

Dentro dos diversos tipos de revisão de software já inventados e utilizados, alguns temas e pilares nunca podem ser deixados de lado. Pilares estes que serão desenvolvidos e explicados ao longo do texto a seguir.

**Revisão de código:** A revisão de código se mostra essencial uma vez que o mesmo é toda a alma responsável pelo funcionamento do mesmo, sendo assim revisões de funcionalidade, eficiência e manutenibilidade são indispensáveis.

**Revisão de Design:** O design e interface de um software é a linha de frente de contato com um usuário, dessa forma, o mesmo deve sempre focar em ser facilmente entendido e funcional para o público alvo e no contexto previsto.

**Revisão de Segurança:** De modo algum o software pode permitir que seus usuários sejam lesados durante seu uso ou que seu uso seja lesado pelos mesmos. Seguindo esse raciocínio é de suma importância verificar as possíveis vulnerabilidades de um software e as maneiras de lidar com as mesmas.

**Revisão de Performance:** Um software completo e funcional de nada serve caso não seja otimizado de maneira correta, já que nesse caso não seria sustentável o uso do mesmo. Pensando nisso, focar em melhorias constantes de funcionamento e eficiência do mesmo de acordo com as necessidades dos usuários é um ponto relevante a todo momento.

**Revisões Com o Cliente:** Apesar de muitas vezes após a solicitação do software os desenvolvedores esquecerem de solicitar esse feedback constante ele é diretamente responsável pela confiabilidade de longo prazo do projeto em atender as necessidades e expectativas do cliente.

**Revisões Internas:** Assim como com o cliente, as revisões internas também são importantes para direcionar de maneira mais eficiente as “forças” dos responsáveis pelo projeto, resultando assim num projeto muito mais fluido e adaptativo à longo prazo.

**Pentes Finos:** Mesmo com um funcionamento correto e as expectativas em dia, é sempre bom manter os testes contínuos e uma visão crítica sob o software, buscando assim evitar futuros problemas e até mesmo descobrir bugs antes mesmo que eles afetem algum usuário real.

**Conclusão:** Dentro da revisão de um único software temos quantidades enormes de caminhos a seguir, contudo, o mais importante é sempre manter o equilíbrio entre eles,

com constância e pensando no funcionamento de longo prazo do projeto (mesmo quando não esperado). Já que resoluções de curto apesar de no momento parecerem uma boa opção pela praticidade podem futuramente causar maiores problemas caso não sejam trabalhadas efetivamente.

Turma CCP1AN-MCD3-25070036

CURSO: CIÊNCIA DA COMPUTAÇÃO

UNIDADE: USJT – MOOCA

email: [RA@ulife.com.br](mailto:RA@ulife.com.br)

Bruno Dias - 82311358

Emilio Palacios - 823112595

Gustavo Baeza - 82319972

Matheus Chuang - 823165173

Raí Joia – 82318841

12/03/2025