We have seen that, ideally, a specification should confine itself to what is needed. We now present a list of desirable qualities for a specification. A good specification should exhibit the following characteristics:

■ implementation free – what is needed, not how this is achieved  
■ complete – there is nothing missing  
■ consistent – no individual requirement contradicts any other  
■ unambiguous – each requirement has a single interpretation  
■ concise – each requirement is stated once only, without duplication  
■ minimal – there are no unnecessary ingredients  
■ understandable – by both the clients and the developers  
■ achievable – the requirement is technically feasible  
■ testable – it can be demonstrated that the requirements have been met.

This list of desirable features can be used as a checklist when a specification is drawn up. Additionally it can be used as a checklist to examine and improve an existing specification.

Let us now look at the requirements specification for a simple piece of software:

*Write a Java program to provide a personal telephone directory. It should implement functions to look up a number and to enter a new telephone number. The program should provide a friendly user interface.*

and apply the checklist above.

On the issue of implementation, the specification says that the program is to be written in Java, which is definitely to do with the “how” of implementation. Second, the specification gives no detail about the detail of the two functions; it is incomplete. Often a requirement is simply unclear or susceptible to alternative interpretations, and this, of course, may well be due to the use of natural language in the specification. Vagueness is a common problem. Thus the requirement to provide a user-friendly interface is hopelessly vague, thereby making the specification incomplete and untestable.

Some words are vague and therefore should be avoided within a specification. Some typical examples are the words “flexible”, “fault tolerant”, “fast”, “adequate”, “user friendly”.

Sometimes requirements contradict each other, as in these two:

*the data will be stored on magnetic tape*  
*the system will respond in less than 1 second.*

because magnetic tape cannot provide a one-second response time.