

# Início rápido em teste e QA - Seção 2

## História e Evolução dos Testes:

- **Charles Babbage**, na década de 40, criou a primeira máquina de somar. Sua colaboração com a filha de Lord Byron ajudou a corrigir falhas no projeto.
- **Hollerith** foi o inventor das primeiras máquinas de cartão de ponto, tecnologia usada, entre outras coisas, para decifrar a **Máquina Enigma**, Hollerith também foi um dos fundadores da empresa **IBM**.
- **Alan Turing**, um dos maiores nomes da computação, enfrentou muito preconceito. Sua contribuição foi essencial para a teoria dos testes e da computação.
- **Grace Hopper**, importante para os testes de software, foi responsável por identificar o primeiro "bug", quando um inseto causou uma falha no computador com lâmpadas.

## Importância dos Testes:

- A regra de **Myers** destaca que é crucial encontrar os defeitos o quanto antes, pois corrigir um erro depois é muito mais caro.
- Nos anos 2000, o Brasil começa a se aprofundar nos materiais sobre testes, com **Ermenson Rios** trazendo conhecimentos internacionais para o país.

## Impactos dos Bugs:

- **Empresas**: Atrasos, perda de confiança e prejuízos financeiros.
- **Governo**: Vulnerabilidades em informações podem resultar em decisões incorretas, afetando desde a economia até vitórias militares.
- **Meio Ambiente**: Alertas atrasados causam desperdício de recursos e poluição.
- **Pessoas**: Risco à saúde, perda de direitos e até risco de vida.

## Fundamentos do Teste (Conforme ISQTB):

- **Teste demonstra a presença de defeitos**, mas não pode garantir que o sistema está sem falhas.
- **Testar tudo não é viável**: Testes focam em riscos e prioridades, e não em exaustão total.
- **Quanto mais cedo o teste, maior o retorno**: Encontrar defeitos cedo reduz custos e melhora a qualidade.
- **Distribuição dos defeitos**: Poucos módulos concentram a maior parte dos defeitos, e é perigoso corrigir um defeito e gerar outro.
- **Testes são diferentes conforme o contexto**: Entender o cliente e usuário é essencial.
- **Ilusão de ausência de erros**: Encontrar e corrigir defeitos não é útil se o sistema não atender às expectativas do usuário. Exemplo: Microsoft Windows vs. IBM OS/2.

## Teste e Qualidade:

- **Lições aprendidas** em projetos anteriores ajudam a aprimorar processos e evitar a reincidência de erros.
- **Integração de testes** em todas as fases do desenvolvimento garante a qualidade.
- **Software é feito por pessoas, para pessoas**: Erros humanos geram defeitos que afetam a qualidade do software.

## Padrões de Qualidade (IEC/ISO 25010):

- **Funcionalidade**: Adequação, completude e correção.
- **Usabilidade**: Reconhecimento, aprendizabilidade, operabilidade e acessibilidade.
- **Desempenho**: Eficiência em termos de tempo e uso de recursos.
- **Manutenibilidade**: Modularidade, reusabilidade e testabilidade.
- **Portabilidade**: Adaptabilidade e facilidade de instalação.
- **Segurança**: Confidencialidade, integridade e autenticidade.

**Testes Automatizados:**

- **Integração contínua:** Testes automatizados são realizados diariamente, identificando falhas rapidamente e garantindo feedback imediato sobre a qualidade do código.
- **Testes de regressão automatizados:** Garantem que mudanças não afetem funcionalidades existentes, permitindo que os testadores se concentrem em novas funcionalidades.
- **Testes de confirmação:** Refazem os testes de defeitos corrigidos para garantir que a solução seja eficaz.