

1. Adequação Funcional

- 1.1. Nesse atributo o projeto deve determinar as maneiras mais eficazes e eficientes de se realizar cada funcionalidade, além de garantir que todos os problemas relacionados ao software podem ser resolvidos por ele.
- 1.2. Acredito que quanto mais intuitivo for o software maior a pertinência funcional e apesar disso fazer parte da adequação funcional, é necessário que a saída final, após os testes, corresponda ao esperado.

2. Eficiência de Desempenho

- 2.1. Para determinar a eficiência do desempenho é necessário estabelecer requisitos e condições específicas para que a partir disso se estabeleça uma comparação que determine se o desempenho condiz com os requisitos especificados.
- 2.2. Comparando o projeto com os requisitos especificados, como tempo, quantidade e tipo de recursos utilizados e a capacidade máxima. Caso o projeto atenda a todos, podemos afirmar que o desempenho alcançou o nível máximo.

3. Compatibilidade

- 3.1. É necessário determinar em qual ambiente operacional o software será executado e com quais programas ele precisará coexistir ou se comunicar.
- 3.2. Testando se o mesmo é capaz de coexistir com os programas determinados no projeto, dentro do ambiente operacional especificado, sem que esse ou os outros programas sofram prejuízos como perda de desempenho.

4. Usabilidade

- 4.1. Deve-se reconhecer o cliente a qual o produto destina-se, para que dessa forma o projeto consiga determinar as maneiras mais fáceis desse ser compreendido, aprendido e operado. Conhecendo as características gerais do cliente (idade, sexo, nível escolar, etc) torna-se mais fácil tornar o produto mais atrativo.
- 4.2. Podemos pesquisar por produtos que já estão no mercado e que se assemelham ao nosso projeto, buscando seus pontos fortes e fracos na visão dos usuários. Além disso, pode-se disponibilizar ao cliente alvo uma versão de teste para que consigamos perceber se o projeto condiz com nossa perspectiva a respeito do cliente.

5. Confiabilidade

- 5.1. É preciso determinar em quais condições o software irá operar em determinado intervalo de tempo. Dessa forma, o projeto pode prever possíveis falhas sob as quais o software deverá apresentar tolerância e capacidade de recuperação de dados e de seu estado.
- 5.2. Colocando o software sob condições falhas, a fim de observar seu comportamento. Para isso pode-se recorrer a ferramentas específicos.

6. Segurança

- 6.1. Nesse atributo o projeto determina aspectos como permissão ou não de usuário, qual o risco ou probabilidade de invasão, como será a comprovação de

autenticidade do usuário e garantia de privacidade, de acordo como a necessidade do cliente.

6.2. Testando o projeto em condições de risco ou de invasões. Para isso é possível recorrer a ferramentas específicas e contratar pessoas para tentar invadir ou quebrar as técnicas de segurança dispostas pelo software.

7. Manutenabilidade

7.1. O projeto deve conter regras específicas para a construção de um algoritmo claro e de fácil entendimento para todos, estabelecendo documentações e comentários que permitam sua compreensão. Além disso, o projeto deve determinar quais as técnicas principais do algoritmo que permitam a modularidade, reusabilidade, analisabilidade e modificabilidade do software. O projeto deve determinar também os critérios de testes mais adequados.

7.2. Colocar o software em teste para determinar se os critérios estabelecidos no projeto foram cumpridos.

8. Portabilidade

8.1. Determinar sob quais hardwares ou softwares o programa será executado e instalado. Para isso o projeto deve observar os hardwares e sistemas operacionais mais utilizados pelos prováveis usuários.

8.2. Testando um protótipo do software, observando seu comportamento ao ser instalado/desinstalado e se é possível fazer isso em todos os ambientes pré-determinados no projeto.