

## Escrever Bons Requisitos

**A principal razão porque as pessoas escrevem maus requisitos tem a ver com o facto de não terem recebido formação ou não terem experiência nessa área. Este artigo procura mostrar o que é um bom requisito e abrange alguns dos problemas mais comuns, assim como a forma de os evitar. Os exemplos apresentados facilitam a compreensão.**

Quando deparamos com problemas na escrita de bons requisitos, o melhor é recorrer a ajuda. Muitos dos licenciados em cursos relacionados com a informática e a computação nunca ouviram falar de gestão de requisitos nas universidades. Aqueles que ouviram falar, provavelmente não foram além de uma breve introdução ao assunto. A informação fornecida neste artigo é utilizada na formação de pessoas para escreverem bons requisitos.



Um dos aspectos importantes da engenharia de sistemas consiste na conversão das “necessidades” dos utilizadores em requisitos de sistema claros, concisos e verificáveis. Este artigo concentra-se sobretudo nos requisitos a nível do sistema, mas também pode ser aplicado a níveis mais baixos.

### **Bons requisitos**

Um bom requisito especifica algo que é necessário, verificável e atingível. Se for verificável, atingível, estiver escrito de forma eloquente, mas não for necessário, não será um bom requisito. Para ser verificável, o requisito terá que especificar algo que possa ser verificado através de um exame, análise, teste ou demonstração. Se a especificação for subjectiva, ou incluir palavras subjectivas – por exemplo, “fácil” – não será verificável. Se o requisito não for atingível, não servirá de muito escrevê-lo. Um bom requisito tem que ser especificado de forma clara.

**Necessário.** Se existirem dúvidas quanto à necessidade de um requisito, deverá perguntar-se qual será a pior coisa que poderá acontecer se esse requisito não for incluído. Se não encontrarmos qualquer problema, então é porque provavelmente não precisamos do requisito.

**Verificável.** À medida que se escreve um requisito, é necessário determinar a forma como iremos verificá-lo, assim como o critério de aceitação. Este aspecto é importante porque ajudará a assegurar que o requisito é verificável.

**Atingível.** Para ser atingível, um requisito tem que ser tecnicamente factível e enquadrar-se no orçamento, calendarização e outros constrangimentos do projecto. Se não tivermos a certeza quanto à execução técnica do requisito, deveremos estudar o assunto e proceder à investigação necessária para determinar se é factível ou não. Se, mesmo assim, se mantiver a dúvida, deveremos especificar aquilo que queremos como objectivo (e não como requisito). Mesmo que um requisito seja tecnicamente executável, poderá não ser atingível devido a limitações orçamentais, de

tempo (calendarização), ou outras. Não vale de nada escrever um requisito para algo que não podemos pagar ou concretizar em tempo útil. Há que ser razoável.

**Claro.** Cada requisito deverá expressar uma única ideia, além de ser conciso e simples. É importante que os requisitos não sejam entendidos de forma errada ou incorrecta – não podem ser ambíguos. Na maior parte dos casos, bastam frases simples para especificar um bom requisito.

### **Problemas comuns**

Os problemas mais comuns em que se incorre quando se escrevem requisitos são os seguintes:

- Tomar como base pressupostos incorrectos
- Escrever implementações (como) em vez de requisitos (o quê)
- Descrever operações em vez de escrever requisitos
- Utilizar termos incorrectos
- Utilizar uma estrutura de frase incorrecta ou expressar-se mal em termos gramaticais
- Deixar requisitos de fora
- Cair no excesso de especificação

### **Pressupostos incorrectos**

Este problema ocorre quando os autores dos requisitos não têm acesso a informação suficiente ou quando não existe informação nenhuma. Podemos resolver o primeiro problema (insuficiência de informação) através da documentação da informação crítica para o projecto, incluindo as necessidades, objectivos, limitações, missões, conceitos operacionais, orçamento, calendarização e gestão/organização.

Seguidamente, esta informação tem que ser disponibilizada a todos os intervenientes. No caso de se ter automatizado o processo, os documentos podem ser disponibilizados em linha. Também se pode filtrar a informação desses documentos para que as pessoas possam obter cópias que contenham apenas aquilo que realmente precisam.

Para resolver o segundo problema (inexistência de informação), os autores dos requisitos têm que documentar todos os pressupostos associados a cada requisito. Desta forma, quando o requisito for analisado, os pressupostos também o poderão ser. Os problemas serão assim identificados rapidamente. Também é de grande utilidade documentar os pressupostos nos casos em que foi fornecida a informação correcta aos autores. Como não podemos garantir que todos os autores leram toda a informação ou a interpretaram correctamente, se documentarem os seus pressupostos poderemos evitar surpresas mais tarde.

### **Implementações versus requisitos**

Começamos com a criação de uma especificação para o desenvolvimento de uma ferramenta de gestão de requisitos. O primeiro requisito é “o fornecimento de uma base de dados”. É frequente a especificação ter a ver com a implementação e não com a necessidade. Este erro é bastante comum nas especificações de requisitos.

As especificações devem referir o que é necessário e não como vai ser disponibilizado. Este é um erro comum cometido por quem escreve requisitos. Na realidade, não pretendem especificar a implementação. Simplesmente não sabem como formular correctamente a necessidade. Para evitar a especificação de implementações, devemos perguntar-nos porque precisamos do requisito.

No exemplo que acabámos de referir, podemos ver claramente que ao perguntarem porquê, os autores poderão definir todas as necessidades a que o sistema deverá responder, especificando depois os requisitos reais. Por exemplo:

- Disponibilizar capacidade para o acompanhamento dos requisitos
- Disponibilizar capacidade para adicionar atributos aos requisitos
- Disponibilizar a possibilidade de ordenar os requisitos

Estes requisitos especificam o que é necessário e não como atingi-lo. Cada um dos requisitos referidos atrás poderá resultar num tipo de base de dados de sistema, mas o requisito para a base de dados não foi necessário. Existem dois grandes perigos em especificar implementações.

O risco que é referido com maior frequência é o de forçar um *design* quando não é isso que se pretende. Se todas as necessidades puderem ser satisfeitas sem uma base de dados, para quê especificar a necessidade de uma base de dados? Se nem todas as necessidades puderem ser satisfeitas sem uma base de dados, haverá outras formas de chegar à solução, independentemente de existir ou não um requisito para uma base de dados.

O segundo perigo é mais subtil e potencialmente muito mais nefasto. Ao especificarem a implementação, os autores poderão convencer-se que estão a considerar todos os requisitos. No entanto, poderão faltar requisitos muito importantes. Por exemplo, um fornecedor poderá disponibilizar aquilo que lhe foi pedido e, mesmo assim, não fornecer aquilo que se pretende. No nosso exemplo, o fornecimento de uma base de dados não seria suficiente para alguém que precisa de uma ferramenta de gestão de requisitos. São as capacidades da ferramenta que precisam de ser especificadas como requisitos.

Em cada nível do desenvolvimento dos requisitos ocorrerá o problema das necessidades versus implementação. A nível do sistema, os requisitos têm que especificar o que é necessário. O modelador do sistema irá então determinar como é que isso pode ser efectuado e identificar posteriormente o que é necessário a nível dos subsistemas. O modelador do subsistema irá determinar como é que uma dada necessidade pode ser satisfeita e terá que definir o que é necessário a nível do componente.

Para nos assegurarmos que não especificamos a implementação, devemos perguntar-nos porque precisamos do requisito. Se depois disto não formos remetidos para a especificação de uma necessidade real, então é porque estamos provavelmente a especificar uma necessidade e não uma implementação.

**A armadilha da implementação.** Se estivermos a efectuar estudos de modelação a um baixo nível, poderemos começar a documentar esses resultados como requisitos de alto nível. Esta é a armadilha da implementação. Estaremos a definir a implementação em vez dos requisitos. Um exemplo do que acabamos de dizer ocorreu com os requisitos do Assured Crew Return Vehicle (ACRV) da NASA.

Um indivíduo submeteu um requisito do tipo: *O ACRV System deve entrar (na atmosfera) quando o estado do mar estiver em condições TBD (To Be Determined) – ou seja, a serem determinadas.* Desta forma, o ACRV não tinha qualquer requisito para aterrar na água – o que era uma opção de desenho. Trabalhou-se então com essa opção de desenho e com base na experiência da Apollo. De acordo com esta experiência passada, a recuperação da tripulação da nave só era possível com determinados estados do mar.

Quando lhe foi perguntado porque era necessário esse requisito, o mesmo indivíduo afirmou que a tripulação não poderia ser deixada no módulo durante muito tempo, pelo que a aterragem precisava de ser num local e numa altura em que o estado do mar permitisse o resgate da tripulação. Na realidade, tinha um requisito válido – mas não aquele que tinha escrito.

Independentemente do ACRV aterrar na água ou em terra, era essencial o resgate da tripulação num período de tempo limitado. Como tal, o verdadeiro requisito seria o seguinte: *o ACRV System deve permitir a remoção da tripulação dentro de um espaço de tempo de aterragem TBD.* A questão “porquê” permite resolver a maior parte dos erros de implementação de requisitos. Deve-se perguntar sempre o porquê da necessidade de um requisito para assegurar que não caímos nesta armadilha dos requisitos de mais baixo nível.

### **Operações versus requisitos**

Este problema é similar ao da implementação. Como exemplo, podemos recorrer ao projecto Simplified Aid for EVA Rescue (SAFER) da NASA. O autor pretendia escrever um requisito de ambiente na especificação SAFER Flight Test Article (FTA), mas não fez mais do que escrever uma descrição de operações – e não um requisito sobre o ambiente. *O SAFER FTA deve ser colocado no Orbiter Airlock Stowage Bag para lançamento, aterragem e navegação em órbita.*

O requisito real é o seguinte: *o SAFER FTA deve ser concebido para o ambiente de carga do Airlock Storage Bag para lançamento, entrada, aterragem e navegação em órbita, tal como definido em TBD.*

Mais uma vez, o requisito seguinte descreve as operações e é confuso. *O SAFER FTA deve ser operado por um membro da tripulação EVA (Extravehicular Activity) vestido com fatos EMU (Extravehicular Mobility Unit) de tamanhos pequeno a extra grande (small a extra large), sem limitar a mobilidade do fato.*

O texto foi rescrito e resultou num requisito e num objectivo de desenho. O objectivo de *design* foi necessário porque não pode ser escrito nenhum requisito quantificável relativamente à mobilidade do fato.

*O SAFER FTA deve ser concebido para ser utilizado com fatos EMU de tamanhos pequeno a extra grande.*

*O SAFER FTA não deve limitar a mobilidade do membro da tripulação EVA.*

O perigo em especificar operações em vez de requisitos pode ser dividido em duas partes. Por um lado, a intenção pode ser mal compreendida. Por outro, a determinação de como de vai efectuar a verificação pode ser um problema.

### **Utilizar termos incorrectos**

Numa especificação existem termos a evitar e outros que devem ser utilizados de uma forma muito específica. Quem escreve requisitos precisa de entender a utilização das palavras “deve ser” (shall), “será” (will) e “deveria ser” (should). (A utilização do verbo *ser* serve apenas para facilitar a compreensão. Poderá utilizar-se outro verbo).

- Os requisitos utilizam o “deve ser”
- As especificações de factos utilizam o “será”
- Os objectivos utilizam o “deveria ser”.

Esta é a utilização standard destes termos nas agências governamentais e na indústria dos Estados Unidos. Se utilizarmos derivados, estaremos a lançar a confusão. Todas as especificações “deve ser” (requisitos) têm que ser verificáveis. Caso contrário, a conformidade não poderá ser demonstrada. Termos como “são”, “é”, “foi” e “tem de ser” não se aplicam na especificação de requisitos. Podem ser utilizados numa secção descritiva ou na introdução de uma secção de requisitos da especificação.

Também existem vários termos a evitar quando se escrevem requisitos, dado que confundem a questão em causa e podem custar-nos dinheiro. Entre esses termos destacamos “suporte”, “mas não limitado a”, “etc.”, “e/ou”. A palavra “suporte” é frequentemente utilizada de forma incorrecta nos requisitos. Será adequada a sua utilização se quisermos uma estrutura para suportar 100 quilos de peso. No entanto, já será incorrecta a utilização da palavra se quisermos dizer que o sistema irá suportar determinadas actividades.

**Errado.** *O sistema deve suportar o coordenador de formação na definição de cenários de formação.*

**Certo.** *O sistema deve fornecer ecrãs de entrada para a definição de cenários de formação. O sistema deve fornecer processos automatizados de cenários de formação.*

As expressões “mas não limitado a” e “etc.” costumam ser utilizadas porque quem

está a escrever os requisitos acha que poderá ser necessário algo mais além daquilo que é referido. A utilização destes termos não responderá àquilo que o autor pretende e poderá virar-se mesmo contra ele. Estes termos são utilizados para abarcar o desconhecido e o fornecedor não aumentará o custo na proposta pelo facto de termos utilizado esses termos.

A única forma de se conseguir que seja adicionado trabalho é colocar uma tarefa de análise na declaração de trabalho (ou Statement of Work – SOW) para determinar se é necessário adicionar mais itens à lista. Na declaração de trabalho podemos controlar o esforço necessário para que o fornecedor responda a esses aspectos desconhecidos. Se forem adicionados mais aspectos, poderá ser necessário aumentar a abrangência do contrato para cobrir essas adições.

No entanto, se tivermos estes termos na nossa especificação de requisitos, o fornecedor poderá utilizá-los como uma desculpa para realizar trabalho desnecessário, que acabará por custar dinheiro ao cliente. Não se sai a ganhar com a utilização desses termos nas especificações.

A expressão “e/ou” também não é adequada numa especificação. Se a utilizarmos e o fornecedor optar pelo “ou”, estará a cumprir os termos do contrato. Na realidade, ou queremos o aspecto um e o aspecto dois, ou ficaremos satisfeitos com o aspecto um ou com o aspecto dois. Se utilizarmos o termo “ou”, o fornecedor cumprirá os termos do contrato se responder apenas a um dos dois aspectos.

#### **Estrutura/gramática dos requisitos**

Os requisitos devem ser fáceis de ler e de compreender. Os requisitos, na especificação de um sistema, são tanto para o sistema, como para o seu nível seguinte – ou seja, subsistema. Cada requisito pode ser escrito normalmente de acordo com os formatos que se seguem:

- *O sistema deve fornecer...*
- *O sistema deve ser capaz de...*
- *O sistema deve considerar...*
- *O subsistema 1 deve fornecer...*
- *O subsistema 2 deve ter interface com...*

**Nota.** O nome do sistema e o de cada subsistema é mencionado. Quando se tratar de nomes complexos, devem utilizar-se acrónimos. Caso contrário, o documento terá uma série de páginas desnecessárias pelo facto de termos escrito um nome longo várias vezes. Cada um destes princípios é seguido por aquilo que o sistema ou o subsistema deve fazer. Cada um deveria ser seguido geralmente por um único predicado e não por uma lista.

Existem, no entanto, situações em que é apropriado utilizar uma lista. No entanto, as listas são demasiado utilizadas. Uma vez que cada item da lista tem que ser verificado – a não ser que todos os itens venham a ser verificados pelo mesmo método e ao mesmo tempo – geralmente não é apropriado colocar itens numa lista. A especificação dos requisitos também não deve ser tornada complexa por explicações de operações, *design*, ou outra informação relacionada. Esta informação que não é requisito deve ser fornecida numa introdução a um conjunto de requisitos ou na justificação.

Se respeitarmos rigorosamente este formato podemos conseguir duas coisas. Em primeiro lugar, evitamos a Armadilha do Assunto. Em segundo lugar, evitaremos problemas gramaticais que afectam os requisitos quando os seus autores enveredam por uma escrita criativa.

**Armadilha do Assunto.** Uma especificação de sistema contém requisitos para o sistema. Por exemplo, podemos querer que o sistema disponibilize controlo. Têm então que ser escritos requisitos como se segue:

- *O subsistema de controlo e assistência deve fornecer controlo e acordo com*

*seis graus de liberdade.*

- *O subsistema de controlo e assistência deve controlar a posição de acordo com  $2\pm 0.2$  graus.*
- *O subsistema de controlo e assistência deve controlar os valores de acordo com  $0.5\pm 0.05$  graus/segundo.*

A armadilha do assunto é criada porque o autor definiu um subsistema de controlo e assistência. O controlo da posição e dos valores é um problema do sistema. Requer não só um subsistema de controlo e assistência, mas também um subsistema de propulsão para alcançar essas posições e valores. A determinação de quais os subsistemas que serão necessários para cumprir os requisitos faz parte do processo de desenho. Os requisitos deverão ser escritos a partir da perspectiva do sistema, como se segue:

- *O sistema deve fornecer controlo de acordo com seis graus de liberdade.*
- *O sistema deve controlar a posição de acordo com  $2\pm 0.2$  graus.*
- *O sistema deve controlar os valores de acordo com  $0.5\pm 0.05$  graus/segundo.*

O autor dos requisitos originais não estava a tentar definir o nível mais baixo de ruptura. Vem provavelmente de um *background* de controlo e vê o sistema a partir dessa perspectiva, o que faz com que escreva os requisitos dessa forma. A repercussão descendente dos requisitos a nível de todos os segmentos, elementos e subsistemas afectados será mal feita se os requisitos não estiverem escritos de forma correcta.

**Problemas gramaticais.** Se utilizarmos mal as regras gramaticais, arriscamo-nos a que os leitores interpretem mal aquilo que está escrito. Se utilizarmos a estrutura de requisitos sugerida atrás, estaremos a eliminar os problemas gramaticais que ocorrem quando os autores tentam escrever frases complexas e utilizam demasiadas frases. Outra solução é escrever os requisitos de forma sucinta. Quando o conteúdo é acordado, uma pessoa com bom domínio da escrita pode converter a informação numa frase para a especificação.

Os autores também costumam tentar colocar tudo aquilo que sabem numa única frase. Isto resulta em frases longas e complexas que contêm provavelmente mais do que um requisito. Uma pessoa com um bom domínio da escrita poderá minorar este problema. O requisito que se segue contém, na realidade, vários requisitos.

*O Sistema ACRV deve fornecer instalações médicas especiais de apoio para um membro da tripulação doente ou ferido, consistindo de suporte médico e equipamento de monitorização e capacidade para limitar as acelerações de impacto para esse membro da tripulação, de forma a não serem superiores a... para um impulso total que não exceda...*

Este requisito precisa de ser partido em pelo menos quatro requisitos e poderá utilizar-se uma introdução como se segue:

*O ACRV será utilizado como ambulância para um membro da tripulação doente ou ferido. Só um membro da tripulação será acomodado de cada vez. Os pontos que se seguem definem os requisitos específicos para essa capacidade.*

- *...fornecer instalações médicas de suporte para um membro da tripulação*
- *...fornecer equipamento de monitorização para um membro da tripulação*
- *...limitar as acelerações de impacto para o membro da tripulação doente ou ferido de forma a não serem superiores a...*
- *...limitar o impulso total para o membro da tripulação doente ou ferido a...*

#### **Não verificável**

Todos os requisitos têm que ser verificados. Como tal, é importante ter em conta essa verificação quando se estão a escrever os requisitos. Os requisitos podem não ser verificáveis por várias razões. No entanto a mais comum costuma ser a utilização de termos ambíguos.

**Termos ambíguos.** Uma das principais causas dos requisitos não serem verificáveis tem a ver com a utilização de termos ambíguos. Essa ambiguidade acontece porque se trata de termos subjectivos – podem significar coisas diferentes, conforme a pessoa que os lê. Isto pode ser evitado fornecendo às pessoas um conjunto de palavras que não devem utilizar. Apresentamos a seguir uma lista de palavras ambíguas que costumamos encontrar.

- Minimizar
- Maximizar
- Rápido
- Amigável
- Fácil
- Suficiente
- Adequado
- Breve

As palavras maximizar e minimizar não podem ser verificadas. Nunca poderemos dizer se as conseguimos concretizar na prática. As palavras mínimo e máximo podem ser utilizadas se o contexto em que se inserem puder ser verificado. E o que é amigável com o utilizador ou rápido? Isto poderá significar uma coisa para os utilizadores e para o cliente e outra inteiramente diferente para um modelador.

Quando começamos a escrever requisitos, pode-se proceder com base naquilo que nós mesmos estamos a pensar. No entanto, é necessário escrever os requisitos de forma que possam ser verificados. Se tivermos que utilizar termos ambíguos nas primeiras versões dos documentos (rascunho), devem-se colocar asteriscos (ou outra forma de destaque) no início e no fim da palavra ou expressão para nos lembrarmos mais tarde que temos de colocar algo mais concreto no requisito, antes de o instituímos como base de trabalho.

Poderão ocorrer situações em que não conseguimos definir exactamente ao nosso nível aquilo que é necessário. Quando isto acontecer, deveremos escrever, por exemplo, um objectivo de desenho e não um requisito. Podemos proceder desta forma indicando claramente que se trata de um objectivo e não de um requisito. A utilização da palavra deveria (em vez de deve) já denotará que se trata de um objectivo de desenho.

#### **Esquecer Requisitos**

O esquecimento de alguns aspectos pode ser evitado utilizando um esquema standard para a nossa especificação – tal como os que são apresentados no Mil-Std-490 ou no IEEE P1233 – e expandindo o esquema para o adaptar às especificidades do nosso projecto. Muitos requisitos são esquecidos porque a equipa que está a escrever os requisitos coloca o enfoque apenas numa parte do sistema – por exemplo, nos requisitos funcionais e de desempenho – esquecendo-se de outros requisitos importantes, mas menos óbvios. A lista de requisitos que se segue deve ser sempre considerada.

- Funcional...Fiabilidade
- Desempenho...Capacidade de Manutenção
- Interface...Operacionalidade
- Ambiente...Segurança
- Capacidade...Rigidez
- Transporte...Segurança
- Exploração...Privacidade
- Formação...Restrições de Desenho
- Pessoal

Será necessário definirmos um esquema detalhado para a nossa especificação dos requisitos funcionais e de desempenho e talvez de outras áreas. Também podemos ter um conjunto de requisitos que precisamos de incluir como referência. Incluem-se

neste caso aqueles standards que definem a qualidade em diferentes disciplinas (materiais e processos) ou para diferentes projectos.

A análise detalhada dos requisitos é necessária para assegurar que todos eles são cobertos. Existem várias abordagens para efectuar a análise dos requisitos, bem como várias ferramentas para realizar esse trabalho. No entanto, a análise detalhada dos requisitos não se enquadra no âmbito deste artigo.

### **Excesso de especificação**

O departamento de defesa americano declarou que o excesso de especificação é a principal causa para o não cumprimento dos limites orçamentais dos seus projectos. Este problema tem a ver normalmente com a especificação de algo que não é necessário ou com a especificação de requisitos demasiado rígidos.

**Requisitos desnecessários.** Os requisitos desnecessários entram numa especificação de várias formas. A única forma de contrariar este problema consiste em implementar um controlo e análise cuidadosos em termos de gestão. Quando é pedido às pessoas para escreverem requisitos, elas escreverão tudo aquilo de que se lembrarem. Desta forma, se não analisarmos cuidadosamente cada requisito e a razão porque é necessário antes da sua instituição como base de trabalho, o resultado será a necessidade de lidar com vários requisitos desnecessários.

**Exemplo:** A Space Station Training Facility (SSTF) tinha um requisito para um campo estelar de alta-fidelidade. O autor sabia que tinha sido desenvolvido um novo campo estelar de alta-fidelidade para o Shuttle Mission Simulator (SMS) e partiu do pressuposto que podiam utilizar a mesma coisa na SSTF. A tripulação precisa de uma base para ver para fora da Estação Espacial, mas não é necessário um campo estelar de alta-fidelidade, dado que não utilizam as estrelas para navegação.

O requisito precisa de ser escrito para uma base visual destinada à orientação da tripulação. O processo de desenho determinará se a utilização do campo estelar SMS é uma solução plausível em termos de custo, ou se é adequado algo mais simples e mais barato.

**Exemplo:** Foram eliminados vários requisitos da SSTF quando os seus autores foram interrogados sobre a necessidade dos mesmos. A justificação da necessidade desses requisitos era que “seria interessante ter”. No entanto, a maior parte dos projectos não tem orçamento para coisas que seria interessante ter.

Os requisitos desnecessários também podem surgir depois de se constituir a base de trabalho. Para tal, basta que descoremos o nosso processo de análise e de controlo. No caso do ACRV foram adicionados vários requisitos desnecessários depois de instituída a base de trabalho inicial. Um caso desses aconteceu devido a um erro no documento que constituía a base de trabalho.

**Exemplo:** O documento base tinha dois requisitos:

- *O ACRV System deve ser capaz de funcionar durante um período de vida operacional planeado de trinta (30) anos.*
- *O Flight Segment deve disponibilizar um período de vida operacional de 30 anos para os elementos de voo.*

O segundo requisito para o Flight Segment não era necessário, dado que o requisito de Sistema era adequado. Deveria, portanto, ter-se eliminado o requisito Flight Segment. Em vez disso, foram adicionados mais dois requisitos para requerer um tempo de vida operacional de 30 anos para os outros dois segmentos. Foi adicionado pelo menos outro requisito que era um duplicado de um requisito existente. As palavras dos dois requisitos diferiam ligeiramente, mas a ideia era a mesma. É necessário ter especial atenção ao detalhe para evitar este tipo de problema.

**Excesso de rigidez.** A maior parte dos requisitos que são inflexíveis são-no acidentalmente e não de forma intencional. Uma causa comum dessa rigidez é quando um autor escreve um número e não considera a tolerância que é permitida.

Como tal, não devemos especificar que algo tem de ter uma dada dimensão (por exemplo, 100 centímetros) quando na realidade pode ter 100+/-10 cm. Também não precisamos de pedir que algo disponibilize uma carga de exactamente 200 se for aceitável um valor maior ou igual a 200.

Algumas das principais histórias surrealistas da indústria aeroespacial estão relacionadas com requisitos demasiado rígidos. Um fornecedor foi severamente criticado por cobrar 25000 dólares por cada cafeteira destinada a aviões construídos para o governo dos Estados Unidos. No entanto, os requisitos para a cafeteira eram tão rígidos que, mesmo que o avião se despenhasse, o café não poderia entornar-se.

Facilmente se compreende que o processo de desenvolvimento e verificação da cafeteira tenha sido bastante caro para cumprir o requisito. Cada duplicado tinha que ser construído de acordo com um *design* rígido. A solução para este tipo de problema consiste na discussão da tolerância permitida para um dado valor e depois escrever o requisito tendo em conta essa tolerância. O custo de cada requisito também tem que ser tido em conta.

*Baseado no artigo "Writing Good Requirements", de Ivy Hooks, presidente e CEO da Compliance Automation. Dos seus cerca de 30 anos de experiência, 20 passou-os na NASA.*

Tel: (+351) 239 497 230 / 2 • Fax: (+351) 239 497 231  
E-mail: [tim@engenharia-software.com](mailto:tim@engenharia-software.com) • Internet: [www.engenharia-software.com](http://www.engenharia-software.com)