

## SUMÁRIO

QUESTIONÁRIO 01 .....	2
1 - Defina Qualidade de Software.....	2
2 - Defina processo de software e comente as características de um bom processo de software.....	3
3 - Comente sobre a categorização das métricas.....	5
4 - Quando falamos de revisões de software, o que é importante que o engenheiro considere no planejamento?.....	7

## QUESTIONÁRIO 01

### 1 - Defina Qualidade de Software.

Qualidade de software possui várias definições, vejamos algumas:

- Qualidade de software é a conformidade a requisitos funcionais e de desempenho que foram explicitamente declarados, a padrões de desenvolvimento claramente documentados, e a características implícitas que são esperadas de todo software desenvolvido por profissionais.
- Um produto de software apresenta qualidade dependendo do grau de satisfação das necessidades dos clientes sob todos os aspectos do produto.
- Qualidade é a totalidade de características e critérios de um produto ou serviço que exercem suas habilidades para satisfazer as necessidades declaradas ou envolvidas.

- Qualidade é a totalidade das características de uma entidade, que lhe confere a capacidade de satisfazer necessidades explícitas e implícitas.

## **2 - Defina processo de software e comente as características de um bom processo de software.**

Um processo de software pode ser definido como um conjunto de atividades, métodos, práticas e transformações que as pessoas usam para desenvolver e manter o software e os produtos associados (por exemplo: planos de projeto, documentos, código, casos de teste e manuais de usuário).

Um processo de software envolve um grande conjunto de elementos, tais como objetivos organizacionais, políticas, pessoas, comprometimentos, ferramentas, métodos, atividades de apoio e as tarefas da engenharia de software. Para que o processo de software seja eficiente ele precisa ser constantemente avaliado, medido e controlado.

Quando o processo é eficiente ele possui as seguintes características:

- O processo continua a despeito de problemas inesperados (Robustez).
- Rapidez na produção do sistema (Velocidade).
- O processo é aceito por todos os envolvidos nele (Aceitabilidade)
- Os erros do processo são descobertos antes que resultem em erros no produto (Confiabilidade)
- O processo evolui para atender alterações de necessidades organizacionais (Manutenibilidade)
- O processo é compreendido (usualmente através de documentação e de treinamento), utilizado, vivo e ativo.
- O processo é bem controlado – a fidelidade ao processo é objeto de auditoria e de controle.

- Medidas do produto e do processo são utilizadas.
- Os papéis e responsabilidades no processo estão claros ao longo de todo o projeto e por toda a organização.

### **3 - Comente sobre a categorização das métricas.**

Com relação à categorização das métricas de software podemos citar:

- Métricas diretas (fundamentais ou básicas): medida realizada em termos de atributos observados (usualmente determinada pela contagem). Ex.: custo, esforço, números de linhas de código, capacidade de memória, números de páginas, números de diagramas, etc.
- Métricas indiretas (derivadas): medidas obtidas a partir de outras métricas. Ex.: complexidade, eficiência, confiabilidade, facilidade de manutenção.

- Métricas orientadas a tamanho: são medidas diretas do tamanho dos artefatos de software associados ao processo por meio do qual o software é desenvolvido. Ex.: esforço, custo, números de KLOC\*, números de páginas de documentação, números de Erros.
- Métricas orientadas por função: consiste em um método para medição de software do ponto de vista do usuário, determinando de forma consistente o tamanho e a complexidade de um software.
- Métricas de produtividade: concentram-se na saída do processo de engenharia de software. Ex.: nº de casos de uso/iteração.
- Métricas de qualidade: Oferecem uma indicação de quanto o software se adequa às exigências implícitas e explícitas do cliente. Ex.: erros/fase.

- Métricas técnicas: concentram-se nas características do software e não no processo por meio do qual o software foi desenvolvido. Ex.: complexidade lógica e grau de manutenibilidade.

### \* KLOC (milhares de linhas de código)

KLOC significa "thousands of lines of code" (milhares de linhas de código). É uma medida do tamanho de um programa de software. A forma mais comum de medir o tamanho de um programa de software é pelo número de linhas de código (LOC). KLOC é simplesmente uma unidade de medida que é igual a 1.000 linhas de código.

KLOC é frequentemente utilizado como uma forma de estimar a quantidade de esforço necessário para desenvolver um programa de software.

## 4 - Quando falamos de revisões de software, o que é importante que o engenheiro considere no planejamento?

Devem ser consideradas as seguintes questões:

- quem participa?
- qual informação é requerida antes da revisão?

- quais pré-condições que devem ser satisfeitas antes que a revisão possa ser conduzida?
- Como Organizar?
- Gerar checklists ou outra indicação do que deve ser coberto na revisão;
- Determinar as condições de término ou critérios que devem ser satisfeitos para que a revisão termine;
- Gerar registros e documentos que devem ser produzidos