Checklist de Auditoria de Qualidade de Software - ISO/IEC 25010

Equipe: Erick Clepton da Silva Jordão 1352315903

Este checklist foi desenvolvido com base na norma ISO/IEC 25010:2011 e visa orientar auditorias de qualidade de software, avaliando características fundamentais do produto. Deve ser utilizado como guia para verificação sistemática e registro de evidências.

**Passo a Passo para Aplicação do Checklist:**

* Análise de código-fonte,
* Uso de ferramentas automatizadas,
* Entrevistas com desenvolvedores e usuários,
* Testes manuais.

- Registrar evidências para cada item, com capturas de tela, métricas, logs e exemplos.

- Pontuar ou classificar os critérios (ex: conforme grau de conformidade: 100%, (50%) parcial, (0%) Não conformidade).

- Ao final, elaborar um relatório consolidado com as principais falhas, pontos fortes, recomendações e plano de ação.

## 1. Funcionalidade (Functional Suitability)

* 1. [ ] O sistema atende a todos os requisitos funcionais especificados?
  2. [ ] As funcionalidades são completas para os objetivos do usuário?
  3. [ ] Os resultados fornecidos são corretos em todos os casos de uso?
  4. [ ] Há consistência na entrega de funcionalidades em diferentes contextos?

## 2. Confiabilidade (Reliability)

* 1. [ ] O sistema continua funcionando corretamente após falhas parciais?
  2. [ ] Há mecanismos de recuperação e tolerância a falhas?
  3. [ ] Os testes automatizados cobrem casos de falhas e exceções?
  4. [ ] Existe análise de disponibilidade e continuidade?

## 3. Usabilidade (Usability)

* 1. [ ] A interface é intuitiva e fácil de usar?
  2. [ ] Os usuários entendem facilmente como interagir com o sistema?
  3. [ ] O sistema é acessível para usuários com diferentes necessidades (ex: acessibilidade)?
  4. [ ] Foram realizados testes com usuários reais?

## 4. Eficiência de Desempenho (Performance Efficiency)

* 1. [ ] O tempo de resposta atende aos requisitos?
  2. [ ] Há uso eficiente de recursos (CPU, memória, armazenamento)?
  3. [ ] Existem métricas de desempenho coletadas em ambiente real ou testes de carga?
  4. [ ] O sistema se comporta adequadamente sob carga?

## 5. Segurança (Security)

* 1. [ ] O sistema protege dados contra acessos não autorizados?
  2. [ ] Existem mecanismos de autenticação e autorização robustos?
  3. [ ] A aplicação está protegida contra vulnerabilidades conhecidas (ex: OWASP Top 10)?
  4. [ ] Há registro e auditoria de ações críticas?

## 6. Manutenibilidade (Maintainability)

* 1. [ ] O código é modular e de fácil leitura?
  2. [ ] Há documentação atualizada e compreensível?
  3. [ ] O projeto adota boas práticas de versionamento e controle de mudanças?
  4. [ ] O sistema é fácil de testar e atualizar?

## 7. Portabilidade (Portability)

* 1. [ ] O sistema pode ser executado em diferentes plataformas (ex: sistemas operacionais, navegadores)?
  2. [ ] A instalação e configuração são simples?
  3. [ ] O sistema é compatível com diferentes resoluções, tamanhos de tela ou dispositivos?
  4. [ ] Há automação para deploy em múltiplos ambientes?

## 8. Qualidade em Uso (quando aplicável)

* 1. [ ] O usuário consegue atingir seus objetivos com o sistema?
  2. [ ] O sistema permite atingir os objetivos com o menor esforço?
  3. [ ] Os usuários estão satisfeitos com o uso do sistema?
  4. [ ] O sistema evita riscos físicos, financeiros ou de segurança ao usuário?