

# Specifying a Project

Why requirement specifications are important  
- a brief Ignys guide.

## Why write a Requirements Document?

When starting a project the end goal is usually somewhat undefined.

A requirements document can help to establish the functionality of the end result. Providing both a discussion document for the customer and developer, and a well defined way of assessing "are we finished?". A good requirements document does not restrict the chosen solution and gives an unambiguous way to judge if the delivery is correct.

It does not usually define how a solution should be engineered - as the implementation could often be one of several ways.

Occasionally a requirements document will specify parts of the implementation (e.g. if it will run on a XYZ processor). These types of requirements, whilst potentially required for commercial reasons should be avoided if possible as they can place unforeseen constraints on the end solution.

## Who is a requirements document for?

The aim of a requirements document is to explicitly map out all the things that a project will and will not deliver.

It forms part of the contract between the customer side and the engineering side of a development. It should be led by the project owner, (typically a product manager) with input sought from all interested parties – such as End users, Product managers, Engineers/Developers, Commercial and Sales staff (etc.)

Requirements documents are often live documents – i.e. as a project progresses its' scope may change, or functionality that was unclear becomes more well defined.

## What a good requirement looks like

- **Unique:** It defines a single aspect of a product.
- **Clear & Unambiguous:** It clearly states the intended outcome, without unresolved ambiguities.
- **Testable:** A requirement that is testable can be proven to have been met.
- **Complete:** A set of requirements covers the whole functionality of the outcome. If there is no requirement to do something there is no recourse if the end outcome doesn't achieve this.
- **Feasible:** It does not matter if a requirement is all of the above if it is not actually possible to create the functionality described. Whilst new developments will inevitably contain requirements that describe things which have not yet been done for a successful project at least some feasibility work should have been carried out to minimise risk.

## Are We There Yet?

Requirements and Test specifications allow a judgement on the completion of a project. Each requirement should be testable, and the outcome of the test should then prove that the final product does what was agreed.

Test specifications are often overlooked in the creation of a project document set, but with a correct and complete Requirements specification a Test specification simply becomes a list of the requirements, an appropriate test for each requirement, and a record of the test result.

### How to state a requirement

There are many ways to state a requirement. The MoSCoW mnemonic technique is frequently used.

**Must Have:** A feature that the end product must be able to demonstrate. Without the must have features the product is not viable.

**Should Have:** A feature that the end product should have it if is possible without missing out a must have, but if it requires significant effort it may be better to leave it out. The product will still function as intended, but will not be as compelling.

**Could Have:** A feature that the end product might have it if is possible without significant effort, or if all the must and should are done and there is time / budget left.

**Won't Have:** A feature that the end product will not have / will not do. Sometimes it is simpler to state what must be missing than it is to include all the things which must be included.

The Must and Wont requirements define the scope of the minimum viable product. Without these the product will not do what it is intended to do

Should and Could are improvements to the product in stages of importance. Missing them will mean the product is not as nice to use, but it will still function as intended.

Splitting requirements in this manner informs the effort that should be put into meeting each requirement, assuming that there are more requirements than there is time / resource available to do them.

Some projects do not have 'should' or 'could' requirements.

### Example of how to state a requirement

The following requirements are for a generic mug (for a hot beverage like tea or coffee).

- Mug must contain hot liquids (100° C)
- Mug must hold 250ml
- Mug must provide access to hot beverage
- Mug must not cost more than £7.50
- Mug should have 350ml capacity
- Mug should not cost more than £3
- Mug should be dishwasher safe
- Mug could be cylindrical
- Mug could be coloured
- Mug could be drop proof (to MIL-STD-810G 516.6)
- Mug could be spill proof
- Mug must not have sharp edges
- Mug must not tip over when placed on flat, level surface



Both the size and price requirements are defined as a must and a more preferential 'should'. Whilst splitting it in this way increases the number of individual requirements, it also provides an explicit hard stop if you cannot meet it - as well as an important target to aim for.

The overall success can be judged by how close the final solution is to the 'should' requirement(s).

### Contact us

enquiries@ignys.com • 0115 772 2825 • ignys.com

