas que acabam resultando na obtenção de um produto diferente daquele que se esperava.
- Muitos fatores podem ser identificados como causas de tais problemas, mas a maioria deles tem uma única origem: **ERRO HUMANO**.
- A construção de software depende principalmente da habilidade, da interpretação e da execução das pessoas que o constroem; por isso, defeitos acabam surgindo, mesmo com a utilização de métodos e ferramentas de engenharia de software (DELAMARO; MALDONADO; JINO, 2007, p. 1).

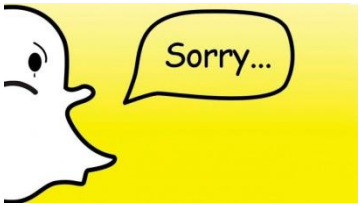
Erros Capitais:

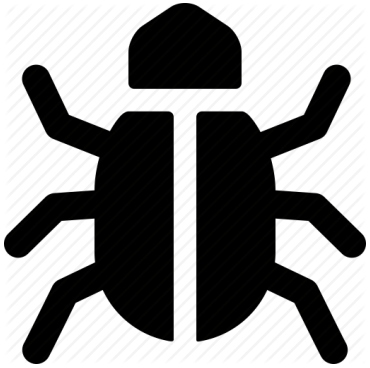
SUMÁRIO

- Termos
- Contexto dos sistemas de software
- Causas dos defeitos de software
- Verificação x Validação na Engenharia de Software
- Definição e objetivos do Teste de Software
- O ISTQB
- Função do teste no desenvolvimento, manutenção e operação de software
- Teste x qualidade
- Depuração de Código x Teste
- Os sete Princípios do Teste
- O Processo de Teste: Planejamento e controle; Análise e modelagem; Implementação e execução; Avaliação do critério de saída e relatório; Atividades de encerramento
- A Psicologia do Teste
- Código de Ética

ERRO X DEFEITO

- Erro e defeito têm significados diferentes e **não há consenso sobre sua definição na literatura.**





ERRO X DEFEITO

- Segundo Pressman (2016, p. 432):
  - Um **erro** é alguma falha encontrada **antes** do software ser entregue ao usuário final.
  - **Defeito** se refere a falhas encontradas **depois** da entrega do software ao usuário final.

ERRO X DEFEITO

- Segundo o ISTQB (2011):
  - **Erro (*error)*** ou **engano** (*mistake*)
    - Ação humana que produz um resultado incorreto (IEEE 610), ou seja, ação humana que produz um defeito (DELAMARO; MALDONADO; JINO, 2007, p. 2).

  - **Defeito (*defect*), *bug*** *(bug) ou* **falha** (*fault)*
    - Falha em um componente ou sistema que pode fazer com que o componente ou sistema falhe ao desempenhar sua função, por exemplo, uma sentença incorreta ou uma definição de dados incorreta. Um defeito, se descoberto durante a execução, pode levar à falha do componente ou do sistema.

CONTEXTO DOS SISTEMAS DE SOFTWARE

- Sistemas de software tornam-se cada vez mais parte do nosso dia-a-dia, desde aplicações comerciais (ex.: bancos) até produtos de consumo (ex.: carros).
- A maioria das pessoas já teve alguma experiência com um software que não funcionou como esperado.
- Softwares que não funcionam corretamente podem levar a muitos problemas, incluindo financeiro, tempo e reputação das empresas, podendo, inclusive, chegar a influenciar na integridade das pessoas.

CAUSAS DOS DEFEITOS DE SOFTWARE

- O ser humano está sujeito a cometer um erro/engano que produz um defeito/falha/bug no código, em um software ou sistema ou em um documento.

- Se um defeito no código for executado, o sistema falhará ao tentar fazer o que deveria (ou, em algumas vezes, o que não deveria), causando uma falha. Defeitos no software, sistemas ou documentos resultam em falhas, mas nem todos os defeitos causam falhas.

- Os defeitos ou falhas ocorrem porque:
  - os seres humanos são passíveis de falha;
  - existe pressão no prazo, códigos complexos, complexidade na infraestrutura, mudanças na tecnologia e/ou muitas interações de sistema;
  - podem ser causadas por condições do ambiente tais como: radiação, magnetismo, campos eletrônicos e poluição, que podem causar falhas em software embarcado (firmware) ou influenciar a execução do software pela mudança das condições de hardware.

O QUE É TESTE DE SOFTWARE?

- É uma atividade da Engenharia de Software que representa a revisão final da **especificação**, **projeto** e da **geração de código**, ou seja, teste não é feito somente no código fonte mas também deve ser feito no Documento de Requisitos, na Especificação do Projeto, dentre outros (PRESSMAN, 2016).

O QUE É TESTE DE SOFTWARE?

- Teste não mostra ausência de defeitos, mas apenas mostra que defeitos de software estão presentes.
- Não se pode provar que um programa nunca irá falhar; ao contrário, pode-se provar, apenas, que ele contém defeitos.
- Sendo assim, **uma visão comum errada** é “um teste bem-sucedido é aquele no qual NÃO SÃO encontrados defeitos”.
- Um bom caso de teste é aquele que tem alta probabilidade de encontrar um defeito ainda não descoberto. Um teste bem-sucedido é aquele que revela um defeito ainda não descoberto.

O QUE É TESTE DE SOFTWARE?

- O Teste de Software é uma das atividades de garantia da qualidade de software, junto com as atividades de validação e verificação (MALDONADO, 2001, p.73).

- Sendo assim, vamos revisar os termos
  - Qualidade de software
  - Garantia da qualidade de software
  - Verificação e validação na Engenharia de Software

QUALIDADE DE SOFTWARE

- No contexto da Engenharia de Software, a **Qualidade de Software** é definida como a conformidade (PRESSMAN, 2002, p193):
  - com requisitos funcionais e de desempenho explicitamente declarados,
  - com padrões de desenvolvimento explicitamente documentados e
  - com características implícitas que são esperadas de todo software desenvolvido profissionalmente.

GARANTIA DA QUALIDADE DE SOFTWARE

- A **Garantia de Qualidade de Software** (SQA – *Software Quality Assurance*) é uma atividade de apoio que é aplicada ao longo do processo de software.

- **É a atividade de fornecer para todos os interessados a evidência necessária para estabelecer confiança em que a função de qualidade está sendo cumprida adequadamente. É o conjunto de atividades sistemáticas fornecendo evidência da habilidade do processo de software em produzir um produto de software que é adequado ao uso.**

- A meta da Garantia da Qualidade é fornecer à gerência os dados necessários para que fique informada sobre a qualidade do produto, ganhando assim compreensão e confiança de que a qualidade do produto está satisfazendo suas metas (PRESSMAN, 2002, p. 190).

VERIFICAÇÃO, VALIDAÇÃO E TESTES NA ENGENHARIA DE SOFTWARE

- Para que tais **defeitos** não perdurem, ou seja, para serem descobertos **antes** de o software ser liberado para utilização, existe uma série de atividades, coletivamente chamadas de Verificação, Validação e Teste, ou "VV&T", **que são atividades da Garantia da Qualidade de Software.**

- VV&T têm a finalidade de garantir que tanto o **modo** pelo qual o software está sendo construído quanto o produto em si estejam em **conformidade** com o especificado.

- Atividades de VV&T não se restringem ao produto final (ao software). **Podem e devem ser conduzidas durante todo o processo de desenvolvimento do software,** desde a sua concepção, e englobam diferentes técnicas *(DELAMARO; MALDONADO; JINO, 2007, p. 1)*.

VERIFICAÇÃO E VALIDAÇÃO

- A **Verificação** se refere ao conjunto de atividades que garante que o software **implementa corretamente** uma função específica, ou seja, busca determinar se os produtos gerados em uma dada fase do processo de software atendem às condições e requisitos estabelecidos no início da fase ou em uma fase anterior e busca responder à seguinte questão (Boehm, 1981, citado por Pressman, 2006, p. 289):
  - Está-se construindo o sistema de **forma** correta?
  - Estamos construindo o produto corretamente?

- A **Validação** se refere a um conjunto de atividades que garante que o software construído **corresponde aos requisitos do cliente**, ou seja, diz respeito a avaliar o software durante ou ao final do processo de desenvolvimento, para determinar se o mesmo **satisfaz aos requisitos especificados** e está livre de falhas (Collofello, 1988, IEEE, 1990, Paulk *et al.*, 1993b) e busca responder à seguinte questão (Boehm, 1981, citado por Pressman, 2006, p. 289):
  - Está-se construindo o **sistema correto**?
  - Estamos construindo o produto certo?

O QUE É TESTE?

- Uma visão comum do processo de teste é de que ele consiste apenas da fase de execução, como executar o programa. **Esta, na verdade, é uma parte do teste, mas não contempla todas as atividades do teste.**

- Existem atividades de teste antes e depois da fase de execução. Por exemplo:
  - planejamento e controle,
  - escolha das condições de teste,
  - modelagem dos casos de teste,
  - checagem dos resultados,
  - avaliação do critério de conclusão,
  - geração de relatórios sobre o processo de teste e sobre sistema alvo e
  - encerramento ou conclusão (ex.: após a finalização de uma fase de teste).
- Teste também inclui revisão de documentos (incluindo o código fonte) e análise estática.

OBJETIVO DO TESTE DE SOFTWARE SEGUNDO PRESSMAN (2016)

- **O objetivo da atividade de Teste de Software é encontrar o maior número de defeitos com a menor quantidade de esforço e tempo** (PRESSMAN, 2016, p. 497).

- Como **benefícios secundários** têm-se:
  - o teste demonstra que as funções do software **parecem estar** funcionando de acordo com a especificação comportamental ou funcional;
  - requisitos de desempenho **parecem estar** sendo satisfeitos.

OBJETIVOS DO TESTE DE SOFTWARE SEGUNDO O ISTQB (2011)

- Testes podem possuir objetivos diferentes:
  - Encontrar defeitos
  - Ganhar confiança sobre o nível de qualidade
  - Prover informações para tomada de decisão
  - Prevenir defeitos.

- Testes dinâmicos e estáticos podem ser usados para atingir objetivos similares e proveem informações para melhorar o sistema a ser testado e o próprio processo de teste.

OBJETIVOS DO TESTE

- No processo de teste, **diferentes pontos de vista levam a diferentes objetivos.**
  - No **teste feito em desenvolvimento** (teste de componente, integração e de sistemas), o principal objetivo pode ser causar o maior número de falhas possíveis, de modo que os defeitos no software possam ser identificados e resolvidos.
  - No **teste de aceite** o objetivo principal pode ser confirmar se o sistema está funcionando conforme o esperado, ou seja, prover a confiabilidade de que esteja de acordo com o requisito.
  - **Em alguns casos o principal objetivo** do teste pode ser avaliar a qualidade do software (não com a intenção de encontrar defeitos), para prover informações sobre os riscos da implantação do sistema em um determinado momento aos gestores.
  - Os **testes de manutenção** podem ser usados para verificar se não foram inseridos erros durante o desenvolvimento de mudanças.
  - Durante os **testes operacionais**, o principal objetivo pode ser avaliar características como confiabilidade e disponibilidade.

QUANTO TESTE É SUFICIENTE?

- Para decidir quanto teste é suficiente, deve-se levar em consideração
  - o nível do risco, incluindo risco técnico, do negócio e do projeto
  - restrições do projeto como tempo e orçamento.

- O teste deve prover informações suficientes aos interessados (*stakeholders*) para tomada de decisão sobre a distribuição do software ou sistema, para as próximas fases do desenvolvimento ou implantação nos clientes.

- No entanto, testes não demonstram ausência de falhas.

É possível dizer que um software não possui erros?

- O Teste Estrutural ou Caixa Branca é um método de projeto de testes que usa a estrutura de controle do projeto procedimental para derivar os casos de teste (Pressman, 2006)
- Baseia-se num minucioso exame dos detalhes procedimentais
- Caminhos lógicos do software são testados
- Não é viável testar todos os caminhos lógicos de um p rograma (teste exaustivo)

Teste Exaustivo

- Programa Pascal com 100 linhas e dois ciclos aninhados que executam entre 1 e 20 vezes cada um dependendo do dado da entrada.
- Dentro do ciclo interior 4 construções se-então-senão.
- 1014 caminhos possíveis de execução
- Se cada caso de teste for executado por um processador "mágico" de testes em 1 mseg
- 3170 anos para completar os testes, ou seja
- Um teste completo e exaustivo não é viável!

ÓRGÃO CERTIFICADOR

- Líder internacional em certificações em teste de software.
- Tem capítulos em vários países.
- Define e mantem um corpo de conhecimento para profissionais na área de teste de software.

- Aplica os exames em **português** no Brasil e segue as regras estabelecidas pelo ISQTB.
- Disponibiliza uma agenda de exames presenciais. Consulte datas e locais em www.bstqb.org.br

- Institutos parceiros do ISTQB que oferecem exames do programa.
- Disponibilizam exames através da PEARSON VUE.

- Aplica exames do iSQI e outros afiliados ao ISTQB.
- Há vários centros disponíveis em todo o Brasil.
- O agendamento deve ser feito com 2 dias úteis.
- Consulte www.pearsonvue.com/isqi

ÓRGÃO CERTIFICADOR

International Software Testing Qualifications Board

- Oficialmente fundado em Edinburgh em novembro de 2002.
- http://www.istqb.org/

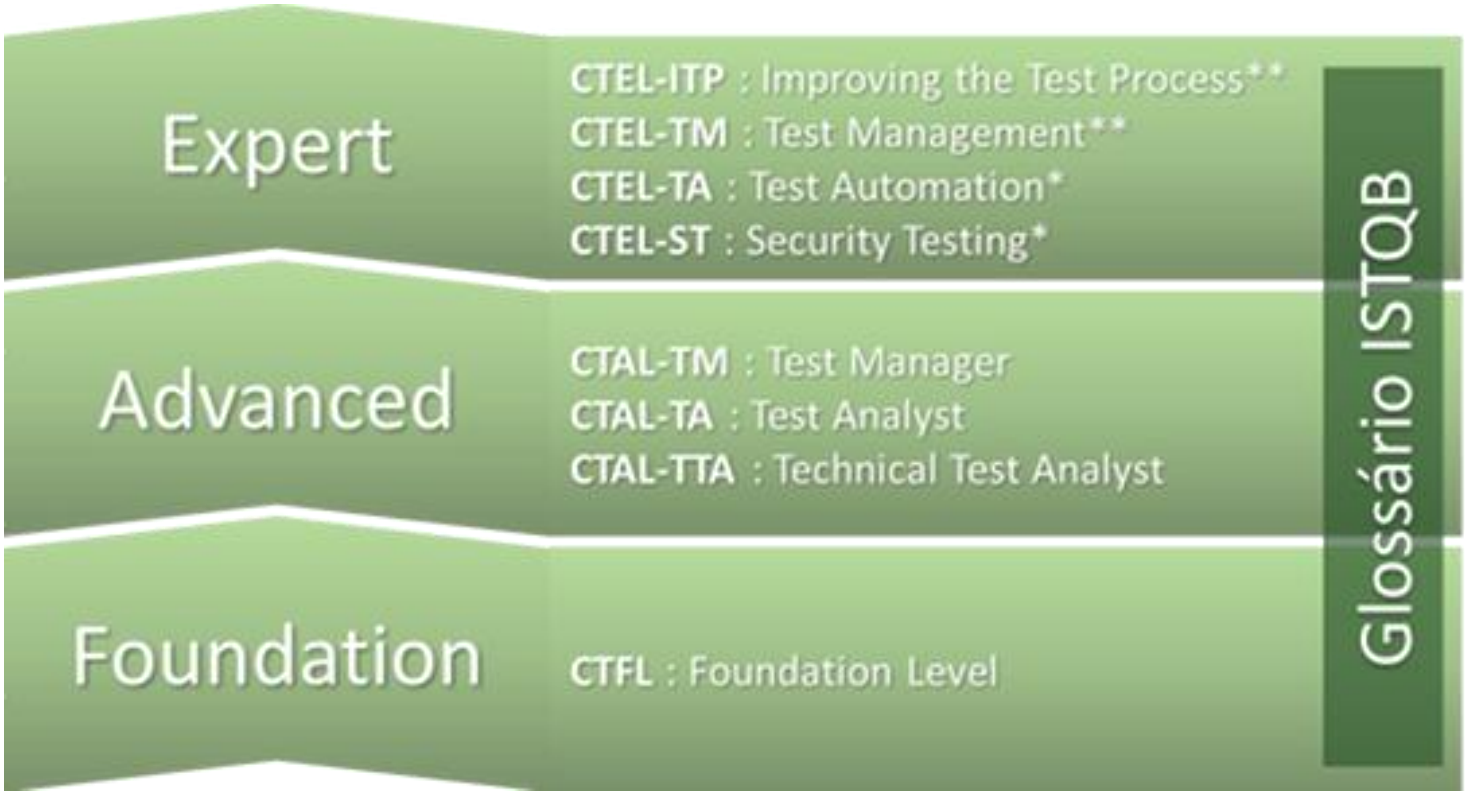
No Brasil: Brazilian Software Testing Qualifications Board
(Comissão Brasileira de Qualificação em Teste de Software)

- Fundado em 2006
- http://www.bstqb.org.br/
- Endereço: Edifício Morumbi Office Tower
  Av. Roque Petroni Junior, 999 – 13.º andar – Sala 132 J
  Vila Gertrudes - 04707-910 - São Paulo – Brasil
- Lista de Certificações ISTQB disponíveis:
  - **Foundation Level (CTFL)**
  - Advanced Level
  - Advanced Test Analyst, Advanced Technical Test Analyst, Advanced Test Manager

VISÃO DO ISTQB

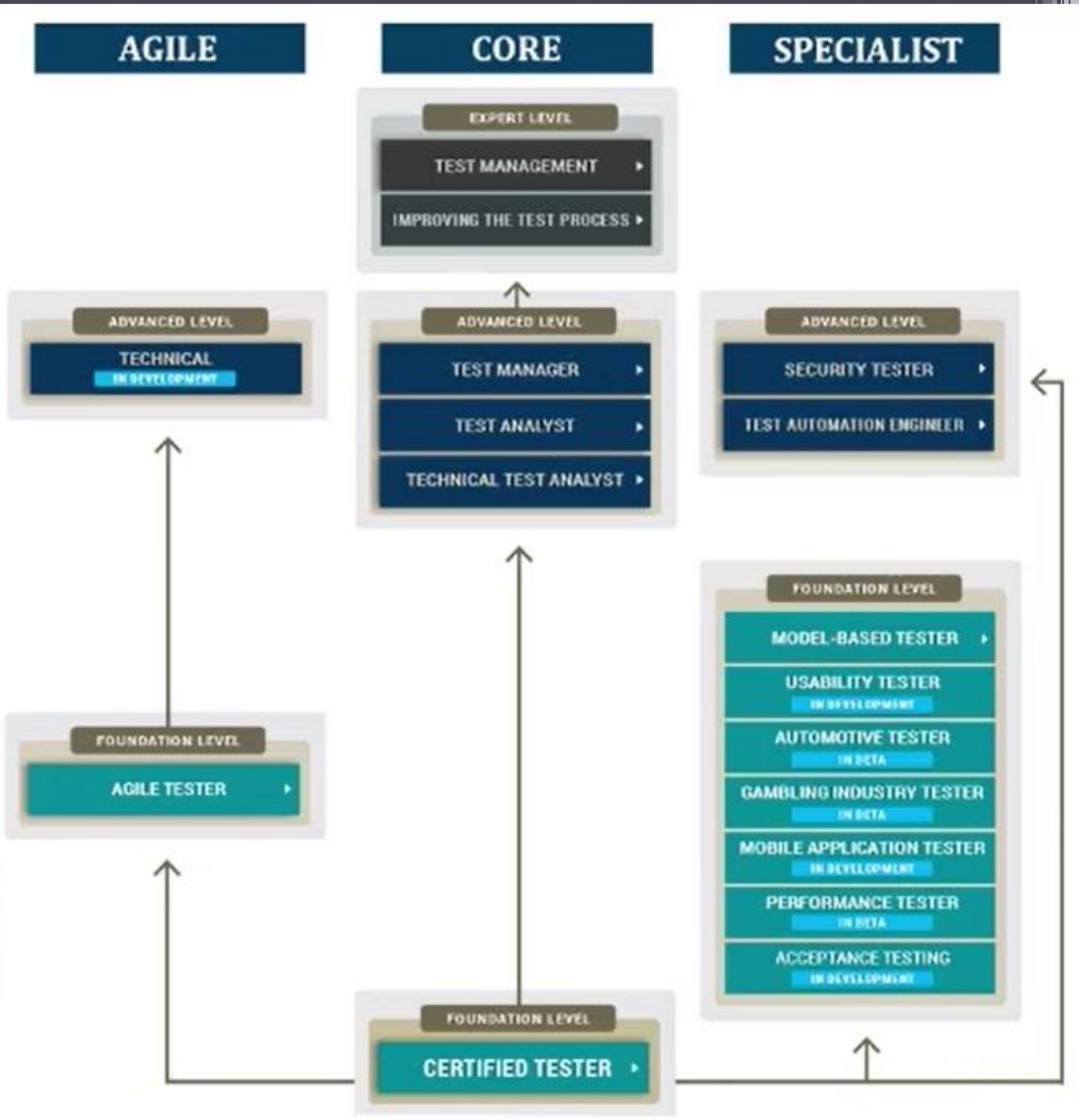
Continuamente melhorar e avançar a profissão de testes de software definindo e mantendo um corpo de conhecimento que permite que os testadores a serem certificados possuam uma base nas melhores práticas, que une a comunidade de teste de software internacionalmente, e incentivar a pesquisa.

CAMINHO DA CERTIFICAÇÃO



- Certificados no mundo: mais de 450.000
- Certificados pelo BSTQB no CTFL: 10.261
- Fonte: BSTQB, ago-2019.

CAMINHO DA CERTIFICAÇÃO



EXAME CTFL – CERTIFIED TESTER FOUNDATION LEVEL

- É caracterizado por 40 (quarenta) questões de múltipla escolha, onde cada questão possui uma pontuação(1).
- Para passar é necessário obter no mínimo 65% dos pontos do exame (26 ou mais pontos).
- A duração do exame é de 60 (sessenta) minutos e um extra de 75 (setenta e cinco) minutos para os candidatos que não tenham como língua nativa a língua-portuguesa desde que avisem a Secretaria do BSTQB (secretaria@bstqb.org.br) durante o período de inscrição.

- Valor: R$ 420,00

SOBRE O SYLLABUS (GUIAS DE ESTUDOS) E SOBRE O GLOSSÁRIO DE TERMOS

- **Objetivo do Syllabus (Guias de Estudos)**
  - Os Syllabus (Guias de Estudos) formam a **base de conhecimento** para a Qualificação Internacional de Teste de Software, tanto para o nível fundamental (CTFL) quanto para o nível avançado (CTAL)

- **Glossário de Termos**
  - apresenta conceitos, termos e definições concebidos para ajudar a comunicação nos testes e disciplinas relacionadas.
  - É elaborado por um Grupo de Trabalho internacional (WG) gerido pelo ISTQB que tem procurado os pontos de vista e comentários na indústria, no comércio e órgãos governamentais e organizações, com o objetivo de produzir um padrão de teste internacional que atenda a todos.

Mapa Conceitual

Fundamentos de teste	O teste durante todo o ciclo de vida do software	Teste estático	Técnicas de teste	Gerenciamento do teste	Ferramentas de suporte ao teste
O que é teste?	Modelos de ciclo de vida	Noções básicas	Categorias de técnicas	Organização de teste	Considerações sobre ferramentas
Por que o teste é necessário?	Níveis de teste	Processo de revisão	Técnicas caixa-preta	Planejamento e estimativa de teste	Uso eficaz de ferramentas
Os 7 princípios do teste	Tipos de teste		Técnicas caixa-branca	Monitoramento e controle dos testes	
Processos de teste	Teste de manutenção		Técnicas baseadas na experiência	Gerenciamento configurações	
A psicologia do teste				Riscos e testes	
				Gerenciamento de defeitos	

Mapa Conceitual

- https://cmaptools.br.uptodown.com/windows
- https://cmap.ihmc.us/
- https://www.mindmeister.com/pt
- https://miro.com/
- https://creately.com/pt/lp/criador-de-mapas-conceitual/

FUNÇÃO DO TESTE NO DESENVOLVIMENTO, MANUTENÇÃO E OPERAÇÃO DE SOFTWARE

- **Rigorosos testes em sistemas e documentações:**
  - podem reduzir os riscos de ocorrência de problemas no ambiente operacional;
  - contribuem para a qualidade dos sistemas de software,

  se os defeitos/falhas/bugs encontrados **forem corrigidos antes** da implantação em ambiente de produção.

- O teste de software pode também ser necessário para atender requisitos contratuais ou legais ou determinados padrões de mercado.

TESTE E QUALIDADE DE SOFTWARE

- Com a ajuda do teste **é possível medir a qualidade do software em termos de defeitos encontrados, por características e requisitos funcionais ou não funcionais do software** (confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade).
- O resultado da execução dos testes pode representar confiança na qualidade do software caso sejam encontrados poucos ou nenhum defeito.
- Um **teste projetado adequadamente** e cuja execução **não encontra defeitos** reduz o nível de riscos em um sistema.
- Por outro lado, quando os testes encontram defeitos, a qualidade do sistema aumenta quando estes são corrigidos.

DEPURAÇÃO DE CÓDIGO X TESTE

- São atividades diferentes.
- **A depuração de código** (*debugging)* é o processo de procurar, analisar e remover as **causas** de falhas no software, ou seja, identifica a causa de um defeito, repara o código e checa se os defeitos foram corrigidos corretamente. Depois é feito um teste de confirmação por um testador para certificar se a falha foi eliminada.
- **Testar (***testing)* é o processo que consiste em todas as atividades do ciclo de vida, tanto estáticas quanto dinâmicas, voltadas para o planejamento, preparação e avaliação de produtos de software e produtos de trabalho relacionados a fim de determinar se elas satisfazem os requisitos especificados e demonstrar que estão aptas para sua finalidade e para a detecção de defeitos.
- **Teste** (*test)* é o conjunto de um ou mais casos de teste (IEEE 829). Testes podem demonstrar falhas que são causadas por defeitos.
- **As responsabilidades de cada atividade são bem distintas: testadores testam e desenvolvedores depuram.**

OS SETE PRINCÍPIOS DO TESTE

- ***Princípio 1 – Teste demonstra a presença de defeitos***
  - O teste pode demonstrar a presença de defeitos, mas não pode provar que eles não existem. O Teste reduz a probabilidade que os defeitos permaneçam em um software, mas mesmo se nenhum defeito for encontrado, não prova que ele esteja perfeito.
- ***Princípio 2 – Teste exaustivo é impossível***
  - Testar tudo (todas as combinações de entradas e pré-condições) não é viável, exceto para casos triviais. Em vez do teste exaustivo, riscos e prioridades são levados em consideração para dar foco aos esforços de teste.
- ***Princípio 3 – Teste antecipado***
  - A atividade de teste deve começar o mais breve possível no ciclo de desenvolvimento do software ou sistema e deve ser focado em objetivos definidos.
- ***Princípio 4 – Agrupamento de defeitos***
  - Um número pequeno de módulos contém a maioria dos defeitos descobertos durante o teste antes de sua entrega ou exibe a maioria das falhas operacionais.

OS SETE PRINCÍPIOS DO TESTE

- ***Princípio 5 – Paradoxo do Pesticida***
  - Pode ocorrer de um mesmo conjunto de testes que são repetidos várias vezes não encontrarem novos defeitos após um determinado momento. Para superar este "paradoxo do pesticida", os casos de testes necessitam ser frequentemente revisados e atualizados. Um conjunto de testes novo e diferente precisa ser escrito para exercitar diferentes partes do software ou sistema com objetivo de aumentar a possibilidade de encontrar mais erros.
- ***Princípio 6 – Teste depende do contexto***
  - Testes são realizados de forma diferente conforme o contexto. Por exemplo, softwares de segurança crítica são testados diferentemente de um software de comércio eletrônico.
- ***Princípio 7 – A ilusão da ausência de erros***
  - Encontrar e consertar defeitos não ajuda se o sistema construído não atende às expectativas e necessidades dos usuários.

PROCESSO DE TESTE

- **Consiste das seguintes atividades:**
  - **Planejamento e controle**
  - **Análise e modelagem**
  - **Implementação e execução**
  - **Avaliação dos critérios de saída e relatórios**
  - **Atividades de encerramento de teste**

- A parte mais visível do teste é a execução mas para se obter eficácia e eficiência, os planos de teste precisam conter o tempo a ser gasto no planejamento dos testes, modelagem dos casos de testes e preparação da execução e avaliação de resultados.

- Apesar de serem apresentadas sequencialmente, as atividades durante o processo podem sobrepor-se ou acontecer de forma concorrente, adaptando essas atividades principais dentro do contexto do sistema e do projeto, quando necessário.

PLANEJAMENTO E CONTROLE DE TESTE

- **Planejamento de teste**
  - é a atividade que consiste em definir os objetivos e especificar as atividades de forma a alcançá-los.

- **Controle do teste**
  - é a constante atividade que consiste em comparar o progresso atual contra o que foi planejado, reportando o status e os desvios do plano.
  - Ele envolve ainda a tomada de ações necessárias para alcançar a missão e objetivos do projeto.
  - Para um controle efetivo, o teste deverá ser monitorado durante todo o projeto.

- O planejamento do teste leva em consideração o retorno de informações das atividades de monitoração e controle.

ANÁLISE E MODELAGEM DO TESTE

- São atividades onde os objetivos gerais do teste são transformados em condições e modelos de teste tangíveis.

- **São compostas pelas seguintes atividades principais:**
  - Revisar a base de testes (como requisitos, nível de integridade do software (nível de risco), arquitetura, modelagem, interfaces).
  - Avaliar a testabilidade dos requisitos e do sistema.
  - Identificar e priorizar as condições ou requisitos de testes e dados de testes baseados na análise dos itens de teste, na especificação, no comportamento e na estrutura.
  - Projetar e priorizar os casos de testes de alto nível.
  - Identificar as necessidades de dados para teste suportando as condições e casos de teste
  - Planejar a preparação do ambiente de teste e identificar a infraestrutura e ferramentas necessárias.
  - Criar uma rastreabilidade bidirecional entre os requisitos e os casos de teste.

IMPLEMENTAÇÃO E EXECUÇÃO DE TESTE

A implementação e execução do teste é a atividade onde os procedimentos ou os scripts de teste são especificados pela combinação dos casos de teste em uma ordem particular, incluindo todas as outras informações necessárias para a execução do teste, o ambiente é preparado e os testes são executados.

São compostas pelas seguintes atividades principais:

- Finalizar, implementar e priorizar os casos de teste (incluindo a identificação dos dados para teste).
- Desenvolver e priorizar os procedimentos de teste, criar dados de teste e, opcionalmente, preparar o ambiente para teste e os scripts de testes automatizados.
- Criar suítes de teste a partir dos casos de teste para uma execução de teste eficiente.
- Verificar se o ambiente está preparado corretamente.
- Verificar e atualizar a rastreabilidade bidirecional entre a base de teste e os casos de teste.
- Executar os casos de teste manualmente ou utilizando ferramentas de acordo com a sequência planejada.

IMPLEMENTAÇÃO E EXECUÇÃO DE TESTE

São compostas pelas seguintes atividades principais **(continuação):**

- Registrar os resultados da execução do teste e anotar as características e versões do software em teste, ferramenta de teste e *testware* (*o ambiente e a infraestrutura de teste para o reuso*).
- Comparar resultados obtidos com os resultados esperados.
- Reportar as discrepâncias como incidentes e analisá-los a fim de estabelecer suas causas (por exemplo, defeito no código, em algum dado específico de teste, na documentação de teste ou uma execução de inadequada do teste).
- Repetir os testes como resultado de ações tomadas para cada discrepância. Por exemplo, re-execução de um teste que falhou previamente quando da confirmação de uma correção (teste de confirmação), execução de um teste corrigido e/ou execução de testes a fim de certificar que os defeitos não foram introduzidos em áreas do software que não foram modificadas, ou que a correção do defeito não desvendou outros defeitos (teste de regressão).

AVALIAÇÃO DO CRITÉRIO DE SAÍDA E RELATÓRIO

- **A Avaliação do critério de saída** é a atividade onde a execução do teste é avaliada mediante os objetivos definidos. Deve ser feito para cada nível de teste.
- A **avaliação do critério de saída** é composta pelas seguintes atividades principais:
  - Checar os registros de teste (*logs)* mediante o critério de encerramento especificado no planejamento de teste.
  - Avaliar se são necessários testes adicionais ou se o critério de saída especificado deve ser alterado.
  - Elaborar um relatório de teste resumido para os interessados (*stakeholders)*.

ATIVIDADES DE ENCERRAMENTO DE TESTE

- Na atividade de encerramento de teste são coletados os dados de todas as atividades para consolidar a experiência, *testware,* fatos e números.
- *Por exemplo, quando um software é lançado, um projeto de teste é completado (ou cancelado), um marco do projeto foi alcançado, ou a implantação de uma demanda de manutenção foi completada.*
- As atividades de encerramento de teste são compostas pelas seguintes atividades principais:
  - Checar quais entregáveis planejados foram realmente entregues
  - Fechar os relatórios de incidentes, ou levantar os registros de mudança que permaneceram abertos.
  - Documentar o aceite do sistema.
  - Finalizar e arquivar o *testware, o ambiente de teste e infraestrutura de teste para o reuso.*
  - Entregar o *testware para a manutenção da organização.*
  - Analisar as lições aprendidas para se determinar as mudanças necessárias para futuros *releases e projetos*.
  - Utilizar as informações coletadas para melhorar a maturidade de teste

A PSICOLOGIA DO TESTE

- A forma de pensar utilizada enquanto se está testando e revisando é diferente da utilizada enquanto se está analisando e desenvolvendo.
- Com a sua forma de pensar, os **desenvolvedores** estão aptos a testarem seus próprios códigos, mas a separação desta responsabilidade para um **testador** é tipicamente feita para ajudar a focalizar o esforço e prover benefícios adicionais, como uma visão independente, profissional e treinada de recursos de teste.
- Teste independente pode ser considerado em qualquer nível de teste.
- Certo grau de independência (evitando a influência do autor) muitas vezes representa uma forma eficiente de encontrar defeitos e falhas.
- Independência não significa simplesmente uma substituição, tendo em vista que os desenvolvedores podem encontrar defeitos no código de maneira eficiente.

A PSICOLOGIA DO TESTE

- Níveis de independência podem ser definidos como:
  - Teste elaborado por quem escreveu o software que será testado (baixo nível de independência).
  - Teste elaborado por outra(s) pessoa(s) (por exemplo, da equipe de desenvolvimento).
  - Teste elaborado por pessoa(s) de um grupo organizacional diferente (ex.: equipe independente de teste).
  - Teste elaborado por pessoa(s) de diferentes organizações ou empresas (terceirizada ou certificada por um órgão externo).
- Pessoas e projetos são direcionados por objetivos. Pessoas tendem a alinhar seus planos com os objetivos da gerência e outros envolvidos (“*stakeholders”) para, por exemplo, encontrar defeitos ou confirmar que o software funciona. Desta forma, é importante ter objetivos claros do teste.*

A PSICOLOGIA DO TESTE

- **Identificar falhas durante o teste pode ser considerado uma crítica contra o produto e o autor (responsável pelo produto).** Teste é, nestes casos, visto como uma atividade destrutiva, apesar de ser construtiva para o gerenciamento do risco do produto.
- Procurar por falhas em um sistema requer curiosidade, pessimismo profissional, um olhar crítico, atenção ao detalhe, comunicação eficiente com os profissionais do desenvolvimento e experiência para encontrar erros.
- **Se os erros, defeitos ou falhas são comunicados de uma forma construtiva**, podem-se evitar constrangimentos entre as equipes de teste, analistas e desenvolvedores, tanto na revisão quanto no teste.
- **O testador e o líder da equipe de teste precisam ter boa relação com as pessoas para comunicar informações sólidas sobre os defeitos, progresso e riscos de uma forma construtiva.** A informação do defeito pode ajudar o autor do software ou documento a ampliar seus conhecimentos. Defeitos encontrados e resolvidos durante o teste trará ganho de tempo e dinheiro, além de reduzir os riscos.

A PSICOLOGIA DO TESTE

- Problemas de comunicação podem ocorrer, especialmente se os testadores forem vistos somente como mensageiros de más notícias ao informar os defeitos.
- De qualquer forma, existem formas de melhorar a comunicação e o relacionamento entre os testadores e os demais:
  - Começar com o espírito de colaboração, ao invés de disputa (conflitos), onde todos têm o mesmo objetivo para alcançar a melhor qualidade do sistema.
  - Comunicar os erros encontrados nos produtos de uma forma neutra, dar foco no fato sem criticar a pessoa que o criou, por exemplo, escrevendo objetivamente o relatório de incidentes.
  - Tentar compreender como a pessoa se sente ao receber a notícia e interpretar sua reação.
  - Confirmar que a outra pessoa compreendeu o que você relatou e vice-versa.

CÓDIGO DE ÉTICA DE TESTE

O envolvimento em teste de software permite que pessoas conheçam informações confidenciais e privilegiadas. Um código de ética é necessário, entre outros motivos, para garantir que a informação não seja usada de forma inapropriada.

O ISTQB® estabelece a **obrigatoriedade** no seguinte código de ética para quem se certificar:

- PÚBLICO – Testadores certificados devem atuar consistentemente com o interesse público.
- CLIENTE E EMPREGADOR – Testadores certificados devem agir da melhor forma para os interesses de seus clientes e empregadores, consistente com o interesse público.
- PRODUTO – Testadores certificados devem garantir que os entregáveis que eles fornecem (produtos e sistemas que eles testam) correspondem aos mais altos padrões profissionais possíveis.
- JULGAMENTO – Testadores certificados devem manter integridade e independência em seu julgamento profissional.

CÓDIGO DE ÉTICA DE TESTE

- GERENCIAMENTO – Gerentes e líderes de teste certificados devem se submeter e promover uma abordagem ética ao gerenciamento do teste de software.
- PROFISSÃO – Testadores certificados devem promover a integridade e reputação da profissão, consistentemente com o interesse público.
- COLEGAS – Testadores certificados devem ser agradáveis e incentivadores com seus colegas, e promoverem a cooperação com os desenvolvedores de software.
- INDIVÍDUO – Testadores certificados devem praticar um aprendizado vitalício em consideração à prática de sua profissão e devem promover uma abordagem ética nessa prática.

Referências

- Baseado no material da Profa. Denise Togneri e do Prof. Ralf Luis de Moura