Curso
Qualidade de software

escola
britânica de
artes criativas
& tecnologia

Qualidade de Software

Módulo 3

escola britânica de artes criativas & tecnologia

Planning: O que você vai aprender

- O que é teste de software
- Tipos de testes
- Fases de teste
- O que é garantia da qualidade
- Bugs
- Como reportar e acompanhar um bug

Teste de software

Aula 1

O que é teste de Software?

- Processo que consiste em todas as atividades do ciclo de vida, estáticas e dinâmicas, relativas ao planejamento, à preparação e à avaliação de um sistema;
- Determina que estes satisfazem os requisitos especificados, para demonstrar que são adequados para o propósito e para detectar defeitos.

ISTQB® (International Software Testing Qualifications Board)

Princípios dos testes

Fonte: Sylabus CTFL - 2018

1 - O teste mostra a presença de defeitos e não a sua ausência

O teste reduz a probabilidade de defeitos não descobertos permanecerem no software,

mas, mesmo se nenhum defeito for encontrado, o teste não é uma prova de correção.

2. Testes exaustivos são impossíveis

Testar tudo (todas as combinações de entradas e pré-condições) não é viável, exceto em casos triviais.

Em vez de tentar testar exaustivamente, a análise de risco, as técnicas de teste e as prioridades devem ser usadas para concentrar os esforços de teste.

3. O teste inicial economiza tempo e dinheiro

Para encontrar antecipadamente os defeitos, as atividades de teste estático e dinâmico devem iniciar o mais cedo possível no ciclo de vida de desenvolvimento de software.

4. Defeitos se agrupam

Um pequeno número de módulos geralmente contém a maioria dos defeitos descobertos durante o teste de pré-lançamento ou é responsável pela maioria das falhas operacionais.

5. Cuidado com o paradoxo do pesticida

Se os mesmos testes forem repetidos várias vezes, esses testes não encontrarão novos defeitos.

Para detectar novos defeitos, os testes existentes e os dados de teste podem precisar ser alterados e novos testes precisam ser gravados.

6. O teste depende do contexto

O teste é feito de forma diferente em diferentes contextos

Por exemplo, o software de controle industrial de segurança crítica é testado de forma diferente de um aplicativo móvel de comércio eletrônico.

7. Ausência de erros é uma ilusão

Algumas organizações esperam que os testadores possam executar todos os testes possíveis e encontrar todos os defeitos possíveis

Além disso, é uma ilusão esperar que apenas encontrar e corrigir muitos defeitos garanta o sucesso de um sistema.

Assim como esperar que o QA é o único responsável pela qualidade.

Relação entre níveis, tipos e técnicas

O que testar?

Tipos de testes
Testes de funcionalidade
Testes de usabilidade
Testes de performance
Testes de segurança
...

Como testar?

Técnicas de testes
Teste funcional (caixa preta)
Teste estrutural (caixa branca)

Quando testar?

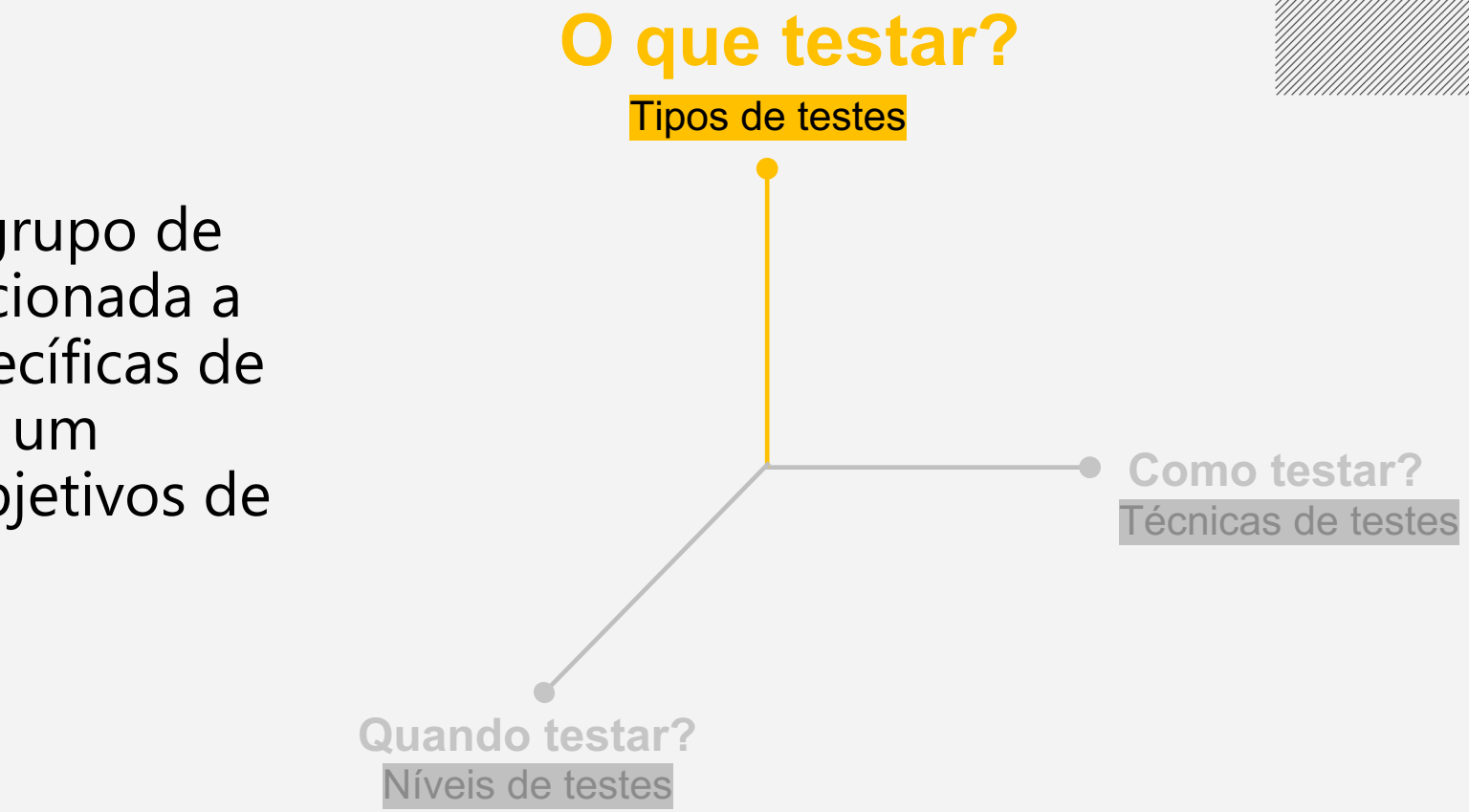
Níveis de testes
Testes de unidade
Testes de integração
Testes de Sistema
Testes de aceitação

Tipos de testes

Aula 2

Tipos de Teste

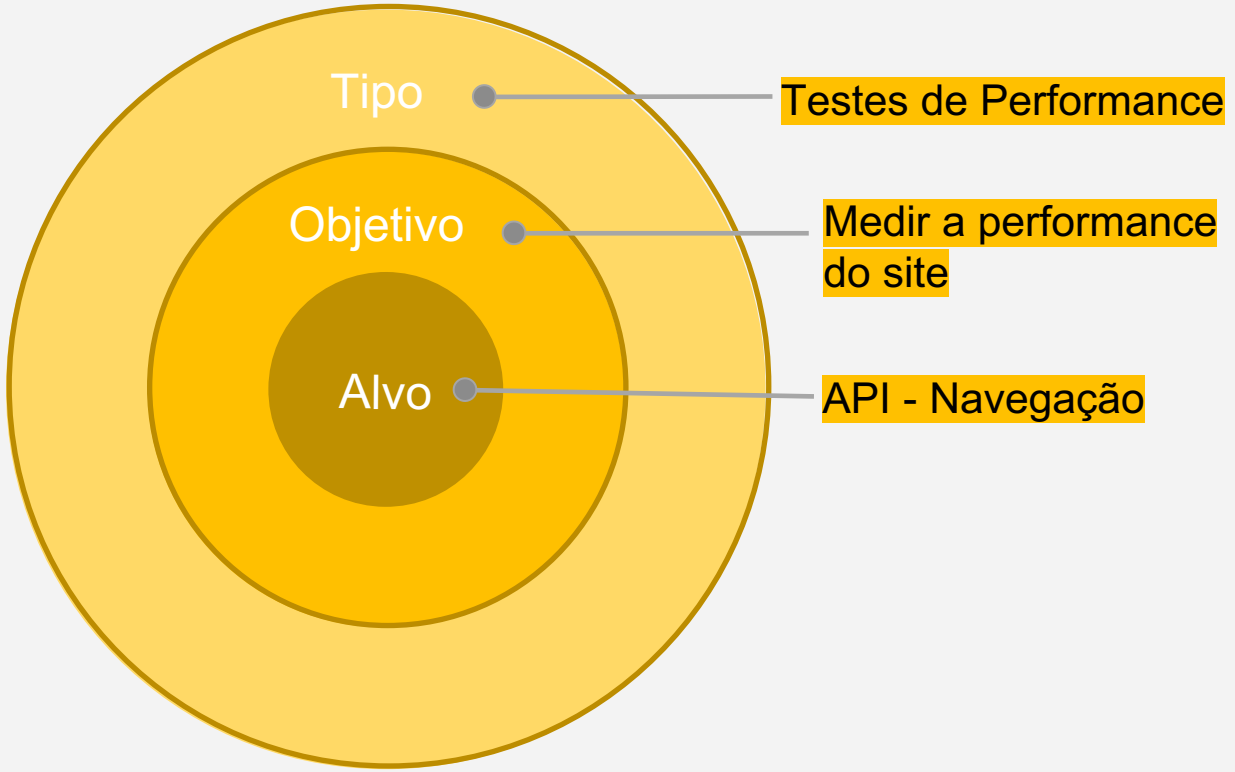
Um **tipo de teste** é um grupo de atividades de testes direcionada a testar características específicas de um software ou parte de um sistema, com base em objetivos de testes específicos.



Objetivos dos tipos de Testes

Tipos de testes servem para avaliar:

- Funcionalidade;
- Usabilidade;
- Performance;
- Segurança;
- Mudança;
- Estrutura
- Confiabilidade;
- Compatibilidade;



Fonte: Sylabus CTFL - 2018

Visão macro dos Tipos de Testes

Para **garantir** cada objetivo, um ou mais tipos de este são utilizado.

escola britânica de artes criativas & tecnologia

I - Testes funcionais

- Avaliar as funções que o sistema deve executar
- **O que** o sistema deve fazer (comportamento) ?
- Valida a implementação correta e apropriada das regras de negócio
- Devem ser realizados em todos os níveis de teste

II - Testes não funcionais

O "quão bem" o sistema se comporta?

- Analisa os aspectos importantes, mesmo que não estejam relacionadas diretamente às funções que o sistema desempenha.
- Avalia características como:
  - Usabilidade
  - Desempenho
  - Segurança
  - Carga de trabalho
  - Recuperação de falhas
  - Etc.

III - Teste caixa-branca

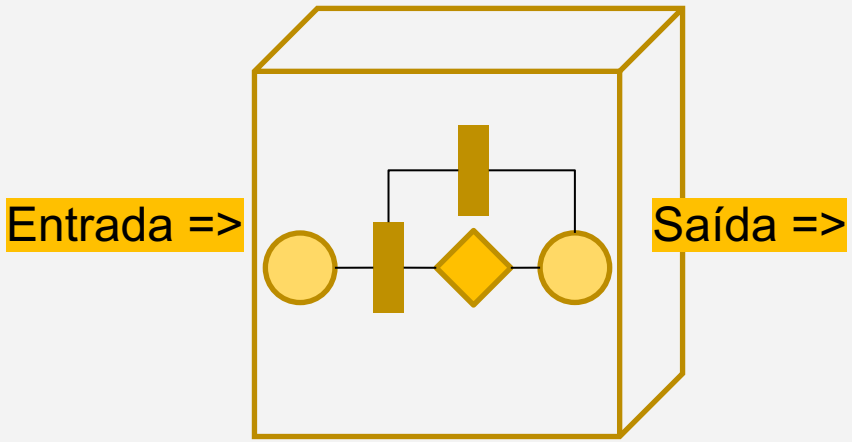
Também conhecido como teste estrutural

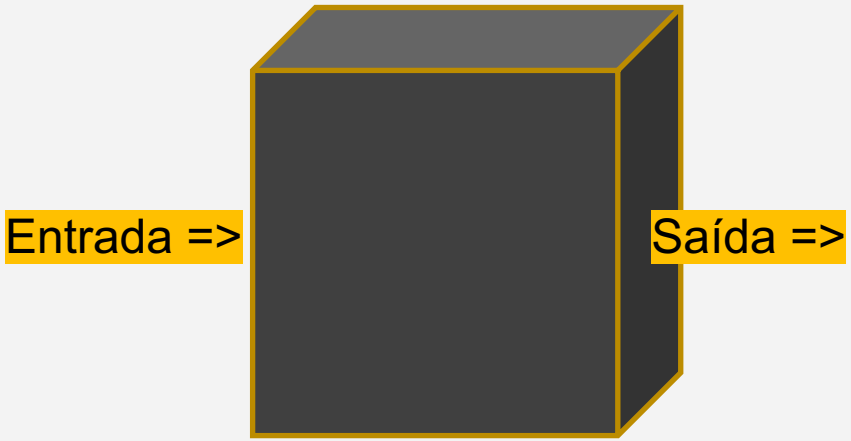
É baseada na estrutura interna ou na implementação do sistema.

Incluí código, arquitetura, fluxos de dados, controle e e trabalho.

Conseguimos medir a cobertura de código.

Diferenças Caixa branca e Caixa preta



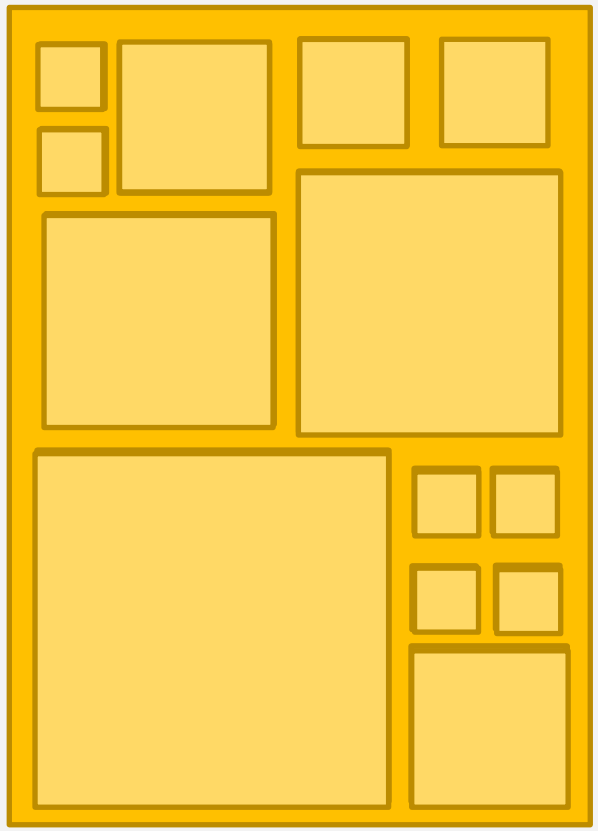


IV - Teste relacionado à mudança

Quando são feitas alterações em um sistema, deve-se testar para confirmar se as alterações corrigiram o defeito ou implementaram a funcionalidade corretamente e não causaram **consequências**.

Testes relacionado à mudança

Sistema v.1.2



Teste de confirmação

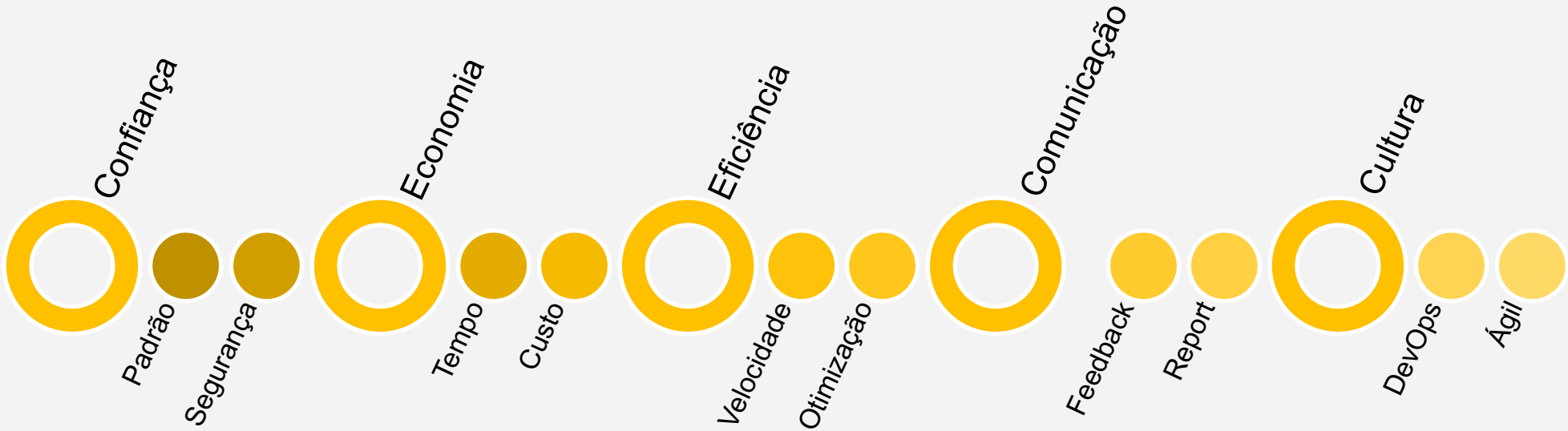
Chamado também de **reteste**, é executado após a correção de um defeito. Valida o sucesso das ações corretivas.

Teste de regressão

Teste realizado para garantir que nenhuma outra parte do sistema foi afetada após alguma modificação no código.

Testes automatizados

Ferramentas para executar os testes de forma automática. Seu objetivo é simular a ação de um usuário de forma automática.

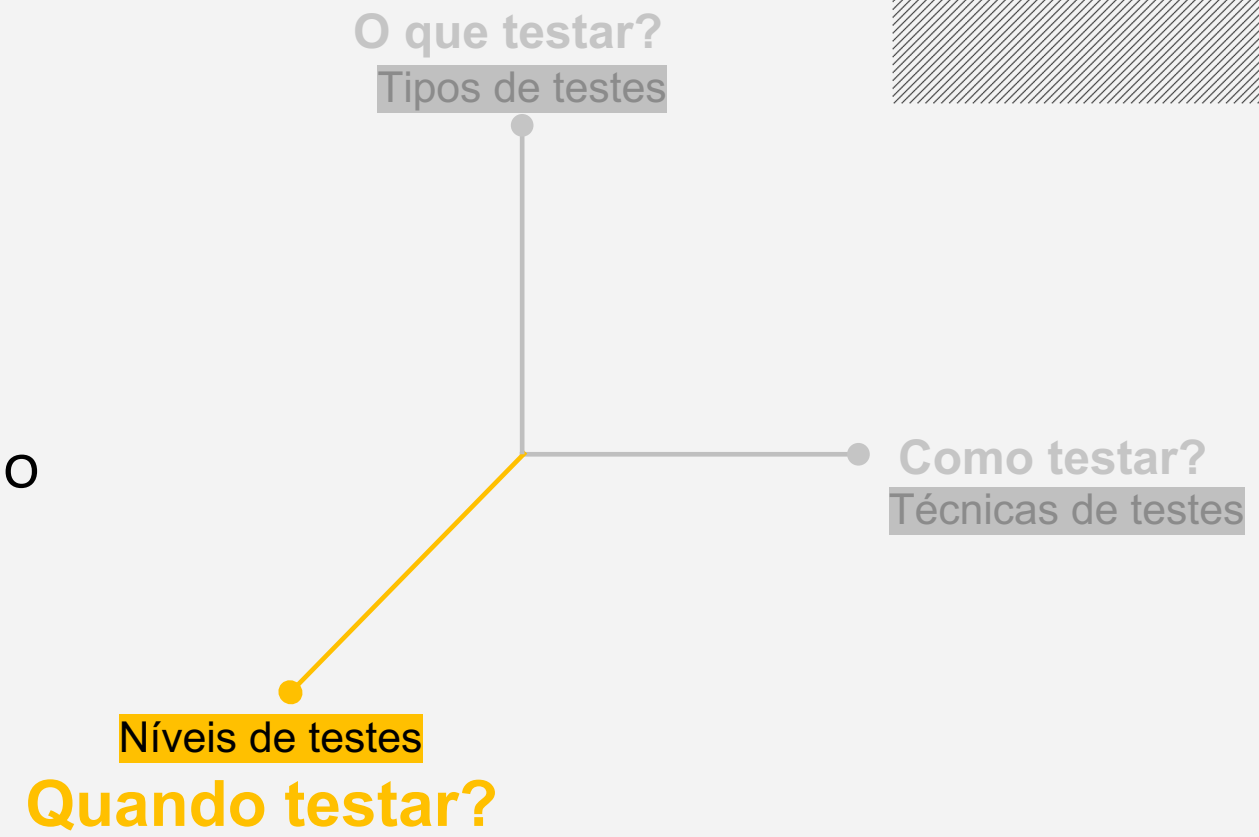


Níveis de testes

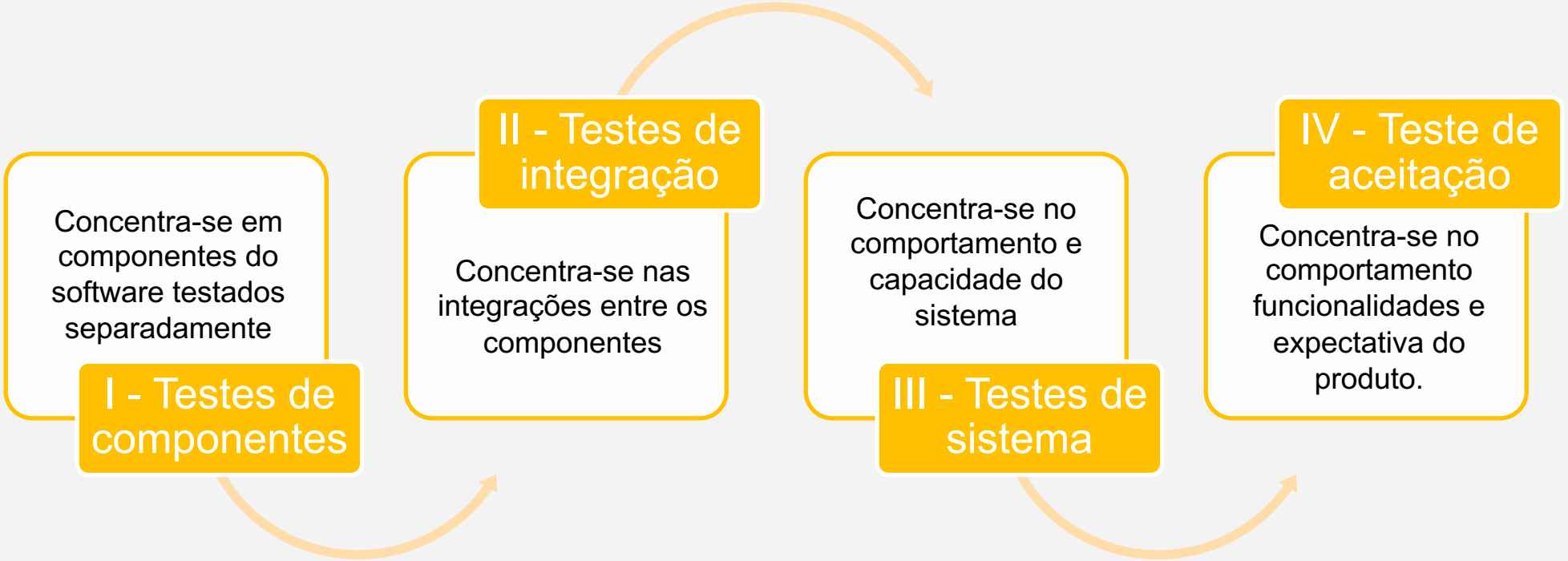
Aula 3

Níveis de testes

- Também chamados de **estágio de testes**;
- Definem basicamente a fase em que estes serão executados, ou seja, qual o melhor momento de executar determinado teste.

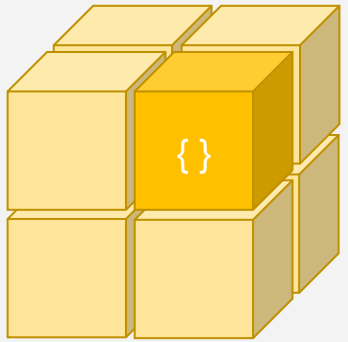


Níveis de testes



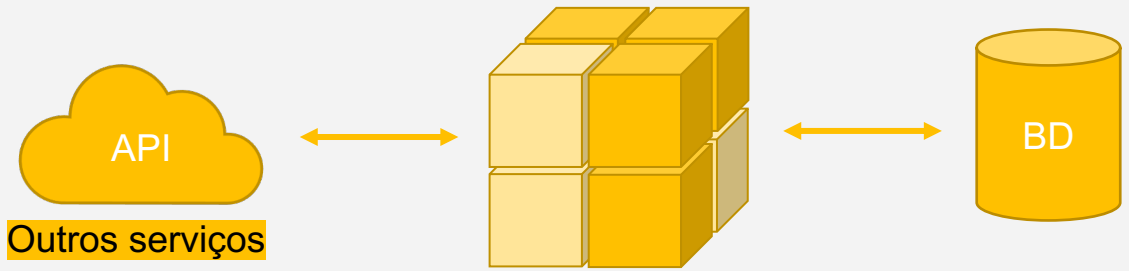
I - Testes de Componentes

- Também chamado de **Testes de Unidade**
- Realizado pelo próprio **desenvolvedor**;
- Testa a **menor** unidade do sistema, geralmente um método;
- Feedback mais rápido;
- O teste deve responder às perguntas:
  - O que eu estou testando?
  - O que o método deveria fazer?
  - Qual o seu atual retorno?
  - O que eu espero que retorne?



III – Teste de Integração

- Testa a integração das unidades que já foram testados individualmente;
- Geralmente é realizado de forma incremental, para que cada módulo ou componente seja incluído sequencialmente até que todos os casos de testes possam ser executados;
- Realizado por desenvolvedores que não participaram da construção ou pela equipe de testes;

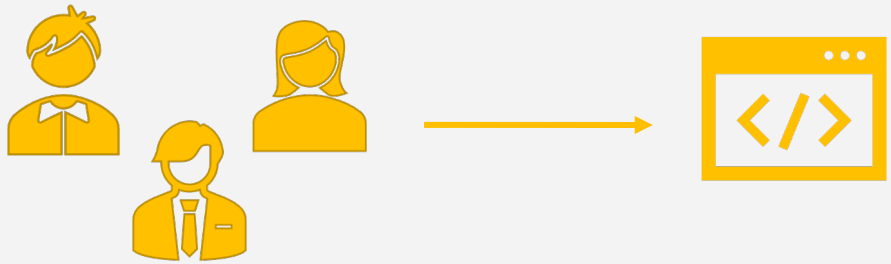


IV - Teste de Sistema

- Também conhecido como Testes funcionais;
- Aplicado pela equipe de testes;
- Verificar as funcionalidades do sistema como um todo utilizando técnicas funcionais para a geração de casos de testes;
- **Ambiente** controlado semelhante ao do cliente;
- Gera parâmetros para permitir a liberação do produto ao cliente;
- Responde a pergunta: “Seus recursos funcionam corretamente?”

IV – Testes de Aceitação

- Testes realizados pelos **usuários** da aplicação;
- Ocorre antes do sistema ser liberado em produção;
- Verifica se a solução atende os objetivos de negócio;
- Composto por duas categorias:
  - Alfa: Ambiente controlado para execução dos testes.
  - Beta: usuário livre para fazer as simulações de uso reais no sistema.



Testes exploratórios

- Uma das técnica dos **teste de aceitação**
- Todos do time podem executar os testes de forma dinâmica e com base nos seus conhecimentos, na exploração do item de teste e nos resultados de testes anteriores.

Testes exploratórios é uma boa alternativa quando:

- O projeto tem um prazo muito apertado;
- Quando a documentação está desatualizada;
- Também deve ser considerado como uma abordagem complementar a um teste planejado;
- Usa vários estilos baseados em experiências;
- Ex.: Estilo Livre, onde o sistema é testado de maneira **ad-hoc** e não há muitas diretrizes ou procedimentos para o teste.

O termo “**Ad Hoc**” vem da língua latina que significa “para isso” ou “para este efeito”.

Bugs

Aula 4

Bug

- O bug ocorre quando eu não consigo realizar a ação no sistema, tornando-se um impedimento na rotina de trabalho.
- Logo, **bug** é causado por erros no código do sistema e devem ser um motivo de preocupação, pois podem causar muitos problemas.
- Basicamente, é quando o software não funciona como esperamos.

Como surgiu

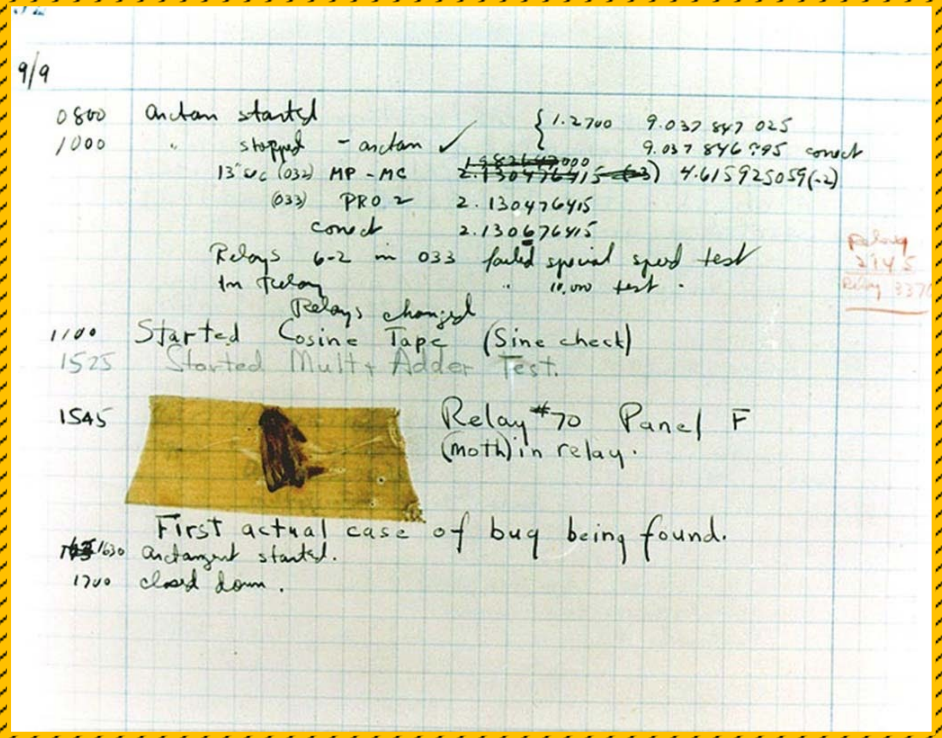
- Em 1947, foi documentado o primeiro “bug” da história da informática.
- Um grupo de investigadores trabalhava no **Mark II.**
- Após várias tentativas de detectar uma **falha** no sistema que impedia o funcionamento do ordenador, detectaram uma **mariposa** presa no relé do Mark II.

Mark II, general view of calculator frontpiece, 1948.

Primeiro “bug” encontrado

Grace Murray Hopper, uma das cientistas que trabalhava no projeto documentou o feito em seu caderno.

“First actual case of bug being found”
(Primeiro caso real de um bug encontrado)



Erros, defeitos e falhas

Erro → **Defeito** → **Falha**

Ou engano: Ação humana que produz um resultado incorreto. Ex. Uma ação incorreta tomada pelo desenvolvedor ou analista;

Ou **bug**: Passo, processo ou definição de dados incorretos. Ex. Uma instrução incorreta no código ou uma falta num documento;

Desvio entre o resultado apresentado pela aplicação em relação aos requisitos. Um comportamento incorreto.

Causas de erro

Erros podem ocorrer por vários motivos, como:

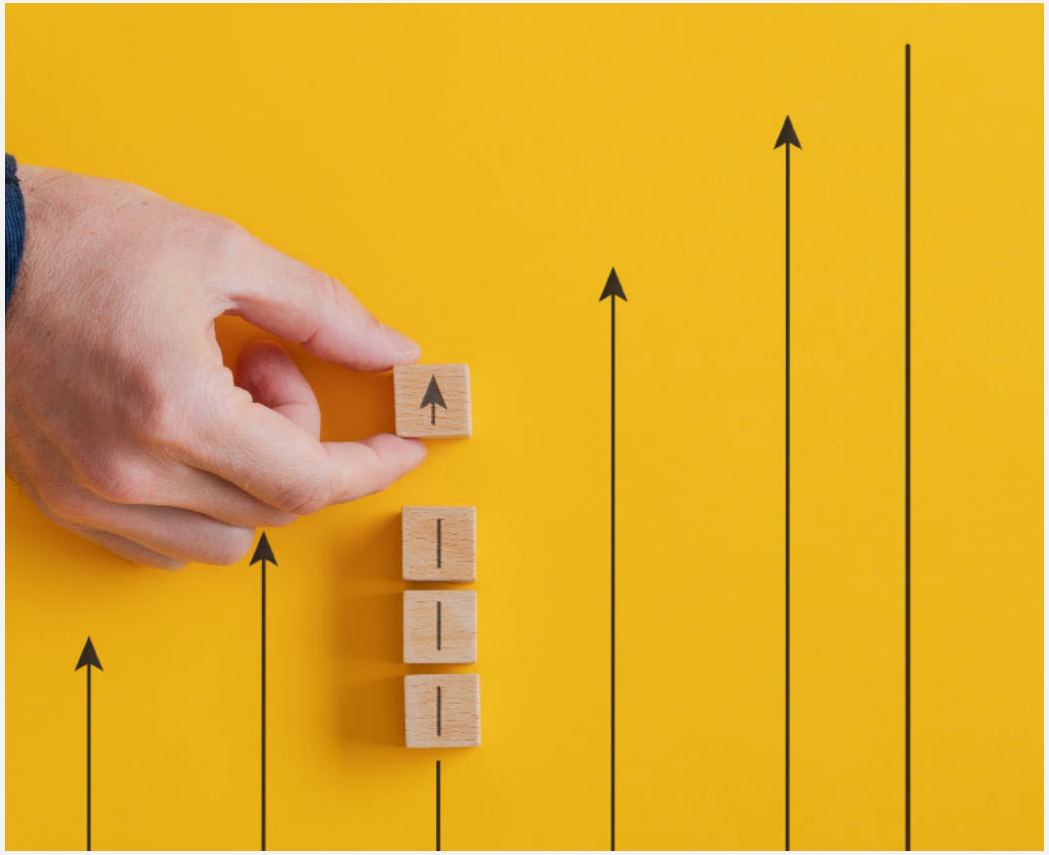
- Pressão do tempo;
- Participantes do projeto inexperientes ou insuficientemente qualificados;
- Falta de comunicação entre os participantes do projeto;
- Complexidade do projeto de desenvolvimento;
- Tecnologias novas ou desconhecidas.

Bugs vs Melhoria

- **Melhoria** é quando eu tenho uma ação no sistema que está **funcionando** conforme os requisitos, mas eu vejo que pode ser melhorada para trazer mais valor para um produto.
- Um **bug** pode ser a origem de uma melhoria como por exemplo uma melhoria de tela, fluxo do sistema, segurança, funcional, etc.
- A melhoria por vezes surge após a implementação de uma funcionalidade em ambientes de homologação ou produção.
- Portanto, bug é causado por um erro e melhoria é uma evolução sistêmica.

Melhoria contínua

- Caso aconteça um erro em tempo de projeto, não se preocupem, pois estamos passíveis de erros.
- O que não pode acontecer é "encobrir" o erro, pois quanto mais tardio a descoberta do erro mais caro fica para consertar.
- Aprendemos com erros e melhoramos!



Acompanhamento de Bugs

Aula 5

Acompanhamento de Bugs

- Também chamado de “Bug Tracking”, ou “Rastreamento de bugs” é a gestão de defeitos no Teste de Software;
- Normalmente auxiliado por um ferramenta colaborativa entre o time de desenvolvimento;

Objetivo

Fornecer **informações** sobre o evento inesperado para o desenvolvedor corrigir o problema;

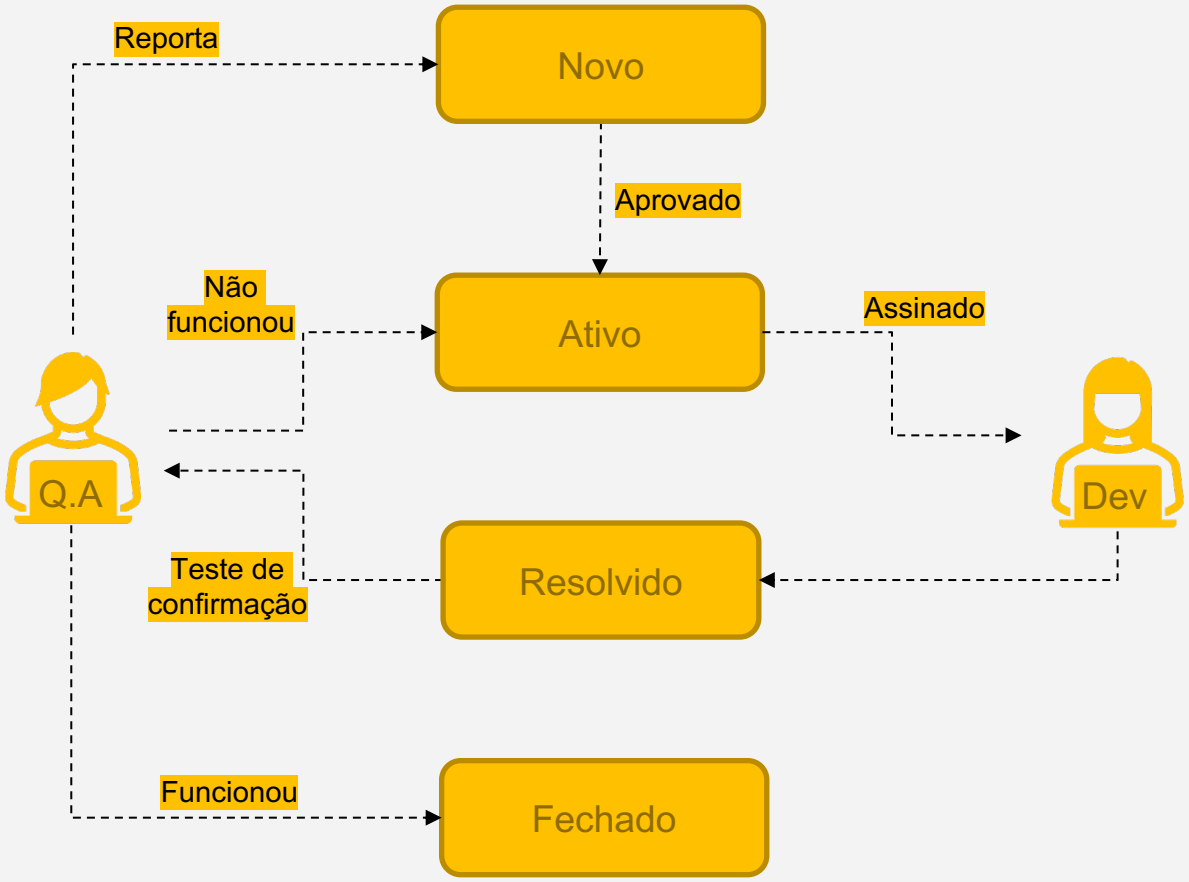
Fornecer aos gerentes um meio de **rastrear** a qualidade do produto e impacto no qualidade;

Fornecer ideia para o desenvolvimento e **melhoria** de processo de desenvolvimento e teste;

Quando reportar

O bug pode ser reportado durante **todo o ciclo de desenvolvimento** e quanto antes melhor, pois diminui o risco de um problema em produção.

Fluxo de correção



Como reportar

Indispensáveis:

- ID / Título:
- Descrição / Resumo:
- Ambiente:
- URL:
- Evidências (Prova visual):
- Etapas para reproduzir:
- Resultado e Comportamento esperado:

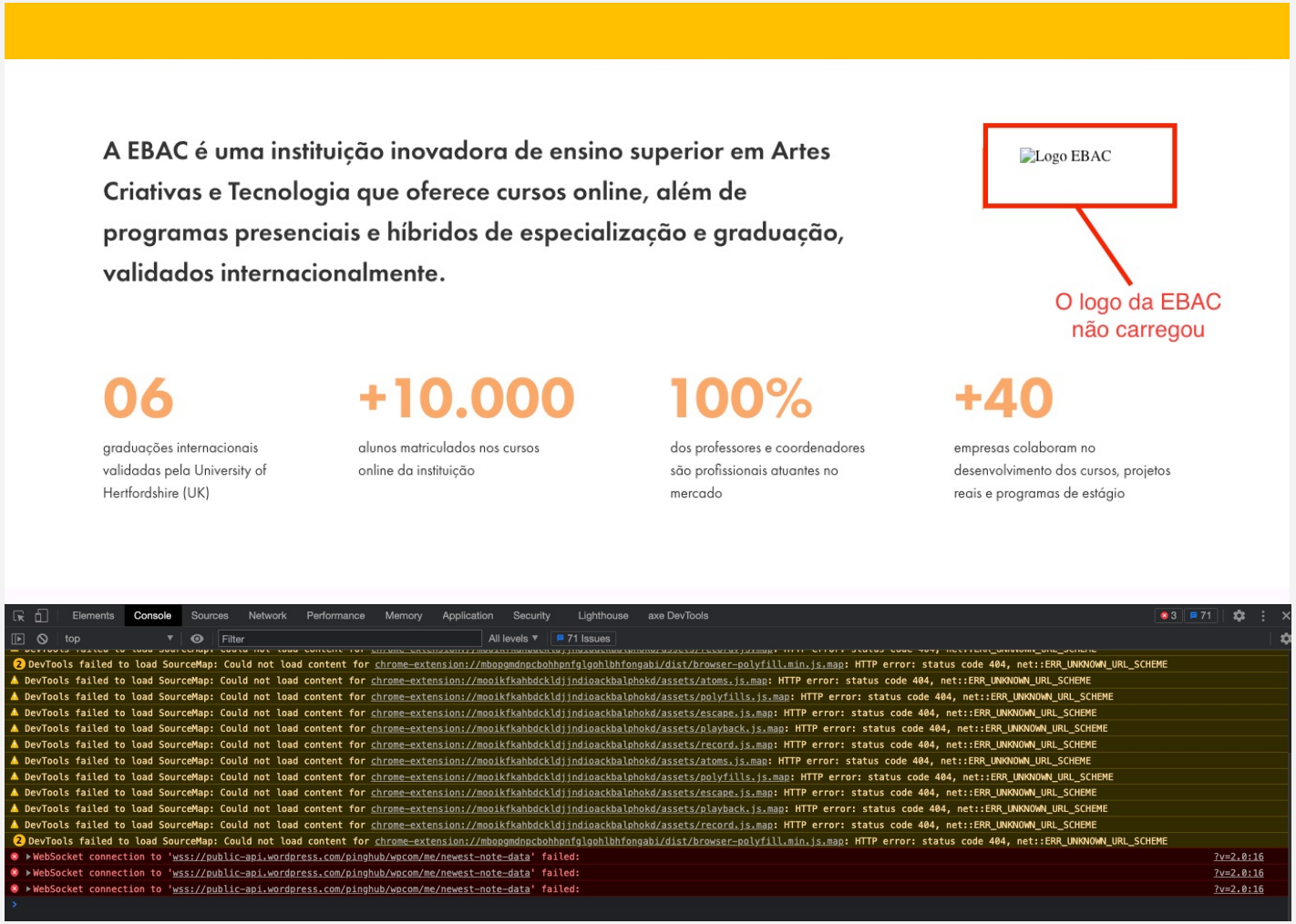
Opcionais:

- Gravidade (crítica, principal, mínimo, trivial)
- Prioridade (alta, média, baixa)
- Criado por:
- Atribuído para:

Evidências

A prova visual pode ser:

- Imagens, com as anotações;
- Vídeos e Gif's;
- Log da "ferramenta de desenvolvedor", como console, Network, Performance, etc.



escola britânica de artes criativas & tecnologia

Exemplos de bug report

Título:	Logo da EBAC não está aparecendo
Ambiente/ URL:	Produção: https://ebaconline.com.br/qualidade-de-software
Evidências:	
Etapas para reproduzir:	1- Acessar o link do curso Qualidade de Software; 2- Rolar a página até a sessão “Sobre a EBAC”
Resultado e comportamento esperado:	Aparece uma imagem com um link quebrado e **deveria** aparecer o logotipo da EBAC: escola britânica de artes criativas & tecnologia
Criado por:	Fábio Araújo
Atribuído para:	Ernesto Barbosa

escola britânica de artes criativas & tecnologia

Review: O que você aprendeu

- O que é teste de software e sua importância
- Tipos de testes e Fases de teste
- O que é garantia da qualidade e sua empregabilidade
- Bugs e diferenças entre erro, defeito e falha
- Como reportar e acompanhar um bug

Referências

- https://bstqb.org.br/b9/doc/syllabus_ctfl_2018br.pdf
- https://istqb-glossary.page/pt/
- https://birdeatsbug.com/blog/bug-report-template#rec151428431
- https://testlio.com/blog/the-ideal-bug-report/