**FUNDAMENTOS DE QUALIDADE E DESENVOLVIMENTO DE SOFTWARE**

- Engenharia de software != programação. Todo programador pode ser considerado um engenheiro de software, mas a Engenharia de software é muito além de só programar.

**\*Tipos básicos de engenheiros de software:**

**1. Frontend**: Parte visual da aplicação e interação com o usuário

**2.Backend**: Processamento de dados e regras de negocio

**3. Quality Assurance:** Validações e verificações de funcionalidade, gestão de defeitos e processos de qualidade

**4. DevOps/SER**: Cultura e processos de operações para garantir a confiabilidade, monitoramento, desempenho e pipelines de desenvolvimento

-Analista de Qualidade de Software = Analista de teste = QA = Engenheiro de qualidade de software

**\*QUAIS OS OBJETIVOS DE TER UM QA NO TIME?**

-Redução de custos e retrabalho

-Identificação de problemas

-Entrega de produtos com qualidade

-Melhora na satisfação do cliente

-Melhora na estimativa dos projetos

-Otimização na rotina de trabalho

**FUNÇÕES**: Cultura de qualidade, testes automatizados, gerenciamento de defeitos de software, auxílio a desenvolvedores, testes manuais, auxílio na compreensão de histórias para validar e verificar os requisitos

**Obs**: Existem perfis diferentes de QA: Alguns vão trabalhar mais com gerenciamento, outros mais com testes automatizados (de UI/Interface, de APIs, de Performance/desempenho, etc.), outros mais com testes manuais.

**CERTIFICAÇÕES DO QA**

\*ISQTB: Esquema de certificações internacionais para desenvolvimento da carreira de quem trabalha com testes de software

\*BSTQB: Certificações brasileiras

**PENSANDO NAS CATACTERISTICAS DO SOFTWARE:**

1. **FUNCIONALIDADES**: São apropriadas para o negócio do cliente? Foram implementadas corretamente? Como estão sendo guardados os dados? O sistema é responsivo?
2. **CONFIABILIDADE**: Como o software se comporta mediante condições de falhas específicas? Quão frequente falha? Qual tempo de recuperação?
3. **USABILIDADE**: Os usuários entendem o software? Qual esforço para essa compreensão?
4. **EFICIÊNCIA**: O time de desenvolvimento segue boas práticas? A arquitetura do projeto foi pensada para ser eficiente?
5. **MANUTENIBILIDADE**: Quão difícil é encontrar um problema na aplicação e corrigi-lo? Qual o esforço para modificar o código?
6. **PORTABILIDADE**: O sistema se adapta a mudanças no ambiente? Quão difícil é migrar um componente de sistema?

**ERROS QUE O QA PODE COMETER:**

1. Falhas na análise de uma ocorrência
2. Medo de fazer perguntas
3. Automações falhas e sem padrão
4. Esquecer do usuário
5. Culpar outros por defeitos/bugs
6. Não ter a visão do que ocorre em produção
7. Não se importar com processos técnicos do desenvolvimento