

Redação de Requisitos em Linguagem Natural

1. A Importância da Estrutura na Documentação de Requisitos

Em sistemas de engenharia, especialmente os de elevada complexidade, a clareza e a consistência da documentação são fundamentais para o sucesso. A ausência de uma estrutura definida pode levar a uma proliferação de documentos inconsistentes e difíceis de gerir, transformando a documentação numa barreira em vez de uma ajuda.

Uma estrutura bem definida ajuda a mitigar este problema, garantindo que a informação essencial é capturada de forma organizada e previsível.

1.1 A Necessidade de um Modelo (Template)

A ausência de um modelo predefinido para a documentação de requisitos concede uma liberdade excessiva às equipas, resultando em documentos muito diferentes de projeto para projeto. Esta inconsistência dificulta a compreensão, a manutenção e a reutilização da informação. A definição de uma estrutura comum, ou template, é uma prática fundamental que se revela particularmente útil em sistemas de alta complexidade, onde o volume de informação pode ser avassalador. Um modelo garante que todos os aspetos críticos do projeto são considerados e documentados de forma sistemática.

2. Princípios Fundamentais para a Redação de Requisitos

2.1 As Regras da Escrita Técnica

A redação de requisitos para sistemas de engenharia deve obedecer às regras fundamentais da escrita técnica, que visam a máxima clareza e a mínima ambiguidade.

- **Linguagem Simples, Clara e Precisa:** O texto deve ser direto e inequívoco, permitindo uma única interpretação.
- **Evitar Figuras de Estilo:** Recursos literários como metáforas devem ser completamente evitados, pois introduzem subjetividade e potencial para interpretações múltiplas.
- **Significado Denotativo:** As palavras devem ser usadas no seu sentido literal e primário, sem dar margem a conotações ou significados alternativos.
- **Estilo Impessoal e Objetivo:** A escrita deve manter um tom objetivo, claro, humilde e educado, focando-se na descrição dos factos e necessidades, sem expressar opiniões pessoais.

2.2 A Regra das Frases Curtas e Simples

As frases devem ser curtas, diretas e afirmativas, utilizando a voz ativa para atribuir responsabilidade de forma clara. Deve-se evitar o uso de frases negativas ou na voz passiva, que podem obscurecer o significado.

2.3 A Importância de um Vocabulário Limitado

Deve-se utilizar um vocabulário limitado e controlado para os conceitos-chave do projeto.

- **Evitar Sinónimos:** O uso de sinónimos para conceitos importantes deve ser evitado, mesmo que seja para tornar o texto menos repetitivo. Por exemplo, se o termo definido é "student", deve-se usar consistentemente "student" em vez de alternar com undergraduate, postgraduate ou scholar, pois essa variação pode criar confusão sobre se os termos se referem a conceitos diferentes.
- **Uso Criterioso de Acrónimos:** Acrónimos e abreviaturas devem ser usados com cuidado. Embora seja aceitável usar expressões já estabelecidas e universalmente compreendidas (ex: BIOS, ERP, GNU, JPEG, MP3, PDF, RAM, SMTP), a criação de novos acrónimos deve ser limitada e sempre acompanhada de uma definição clara.

3. Formatos Padronizados para a Definição de Requisitos

A utilização de formatos padronizados para redigir diferentes tipos de requisitos ajuda a garantir que cada especificação é **clara, consistente, completa e testável**.

3.1 Formato para Requisitos de Utilizador

Os requisitos de utilizador descrevem os **objetivos ou tarefas que os utilizadores devem ser capazes de realizar com o sistema**. Um formato eficaz inclui os seguintes componentes:

- **Sujeito:** O tipo de utilizador que beneficia do requisito.
- **Resultado pretendido:** Um predicado que descreve a funcionalidade (verbo) e os seus complementos (objetos, conceitos).
- **Mecanismo de teste:** Um critério mensurável que permite verificar se o requisito foi cumprido.

Exemplo: (users) The hotel receptionist (functionality) should visualise (object/concept) the room number of a guest, (test) 2 s after making the request.

Neste exemplo, o sujeito é "The hotel receptionist", o resultado pretendido é "visualise the room number of a guest" e o mecanismo de teste é a verificação de que a ação ocorre em menos de "2 s".

3.2 Formato para Histórias de Utilizador (User Stories)

Comuns em metodologias ágeis, as user stories são descrições curtas de uma funcionalidade na perspetiva de quem a necessita. Colocam o foco no valor para o utilizador. A sua estrutura clássica é:

As a , I want to for .

Exemplo: As a hotel receptionist, I want to visualise the room number of a guest for calling him if someone wants to contact him.

Este formato coloca o utilizador no centro ("As a hotel receptionist"), define claramente o objetivo ("I want to visualise the room number of a guest") e justifica a sua necessidade ("for calling him..."), o que facilita a priorização e a compreensão do contexto.

3.3 Formato para Requisitos de Sistema

Os requisitos de sistema são especificações mais detalhadas sobre o comportamento do próprio sistema. O formato padronizado inclui:

- **Sujeito:** O sistema ou uma entidade de design específica.

- **Resultado pretendido:** Um predicado que descreve a funcionalidade a ser implementada (verbo) e outros elementos que completam a descrição.

Exemplo: (system/entity) The signal of the battery (functionality) must turn on, (description) when the charge is lower than 20mAh.

Aqui, o sujeito é "The signal of the battery" e o resultado pretendido é a sua ativação sob uma condição específica.

3.4 Formato para Requisitos Não-Funcionais

Os requisitos não-funcionais definem as **qualidades do sistema** (ex: desempenho, segurança, usabilidade) em vez das suas funcionalidades. A sua estrutura é geralmente composta por:

- O sistema ou entidade a que a qualidade se aplica.
- A qualidade a ser alcançada, expressa através de um verbo e uma descrição.

Exemplos:

- The product shall be easy to use for illiterate persons.
- The product must continue to function at 30 metres under water.
- The product must be prepared to be translated to any language.
- The source code of the product programs should contain comments.

4. Armadilhas Comuns a Evitar na Escrita de Requisitos

4.1 Terminologia Vaga

Muitas palavras usadas informalmente para descrever características desejáveis são demasiado vagas para constituírem requisitos testáveis.

- Exemplos de termos a evitar: easy to use/learn, versatile, flexible, intuitive, modern, improved, efficient, approximately, more possible, minimal impact.

Considere o seguinte requisito vago: The e-mail application must have an intuitive user interface.

4.2 Ilusões e Fantasias (Requisitos Impossíveis)

Em engenharia, não existem componentes perfeitos. Os requisitos devem ser realistas e exequíveis. É crucial evitar o chamado wishful thinking, ou seja, a especificação de objetivos impossíveis de alcançar.

- Termos a evitar: 100% reliable, totally safe, never fails, satisfies all users, handle all unforeseen situations.

Exemplos de requisitos irrealistas:

- The printer shall be always operational.
- The software application shall handle all unexpected errors.
- The biometric authentication system shall be 100% reliable.

4.3 Múltiplos Requisitos numa Só Frase

Incluir vários requisitos numa única frase, especialmente usando conjunções coordenativas (FANBOYS: for, and, nor, but, or, yet, so), é uma fonte comum de ambiguidade.

Por exemplo, a frase The guest shall pay the bill with money or credit card deve ser dividida em requisitos mais simples e claros.

Um caso mais complexo de ambiguidade surge em frases como: The system shall produce a beep or visual signal to be sent to the director or secretary.

Esta frase pode ser interpretada de múltiplas formas: um sinal sonoro para o diretor e um visual para a secretária? Ou ambos os tipos de sinal para ambos os destinatários? A solução é dividir a frase em requisitos atómicos e inequívocos.

4.4 Inclusão Prematura de Detalhes de Design

Os requisitos devem focar-se no "quê" (a funcionalidade necessária), e não no "como" (a forma de implementação). A inclusão de decisões de design no documento de requisitos é prematura, limita a exploração de soluções técnicas mais adequadas e pode levar a uma arquitetura subótima.

Exemplos de requisitos com detalhes de design desnecessários:

- The clock shall present the current civil local time, by receiving code time signals from a radio station. (Especifica a tecnologia de rádio, quando o objetivo é apenas mostrar a hora correta).
- The guest shall complain about the hotel services through a form available in the web. (Especifica uma "web form", quando o requisito poderia ser simplesmente permitir que o hóspede registre uma reclamação).

4.5 Misturar Requisitos com Planos de Projeto

Um documento de requisitos e um plano de projeto são artefactos distintos com propósitos diferentes. Informações como datas, fases do projeto, alocação de recursos e atividades de gestão não devem ser incluídas no documento de requisitos. Estes detalhes pertencem ao plano de projeto e devem ser mantidos num documento separado.

A ambiguidade, um problema recorrente nas armadilhas mencionadas, merece uma análise mais aprofundada devido ao seu impacto prejudicial nos textos técnicos.

5. Análise Aprofundada da Ambiguidade na Linguagem

5.1 Causas de Ambiguidade Lexical: Homonímia e Polissemia

A ambiguidade pode surgir ao nível das palavras individuais. As suas principais causas são:

- **Homonímia:** Ocorre quando palavras com a mesma forma (escrita ou sonora) têm significados distintos e sem relação etimológica.
- **Polissemia:** Ocorre quando uma única palavra possui vários significados que estão, de alguma forma, semanticamente relacionados.

Na prática, a distinção entre estes dois fenómenos nem sempre é fácil, mas o resultado é o mesmo: o potencial para uma interpretação incorreta.

5.2 Causas de Ambiguidade Estrutural

A ambiguidade também pode derivar da forma como as palavras são organizadas numa frase, independentemente do significado individual de cada palavra.

Pronomes Possessivos

O uso de pronomes possessivos na terceira pessoa (ex: "seu", "sua", his, her, their) pode criar confusão quando existem múltiplos sujeitos na frase.

Considere a frase: The director calls the doctor about his problems.

Esta frase é ambígua porque "his problems" pode referir-se tanto aos problemas do diretor como aos problemas do médico. Para eliminar a ambiguidade, a frase teria de ser reescrita para clarificar a quem pertencem os problemas.

Conjunções e Enumerações

As conjunções, especialmente "e" (and) e "ou" (or), são fontes frequentes de ambiguidade estrutural, tal como as enumerações.

- O "ou" pode ser interpretado de forma inclusiva (A ou B, ou ambos) ou exclusiva (A ou B, mas não ambos). O contexto técnico exige que esta distinção seja clara.
- A expressão "e/ou" (and/or) deve ser proibida em documentação técnica, pois é inerentemente ambígua. É preferível usar apenas "ou" e clarificar o seu significado se necessário.