

Modelo para Documento de

Especificação de Requisitos de Software

Prof. Dr. Juliano Lopes de Oliveira

(Baseado na norma IEEE Std 830-1993 - Recommended Practice for Software Requirements Specifications)

A boa organização lógica do documento e a sua redação correta são condições essenciais para que ele se torne útil na prática. Algumas recomendações neste sentido, antes de examinarmos o conteúdo da especificação de requisitos:

- Os pontos básicos que devem ser definidos no documento são: **funcionalidade**, **interfaces externas**, **restrições**, critérios de **qualidade** e de **desempenho**. O documento deve definir quais funções são executadas sobre que dados para produzir quais resultados em qual local e para quem.
 - ◊ Toda a equipe de desenvolvimento (usuários, técnicos e gerentes) deve participar da elaboração do documento, que deve definir corretamente todos os requisitos do software, mas sem descrever detalhes de projeto ou de implementação (a menos que seja realmente uma restrição a ser considerada). O documento descreve as necessidades do produto de software, e não o processo de produção.
 - ◊ Gráficos e diagramas podem ajudar bastante na especificação. Porém, não se trata de um documento de projeto: não se faz particionamento de módulos internos, escolha de estruturas de dados, e outras atividades de projeto. Tampouco se faz atividades de planejamento de software, tais como cálculo de custos, cronograma de entrega, critérios de validação, etc.
- É preciso garantir que o documento define estes pontos de maneira correta, não ambígua, completa, consistente, classificada de acordo com a importância dos requisitos, verificável, modificável e rastreável.
 - ◊ Um requisito **correto** é um requisito que o software deve atender. Um requisito **não ambíguo** permite somente uma interpretação.
 - ◊ Uma especificação de requisitos é **completa** se contém todos os requisitos importantes, definindo todas as definições de respostas do software para todas as classes imagináveis de entrada de dados em todas as situações imagináveis. Os termos e unidades de medidas usadas na especificação também devem ser bem definidos. Todas as figuras, tabelas e diagramas devem ter uma legenda completa.

- ◇ Uma especificação de requisitos deve ser **consistente** com documentos "maiores" (leis, por exemplo), e também internamente consistente (sem conflitos internos).
 - ◇ O documento deve especificar todas as necessidades que poderiam ser atendidas pelo software, **classificando-as de acordo com a importância** em: Essenciais (atendimento obrigatório), Desejáveis (atendimento fortemente recomendado), e Opcionais (seria bom atender).
 - ◇ Um requisito deve ser **verificável**, ou seja, deve haver uma maneira de garantir que o requisito é atendido pelo software (exemplo: a função deve usar no máximo 2Mbytes de memória; contra-exemplo: a interface deve ser *amigável*).
 - ◇ O documento deve ser **modificável**: referências cruzadas entre seções são absolutamente essenciais para evitar redundância, melhorar a legibilidade do documento e facilitar sua modificação. Utilizar uma notação e uma linguagem consistente ao longo de todo o documento é fundamental. Utilizar índice de conteúdo, lista de tabelas e lista de figuras contribui para facilitar a leitura e a modificação do documento.
 - ◇ Cada requisito deve ser **rastreável**: um requisito deve identificar sua origem (*backward traceability* - documento ou ata de reunião, por exemplo) e deve ser univocamente identificado, de forma que documentos posteriores possam referenciá-lo (*forward traceability*).
- É preciso levar em conta que o documento pode evoluir ao longo do tempo. Um processo formal deve ser estabelecido para identificar, controlar e rastrear as modificações efetuadas.

Conteúdo do Documento

1. Introdução

1.1. Propósito (do documento)

Estabelecer o objetivo do documento e o público alvo.

1.2. Escopo (do software)

Definir o(s) produto(s) especificado(s) no documento: nome, objetivos, limites de atuação, benefícios esperados, ...

1.3. Visão Geral (do documento)

Descrever a organização e o conteúdo de cada seção do documento

2. Informações de Suporte

2.1. Definições

Descrever todas as siglas e abreviações usadas no documento, bem como os termos necessários ao seu entendimento.

2.2. Referências

Listar todos os documentos citados no documento (normas, livros, manuais de outros produtos, ...).

3. Descrição Geral do Software

Lembre-se que todos os itens desta seção serão detalhados na próxima seção (a idéia aqui é apresentar uma visão geral, sem detalhes).

3.1. Perspectiva de Produto

Descrever o contexto em que o software deverá ser executado: interconexões, interfaces externas, operações, sistemas de comunicação, restrições de memória (primária e secundária) e requisitos da plataforma. Um diagrama de arquitetura de software pode ajudar esta definição.

3.2. Funções do Produto

Especificar as principais funções do software, dando uma idéia de como elas cooperam para realizar os objetivos do sistema.

3.3. Características de Usuários

Descrever o perfil de usuário esperado, e as razões para eventuais requisitos de formação educacional, conhecimento técnico ou experiência.

3.4. Restrições

Definir obrigações ou limitações às opções de desenvolvimento, justificando-as (preferencialmente com o apoio de documentos).

3.5. Suposições e Dependências

Definir as hipóteses que são consideradas e que têm influência sobre os requisitos do software. Ex: supõe-se que o SGBD Oracle estará disponível; caso contrário os requisitos de portabilidade de bases de dados deverão ser revistos.

3.6. Requisitos Futuros

Citar os requisitos identificados mas que não serão contemplados nesta versão do software.

4. Requisitos Específicos

4.1. Interfaces Externas

Definir detalhes de intercâmbio de dados com entidades externas:

- Usuário: todos os aspectos de IHC;
- Hardware: descrever o relacionamento do software com a plataforma alvo (dispositivos suportados, família de processador, ...);
- Software: ligações com sistemas (SO, SGBD, outros sistemas aplicativos, ...);
- Sistemas de Comunicação: APIs, protocolos de rede, software de suporte a distribuição (ex: CORBA), considerações sobre inter/intranet,...

4.2. Funções

Para cada função descrever:

- entradas válidas e inválidas;
- operações realizadas;
- resultados ou saídas aceitáveis;
- tratamento de exceções.

Considerar não apenas as funções específicas do software, mas também as de suporte (por exemplo, administração do sistema, backups, auditoria, etc).

É importante organizar a descrição das funções, definindo esta organização no início da seção: por modo, por classe de usuário, por objeto, por hierarquia funcional, ...

4.3. Requisitos de Desempenho

Estáticos: quantidade de terminais na rede, conexões simultâneas, volumes de dados armazenados, número de threads, ...

Dinâmicos: transações por segundo, tempo de resposta em condições normais e de pico, memória usada por um processo, ...

4.4. Requisitos de Bases de Dados Lógicas

Descrição dos dados: facilidades de acesso aos dados, frequência de uso, restrições de integridade, persistência, ...

4.5. Restrições de Projeto

Pontos que não podem ser modificados devido a: padrões, regras, normas, leis, limitações de hardware, ...

4.6. Atributos do Sistema de Software

Fatores de Qualidade: portabilidade, segurança, manutenibilidade, ...

5. Apêndices

5.1. Comentários Adicionais

Descrever ou comentar o processo que levou a cada requisito importante especificado no documento, mostrando as alternativas examinadas e o motivo da escolha.

5.2. Documentos Importantes

Anexar ao Documento de Especificação de Requisitos todos os documentos que serviram de base para sua elaboração, tais como: normas de desenvolvimento de software, leis que justificam decisões de projeto, ...