**Resumo Seção 02**

Como surgiram os testes?

O defeito é uma manifestação da existência humana, de acordo com essa condição surgiram os testes.

Em 1940, foi criada uma máquina que substituiria os ábacos, Charles criou um sistema com furos que representavam as operações. Ele conheceu Ada Lovelace, a primeira programadora a qual ajudou muito no projeto, e tem registros de uma carta que ela informava que havia encontrado um defeito.

No final do século 19, ele deu continuidade à ideia de Charles e a transformou em uma máquina que economizava muito tempo, ele deu início também a comunicação por furos no cartão, no qual registravam dados.

Não usaram somente para o bem essas tecnologias, os nazistas utilizaram para enviar mensagens secretas.

“Os defeitos cobram juros”- Nunca deixe para corrigir algo depois.

Importância dos testes:

Tem forte relação com os danos dos bugs, que podem causar grande impacto no projeto.

Possíveis danos: prejuízo financeiro e de imagem.

# Os 07 fundamentos do Teste (ISTQB)

1. O teste demonstra a presença de defeito, mas não pode provar que eles não existam (um teste reduz a probabilidade que os defeitos permaneçam em um software).
2. Teste exaustivo não é possível (deve priorizar o que é mais importante).
3. Teste antecipado (começar o mais breve possível).
4. Agrupamento de defeitos (bugs distribuídos de forma heterogênea).
5. Paradoxo do Pesticida (para revisar e atualizar frequentemente).
6. Cada teste depende do contexto.
7. O software deve atender a necessidade do usuário.

Teste X Qualidade

Teste avalia o produto e QA avalia os processos.

Erros (enganos), defeitos (bugs), falha (só existe quando o defeito é executado).

# IEC/ISO 25010

Conhecida como SQuaRE (Systems and software Quality Requirements and Evaluation)

Essa norma se divide em 8 características:

1. AF: Adequação funcional (funcionalidade):
   1. Completude (se a funcionalidade está completa)
   2. Correção (Dar um resultado certo)
   3. Apropriado (A forma de entregar a informação deve estar apropriada)
2. Usabilidade:
   1. Reconhecibilidade (facilitar para que o cliente reconheça os elementos e comportamentos)
   2. Aprendizibilidade (facilitar o aprendizado do usuário)
   3. Operabilidade (facilidade de operação e navegação)
   4. Proteção contra erro do usuário
   5. Estética (da interface do Usuário)
   6. Acessibilidade (facilitar o acesso a todas as pessoas)
3. Compatibilidade
   1. Coexistência
   2. Interoperabilidade (facilidade de comunicação - enviar/receber dados)
4. Confiança
   1. Maturidade (perceber e prevenir a falha antes que aconteça)
   2. Disponibilidade (manter-se a disposição de usuários e sistemas)
   3. Tolerância a falhas (perceber e compensar as falhas em tempo real)
   4. Recuperabilidade (Recuperar-se de falhas e travamentos)
5. Eficiência (de desempenho)
   1. Comportamento em relação ao tempo
   2. Utilização de Recursos
   3. Capacidade
6. Manutenibilidade
   1. Modularidade (organizada em módulos)
   2. Reusabilidade (facibilidade de reutilizar)
   3. Analisabilidade (facilidade de analisar)
   4. Modificabilidade (facilidade de modificar)
   5. Testabilidade (é viável testar)
7. Portabilidade
   1. Adaptabilidade
   2. Instabilidade (facilidade de instalar/desinstalar e ligar/desligar recursos)
   3. Substituibilidade
8. Segurança
9. Confidencialidade
10. Integridade (se houver mudança deve ficar registrado quem mudou e quando)
11. Não repúdio
12. Responsabilidade (auditável - que permite prestar contas)
13. Autenticidade

## Teste automatizado:

Existem tipos de testes que só podem ser feitos com automação.

As estruturas de programação estão sempre mudando, é necessário ter um planejamento, criar testes todos os dias, para assim, se algo estiver errado, poderá ser consertado.

* Testes de regressão automatizado: para dar visibilidade do que está funcionando no software.
* Teste de confirmação: Para garantir que um defeito que foi reportado foi removido
* Teste de regressão generalizada: Para garantir que as mudanças utilizadas para corrigir o defeito não “quebrou” nada em nenhum módulo.

## Resumo:

Sobre o surgimento dos testes de software, primeiramente defeito é uma manifestação da existência humana. Há registros de uma carta de Ada Lovelace (primeira programadora) reportando um primeiro defeito encontrado no projeto.

Testar os softwares tem importância relacionada com os danos dos bugs, que podem causar grande impacto no projeto.

A IEC/ISO 25010 estabelece um padrão que devemos seguir ao realizar testes de software, do qual possui 8 características: Adequação funcional (funcionalidade), usabilidade, compatibilidade, confiança, eficiência (de desempenho), manutenibilidade, portabilidade e segurança.

## Link do Documento:

<https://bremeirah.atlassian.net/wiki/spaces/SOFTWAREDE/pages/edit-v2/3080257?draftShareId=481ed2c4-14cf-490d-85de-436913483b62>