

Construção de Software

Prof. Rubens de Castro Pereira

Lista de Verificação – Requisitos

Referência:

- Code Complete – Um guia prático para a construção de software, 2ª ed., Steven McConnell, Bookman – Cap. 3.

Lista de verificação de requisitos com perguntas a serem feitas sobre os requisitos do projeto.

1. Requisitos Funcionais Específicos

- 1.1. Todas as entradas do sistema estão especificadas, incluindo sua origem, precisão, intervalo de valores e frequência?
- 1.2. Todas as saídas do sistema estão especificados, incluindo seu destino, precisão, intervalo de valores, frequência e formato?
- 1.3. Todos os formatos de saída estão especificados para as páginas da Web, relatórios, etc?
- 1.4. Todas as interfaces externas de *hardware* e *software* estão especificadas?
- 1.5. Todas as interfaces de comunicação externas estão especificadas, incluindo *handshake* (protocolo inicial de comunicação entre dois computadores), verificação de erro e protocolos de comunicação?
- 1.6. Todas as tarefas que o usuário deseja executar estão especificadas?
- 1.7. Os dados usados em cada tarefa e os dados resultantes de cada um estão especificados?

2. Requisitos Não-Funcionais Específicos (qualidade)

- 2.1. O tempo de resposta esperado, do ponto de vista do usuário, está especificado para todas as operações necessárias?
- 2.2. Outras considerações de cronometragem estão especificadas, como tempo de processamento, taxa de transferência de dados e desempenho do sistema?
- 2.3. O nível de segurança está especificado?
- 2.4. A confiabilidade está especificada, incluindo as consequências de falha do *software*, as informações fundamentais que precisam ser protegidas contra falhas e a estratégia de detecção e recuperação de erros?
- 2.5. A memória de máquina e o espaço livre em disco mínimos estão especificados?
- 2.6. A manutenibilidade do sistema está especificada, incluindo sua capacidade de se adaptar às mudanças na funcionalidade específica, às mudanças no ambiente operacional e em suas interfaces com outro *software*?
- 2.7. A definição de sucesso está incluída? E a de falha?

3. Qualidade dos Requisitos

- 3.1. Os requisitos estão redigidos no idioma do usuário?
- 3.2. Cada requisito evita incompatibilidade com outros requisitos?
- 3.3. Os compromissos aceitáveis entre atributos concorrentes estão especificados – por exemplo, entre robustez e correção?
- 3.4. Os requisitos evitam a especificação do *design* (desenho)?
- 3.5. Os requisitos estão em um nível de detalhe totalmente consistente? Algum requisito deve ser especificado com mais detalhes? Algum requisito deve ser especificado com menos detalhes?
- 3.6. Os requisitos são claros o suficiente para serem transformados em grupos independentes para construção e ainda serem entendidos? Os desenvolvedores também pensam assim?
- 3.7. Cada item é relevante para o problema e para sua solução? Cada item pode ser remontado à sua origem no ambiente do problema?
- 3.8. Cada requisito pode ser testado? Será possível, por meio de testes independentes, determinar se cada requisito foi satisfeito?
- 3.9. Todas as alterações possíveis nos requisitos estão especificadas, incluindo a probabilidade de alteração?

4. Integralidade dos Requisitos

- 4.1. Onde não há informações disponíveis antes que o desenvolvimento comece, as áreas de imperfeição estão especificadas?
- 4.2. Os requisitos estão completos no sentido de que, se o produto satisfizer cada requisito, ele (produto) será aceitável?
- 4.3. Você está à vontade com todos os requisitos? Você eliminou os requisitos impossíveis de implementar e que você havia incluído apenas para contentar seu cliente ou seu chefe?