# ***Requirements Management***

*Poor requirements management is more often than not the culprit causing schedule delays and cost overruns on design and development projects. Generating well-defined requirements, documenting, and carefully managing these requirements is key to a successful project.*

**Project activities in which requirements management is useful:**

- The entire lifecycle of a project but especially in the earlier stages prior to design conceptualization and evaluation of concepts

**Other tools that are useful in conjunction with requirements management:**

- Configuration Management
- Design Reviews
- Eliciting Input
- Engineering Records
- Evaluation Matrices
- Functional Cost Analysis
- Kano Model
- Prioritization Matrices
- Quality Function Deployment

## Introduction

One of the most crucial aspects of any design project is to convert user needs into necessary, verifiable and attainable requirements. These requirements become the basis of understanding between the customer and the project team against which success will be measured.

The requirements management process extends though the entire lifecycle of the project. Due to the often dynamic nature of requirements, careful and thorough management is critical to ensure the customer receives the expected product, on-time and within budget.

### Types of Requirements

There are two basic types of requirements and within each type, there can be a number of sub-categories:

1. ***Project Requirements***
    - Cost and schedule
    - Legal (e.g., confidentiality, environmental laws)
    - Regional/minority benefits specifying a proportion of work that must be done in a specific region or by a specific demographic group
    - Administrative
2. ***Product (technical) Requirements***
    - Functional (qualitative)
    - Performance (quantitative)
    - Design constraints

In general, project or program management will concern themselves with project requirements. Engineering's focus will be on the product but needs to be cognizant of project requirements that impact engineering design and development.

## Application of Requirements Management

### Requirements Process

The requirements process begins with discovering customer needs. In some cases, the customer will provide their own requirements document as is often the case when multiple companies are asked to forward a proposal and bid on a project. In many cases however, the project team must elicit this information from multiple customers including the consumer/end user as well as governing bodies (e.g., Transport Canada). Occasionally, needs are derived by the design team in order to be able to meet customer needs.

Once needs are identified, they are analysed and selected based on their applicability. Prioritization matrices, the Kano Model and QFD are useful tools for prioritizing the selected needs. The prioritized needs are written as requirements and documented in a functional specification. These requirements are updated and refined as technological, and other constraints or opportunities, are discovered during the concept generation process.

### Functional or Product Development Specification

The functional specification is a living document that can developed in any number of formats from a simple table to an extremely detailed document with many sections and subsections. Each functional specification should have as a minimum a title, version number and history, in addition to the product requirements. Attachment A provides an example outline of headings that could be used in a functional specification.

Each requirement within a functional specification should have as a minimum an identifier (often the corresponding paragraph number) and the text of the requirement. Other information that is useful to maintain for each requirement includes:

1. Owner (person on the design team responsible for ensuring the requirement is addressed);
2. Origin;
3. Rationale ;
4. Status (incomplete, proposed, confirmed, modified, satisfied, verified, etc.); and
5. Priority.

## *Writing Requirements*

### Mandatory versus Desired Attributes

Requirements should be written as either mandatory or desired attributes. A requirement for a mandatory attribute is written using the term "shall" (e.g., "The product shall weigh…"; "The product shall be capable of…"). A desired attribute is a design goal that is written using the term "should".

### Referencing Other Requirements

Requirements can reference other requirements documents in whole or in part (e.g., "The product shall be compliant with CAN/CSA International-M2860-M91, *Minimum Access Dimensions for Servicing Machines"* or "The product shall be compliant with sections 2, 4 and 5 of…"). If only a small portion of another document is referenced, including the full text of the applicable sections in the product specification is recommended.

## Requirements Checklist

When writing requirements, there are a number of basic questions that should be asked of each requirement. The answer to each of these questions should be “yes”. If the answer is “no” to any of the questions, the requirement should be reviewed and re-written or even eliminated.

1. ***Is this requirement valid and accurate?*** In other words, does it reflect what the customer needs or is it a derived requirement necessary in order to meet other requirements?
2. ***Is there a need for this requirement?*** To help answer this question, consider if there would be any repercussions if this requirement was not included or the tolerances were reduced. If repercussions are minimal, consider eliminating the requirement or making it a design goal.
3. ***Is this requirement verifiable?*** Determine how the requirement would be verified and the pass/fail criteria.
4. ***Is this requirement attainable?*** Consider the technology available as well as project constraints such as budget and schedule.
5. ***Is this requirement implementation free?*** One of the most common mistakes when defining requirements is stating “how” is to be done rather than “what” it is to be done. Consider if the requirement directs the solution or whether the designers are left to pursue multiple concepts all capable of fulfilling the requirement. Interface-type requirements are the exception in that a specific solution may be directed.
6. ***Is this requirement written in a concise manner?*** Consider whether the requirement is easy to read and understand, and if it contains a single requirement stating only what must be done.
7. ***Is this requirement unambiguous?*** Consider whether the requirement has only one interpretation and leaves no doubt as to the intended descriptive or numeric value. General or vague terms should not be used (e.g., user friendly, easy, maximizes, light, fast, fault tolerant, adaptable, flexible, support). The terms “And/or”, “etc., “but not limited to” and “may” should also be avoided.
8. ***Is this requirement complete?*** Consider whether the requirement requires further explanation.
9. ***Is this requirement consistent with other requirements?*** Review the requirement against other requirements to ensure there is no duplication and the requirement is not in contradiction with any other requirements.

10. ***Is the requirement traceable?*** Consider the origin of the requirement (e.g., customer need, legal, derived for a specific purpose).

### Requirements Traceability

It is good practice to maintain requirements such that their origin can be traced. This helps to ensure requirements are met, unnecessary requirements are not added and if there is risk of not meeting a requirement, it is easy to identify with whom negotiations are conducted.

If the origin is a specification generated by the customer, legislation or a standard, the name of the referenced document and the applicable section numbers is recorded. In the situation where a requirement has been derived in order to meet a customer requirement, the customer requirement number is referenced.

In large systems, it is common to have a functional specification for the system as a whole, and separate specifications for each major subsystem. The subsystems may be further broken down into different components, each requiring a specification. The requirements contained in each functional specification must reference the applicable requirements in the specification one level above.

### Change Management

Managing changes made to requirements is extremely crucial. Projects that fail in design, cost and/or schedule often attribute this failure to poor or non-existent change management.

A requirement change can be an addition, deletion or modification of a requirement. Changes should be proposed to a cross-functional group of representatives (sometimes called a Change Review Board) that review and analyse the impact of the changes on other requirements. The group makes a decision to either accept the change as-is or with modification, or reject the change. Minutes of these decisions are maintained for future reference.

The functional specification is updated to reflect approved changes. The numbering of the requirements in the specification should not be modified in lieu of changes. If a requirement is deleted, simply mark “Deleted on <date>” next to the requirement number. For added requirements, new numbers are used.

## References

Hooks, Ivy, *Writing Good Requirements*, Proceeding of the Fourth Annual Symposium, INCOSE Working Group Papers Volume II. http://www.incose.org.rwg/writing.html (Retrieved February, 2002)

Kar, Pradip and Bailey, Michelle, *Characteristics of Good Requirements*, Proceeding of the Sixth Annual Symposium. http://incose.org/rwg/goodreqs.html (Retrieved February, 2002)

Hooks, Ivy, *Why Johnny Can't Write Requirements*, Proceeding of the AIAA Conference, 1990. http://www.complianceautomation.com/papers/whyjohnny.htm (Retrieved February, 2002)

Mil-Std-490X, *Specification Practices*, June 4, 1985.

Jyrki, Kontio, *Requirements Management,* Course notes for Tik-76.601 Introduction to Software Engineering, Helsinki University of Technology, 1998. http://mordor.cs.hut.fi/~jkontio/req_mgmt/ppframe.htm (Retrieved February, 2002)

Pahl, G. and Beitz, W, *Engineering Design: A Systematic Approach*, The Design Council, London, 1988, pp. 53-55.

# Attachment A

## Functional Specification
## for
## Product XYZ

**Version 2.0**

**February 29, 1999**

| Rev. | Date | Author | Notes |
|---|---|---|---|
| 1.0 | Jan 3/99 | B. Smith | • Initial release |
| 2.0 | Feb 29/99 | D. Jones | • Safety requirements added<br>• Removed transportation requirements |

1. **Product Scope** (what is included/not included in the development of the product)
2. **Functional Characteristics** (functionality and constraints)
3. **Product Characteristics**
   - **3.1 Physical Characteristics** (weight, size, fit with other items, durability, etc.)
   - **3.2 Reliability** (number of operating hours, etc.)
   - **3.3 Maintainability** (tools, servicing intervals, exchange and repair, cleaning, etc.)
   - **3.4 Environmental Conditions** (operation, storage, etc.)
   - **3.5 Transportability**
   - **3.6 Materials** (physical, chemical, etc.)
   - **3.7 Safety** (direct protection, operational, environmental, etc.)
   - **3.8 Human Factors Engineering** (man-machine interface, anthropometrics, lighting, controls/displays, operational posture, etc.)