WRITING SPECS FOR FUN AND PROFIT

Matthew Squair

BE (Mech), MIEAust

Principal Engineer Systems Assurance
UGL Rail

Neil Saville

BE (Mech), FIEAust, CPEng

Chief Engineer
UGL Rail

SUMMARY

Technical specifications play an important part in the agreement between an acquirer and a supplier. Well-written specifications contribute to a successful outcome; financially, commercially and technically; for both parties. A project suite of specifications form the vital link between customer need and a supplier's 'capability'.

The nature and structure of organisations across the rail industry is changing, and so is the scope of contracts for the supply of rolling stock. Today's purchasers may be traditional rail operators, government departments, mining or agricultural companies or private equity firms. This is accompanied by a strong move towards standardisation of product to reduce costs and risks.

This paper examines the subject of technical specifications from the perspective of a major rolling stock supplier and maintainer in the Australian rail industry. The intent is to provoke discussion within the rail industry on a subject that plays a critical role in achieving outcomes, but typically draws little focused debate or discussion.

INTRODUCTION

Technical Specifications are typically the source of a lot of heartache and pain in the Australian rail industry, when they're supposed to ensure the opposite! Truly well-written specifications are a rarity, and not just at the top end of the supply chain. Ultimately, a good specification can support a good relationship between an acquirer and a supplier, but it shouldn't be confused with the relationship itself and it certainly can't be used to fix a poor relationship. But a poorly written specification can contribute to confusion between the parties and deterioration in the relationship.

A common mistake is to write specifications as "fall-backs" for when a disagreement occurs between the parties. This is largely historically based, as each successive failure to deliver the right outcome results in good intentioned practitioners "dictating" solutions through specifications. A more pragmatic view suggests that there wasn't clarity in the agreement in the first instance, and that the addition of further levels of detail will not resolve such a fundamental disconnect.

This paper examines the fundamental questions of 'what is the real purpose in writing a specification' and 'how do I know when I've written a good one?'

NOTATION

AAR	Association of American Railroads
ACOP	Australian Code of Practice
ARA	Australasian Railway Association
ANZR	Australian and New Zealand Railways
OEM	Original Equipment Manufacturer
RISSB	Rail Industry Safety & Standards Board
ROA	Railways of Australia
TBC	To be confirmed
UGL	UGL Limited

SETTING CONTEXTS

1. The Market Context

Historically, rolling stock in Australia has been designed and manufactured to unique technical specifications developed by each operator. Differing gauges in neighbouring Australian states restricted vehicle movement and encouraged local variation. This resulted in a lack of harmonisation of design requirements for rolling stock across Australia.

It is also significant that the nature and structure of organisations across the rail industry is changing. Over the last decade or two, we've seen the significant change from government based

organisations to the major involvement of private companies. There have also been significant changes in many entities from the vertically integrated organisations that were typical of the past, to a mix of vertically and horizontally integrated organisations, along with a large number of organisations that focus only on a single area of the industry.

Operators are now likely to be involved in rail activities across multiple states and multiple application types. Purchasers may be traditional rail operators, government departments, mining or agricultural companies or private equity firms.

The supply chain is also changing rapidly. International companies are pushing into the Australian market, recognising that while it is small it is also growing rapidly and has mature product needs. Local manufacturers are forced to pursue low-cost, off-shore supply chains to compete; and/or to create relationships with international suppliers for either volume or technology advantage.

The scope of supply contracts is also changing. In particular there is a strong move towards standardisation of product. So, where the scope of supply in the past typically included full product design and manufacture activities, more typically today the product is already designed and the scope is to manufacture only, possibly with some minor design modifications to address specific operator needs.

In a number of cases, as they have sought market advantage over their competitors, operators have developed their own product designs. In these cases, the scope of supply is for manufacture only with Design Authority retained by the purchaser.

In a growing number of cases the purchaser is not a traditional rail operator, and does not wish to assume the role of Design Authority in any form, but to rely on the expertise of the OEM.

2. The Commercial Context

The rolling stock supply and maintain business in Australia features a number of inherent risks. There is an immature regulatory environment that is undergoing change. The industry is undergoing continual change to become more relevant and competitive. There is a large exposure to, and interactions with, humans – operators, passengers, and the general public (including rail crossings). The competitive environment features increasing off-shore and on-shore competition as new players enter the market. There is high political exposure, particularly in passenger operations. The market is relatively small, but the customers in the market are known for their desire to push technical boundaries.

For a business like UGL, there are many issues that shape how we respond commercially as a rolling stock supplier. These include;

- the obligation to manage compliance with regulations and industry standards,
- decisions on product offering, i.e. bespoke designs vs. standard product,
- the introduction of new technology and associated risk,
- responding to operators (customers) wishing to push performance boundaries,
- the approach to delivering safety and reliability,
- the management of underlying technical risks, and
- the risk vs. reward arrangements.

There are three principal classes of project risk; cost, schedule and technical. The risk triangle below illustrates the interdependency between these three classes, and illustrates how technical problems can drive both cost and schedule risk while cost or schedule constraints can result in technical risk.

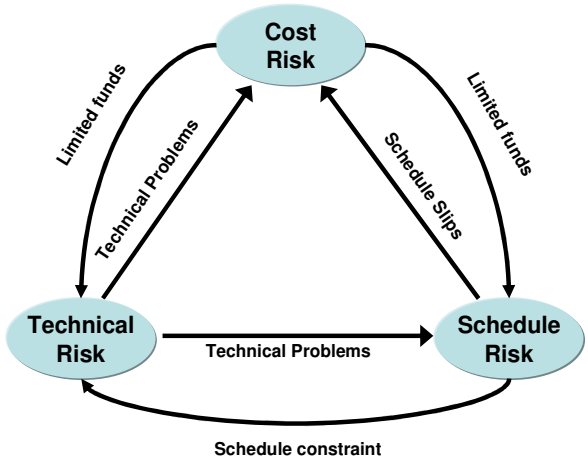

Figure 1: Risk Triangle

As part of the response to technical risk, where possible, open industry standards for common items are used in preference to the development of product unique specifications. Alternatively, these standards may be referenced within product specifications in place of detailed technical requirements.

3. The Technical Context

Railway vehicles are complex machines with many internal interfaces and lots of human interaction. The applications that products are put into can vary greatly from hauling massive iron ore trains in the heat and dust of the Pilbara, to the safe transport of millions of commuters each weekday morning and evening in our capital cities.

There is an expectation from key stakeholders that products that are delivered are safe and reliable, meet all statutory and operational requirements, and deliver on their business plan. This expectation is increasing and never-ending. A rigorous standardised approach is needed that helps to deliver on these expectations and

provides the appropriate evidence that it has been delivered. System Engineering is a widely recognised approach that can help to meet this need.

Systems engineering is a pragmatic and interdisciplinary management process that assures the evolution of an integrated, lifecycle balanced, set of system solutions that satisfy customer and other stakeholder lifecycle needs. The primary output of the systems engineering process is a complete, consistent and correct set of requirements documented in a technical specification. The format of this document and lower tier specifications should follow a standard outline to ensure completeness and promote ease of understanding.

It is highly desirable that a suite of specifications are then developed throughout the product lifecycle and that they are consistent and familial. This should be achieved through a "flow down" of requirements from operator to original equipment manufacturer (OEM) to the various levels of equipment suppliers. While this seems intuitive, it can be difficult to achieve in practice.

This difficulty is believed to stem from embedded culture, particularly in the rail industry. The desire to invent and improve is strong in engineers, but their understanding of commercial realities and how contracts work is on average low. It is also influenced by history, as each perceived failure to deliver the right outcome to a customer results in good intentioned practitioners 'dictating' solutions to old problems through specifications for the next product.

SPECIFICATIONS VS STANDARDS

While each product may have a unique set of specifications that define its performance, these documents typically incorporate references to many Australian and International standards which define items, approaches, processes, materials or test activities to be used in the development or manufacture of the end product.

What's the difference between a standard and a specification? In practice they're both requirements documents, and should share common features and attributes. A specification rises to the status of standard when it becomes recognised by an appropriate authority or given consent by a representative group. Standards facilitate the reuse of learned knowledge and provide a basis of comparison, or an approved model. Specifications are usually applicable to a specific proposed thing (building, machine, bridge), and should provide clarity with respect to that thing.

Standards are used to provide new products with the benefit of previous experience, to promote commonality and to minimise production and support costs. However, the use of standards also needs to be balanced against the cost of compliance and the appropriateness of their use in the specific application.

While standards capture collective knowledge, they are by their nature general rather than specific. The preparation of a specification provides for definition of the implementation of a standard with respect to an intended product. This can be achieved by:

- A simple reference to the standard
- Defining which bits of the standard are considered relevant
- Providing some level of interpretation of the relevant bits

This presumes that there is an appropriate set of standards to reference. The previous rolling stock standards recognised throughout Australia were the manuals produced by the ANZR, the ROA and the AAR. The ANZR and ROA Manuals addressed rolling stock used in standard gauge interstate operations; the most recent was the 1992 version of the ROA Manual. The North American AAR Manual of Standards and Recommended Practices continue to be a major reference for freight rolling stock in Australia.

The Rail Industry Safety and Standards Board (RISSB) was formed by the Australasian Railway Association (ARA) in 2006 to develop and manage the Australian Code of Practice (ACOP). RISSB's mission is the development and maintenance of harmonised standards, rules, and codes of practice and engaging in programs that enhance the safety, effectiveness and efficiency of the Australian Rail Industry. A key strategy in achieving this mission is to promote the adoption of RISSB standards throughout the industry and facilitate harmonisation of Australian rail industry practices.

RISSB standards are, as a rule, not compulsory unless the ARA Board dictates their use. But they are used as the benchmark for the Australian Rail Industry, and they provide a valuable source of Australian based requirements in the preparation of technical specifications.

THE ROLE OF SPECIFICATIONS IN CONTRACTS

Technical specifications form part of the agreement between a purchaser and a supplier. They should contribute to a successful outcome; financially, commercially and technically; for both parties. A project suite of specifications form the vital link between customer need and a suppliers' 'capability'.

To put technical specifications into perspective it is important to consider the following questions:

- What should a specification be, and what should it not try to be?

- What is the difference between a requirements document and a specification?
- What are the various types of specification and when should they be used?
- What's the "right" answer in the age old debate of performance vs. detailed specification?
- How should 'back to back' agreements within a specification be managed?
- What is traceability and how much is enough?
- What are the appropriate stages in the lifecycle of a specification?
- What are the features of a "good" specification and what are the rules for writing one?
- How should standards be used?

But Specifying is not easy…

A specification is generally accepted as a document containing a complete description of what a product will do without describing how it will do it.

So if one asked a train operating company 'what' their need was for a new train-line they might express it in terms of the number of passengers transiting the network per day. But for the same project, if you asked the rolling stock designer for their 'what', they might explain that the need is actually the number of passengers carried per car set and the time taken to load and unload at a station. Finally, ask a passenger car interior designer and they might explain the 'what' as seat pitches and minimum door apertures. Each would argue that they are addressing a need, yet paradoxically their need can also be seen as a solution for the preceding stakeholder.

The resolution of this paradox lies in recognising what Davis (1990) termed the 'what versus how' dilemma or that as Figure 2 below illustrates each level of specification below the first level of user need is both the 'how' for the next higher level and also the 'what' for any subsequent lower level.

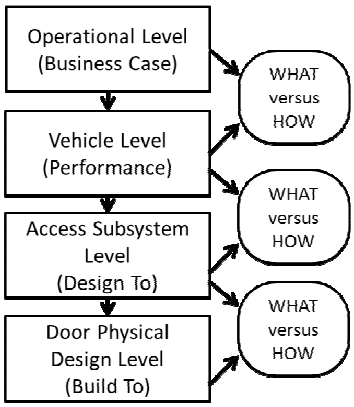

Figure 2: The What Versus How ladder

Each lower level therefore presupposes a level of design decision above. In the case of our hypothetical interior designer their requirements would be in the context of the passenger car's architecture (length, number of doors, single or double deck and so on).

So simply defining a specification as the 'what' is insufficient. We also need to consider what level of the system we are working at, the preceding higher level design decisions and the level of abstraction of the requirements. In the example given, the design context for interior design is quite different for a single deck passenger car versus a double deck car.

One consequence of this is, to quote Pahl and Beitz (2007), that "any attempt to formulate all possible requirements at the start of a project will fail and would cause considerable delays". Another consequence is that as we proceed through the refinement process the types of specifications that we write (and what's in them) will change. At the highest level we are almost purely concerned with a business capability, at the mid-level with 'design to' type requirements while at the lowest we're concerned with the product itself and 'build to' type requirements.

Despite this, most large organisations still proceed on the basis that a complete specification is both practical and desirable *before design commences*. Each stakeholder in the process imposes their own personal wish list, potentially leading to inconsistency and over-specification within the inevitable committee arrangement. Unfortunately at this stage there is no one involved who can advocate for the product's conceptual integrity or critical performance. In passenger car parlance, everyone loves to play with plug doors but no one wants to own reliability.

The result is an over-inflated set of requirements that pay no heed to system constraints. As a real example, in one project the specification for a new DMU required both the incorporation of crashworthiness design features and additional subsystems, yet specified a vehicle mass derived from the preceding generation of DMU, with predictable results.

Performance versus Design in Specifications

As a consequence of the 'what versus how' effect: two types of 'requirements' documents are generally written to define the requirements for a product. Each type of document reflects the roles and objectives of the authoring organisation:

- A performance specification (or requirements document) establishes the services that the item should provide, or the *what*. Generally it should be contractually precise so it can act as an agreement between the item procurer and the supplier. If you are writing this sort of

document you are assuming the role of a customer.

- Design specifications serve as the basis for the detailed design and implementation: in essence they describe *how* the requirements will be achieved. If you're writing this then you are assuming the role of a design authority.

A practical problem can arise when somebody (usually the customer) sits down to write a performance specification but ends up writing a design specification. This is especially prevalent in large organisations where engineers may specify a solution because they 'know' what they want or where the organisation has traditionally acted as the design authority in the past. Problems then arise because the notional design specification doesn't actually reflect the derived requirements for the next level of the design, or is incomplete.

Conversely, if the scope of supply doesn't include design activities then the specification shouldn't include 'design to' requirements.

The supplier (design authority) upon being given a performance specification should be able to derive a design specification. Any customer should be justifiably concerned if the supplier is unable or unwilling to do so.

Contracts and Lifecycles

As Brooks (2010) suggests, in the ideal world we would adopt an evolutionary approach to developing requirements in parallel with the design concepts evolution. But as he also notes, we don't live in an ideal world and as a result we need to establish clear and enforceable agreements to guard ourselves from the misdeeds and mistakes of others (and ourselves). So it seems that we are stuck with contracts and formal specifications and with it the temptation to imagine that we can establish all requirements before we commence the design.

The Completeness Problem

The other major problem in writing specifications is how to figure out when the specification is detailed enough so that we can differentiate between a 'good' solution and a 'not good' solution.

There's no 'magic bullet' for this problem and it affects novice and expert designer alike (Smith (1989) as cited by Leveson (2000)). Unfortunately customers regularly introduce this problem when they transition from writing performance specifications into writing design specifications.

WRITING GOOD REQUIREMENTS

So what's a Requirement Anyway?

As defined earlier a specification contains requirements, but what is a 'requirement'. One definition is that a requirement is the intersection of a capability and need, i.e. "I need 'X', and you can supply 'X'". Another useful way to think of a requirement is as, "an achievable, verifiable statement of need".

What most definitions miss is that a requirement is inherently a method of communication. If you express a requirement so poorly that it's misinterpreted then it's not going to matter how good the subsequent design is. As Popeye remarks "I say's what I needs, and I needs what I says".

What Happens when Engineers get it Wrong

The following is a real example of the consequences of a requirements error:

- As written – "The system shall ignore *all anomalies 20 seconds prior to* shutdown"
- As built – "The system actually *cleared the anomalies list 20 seconds prior to* shutdown"
- What was needed – "The system should have *ignored all anomalies occurring in the 20 seconds prior to* shutdown"
- What happened - The system detected an anomaly within the window of vulnerability, responded and as a result destroyed itself.

While this example is a safety related one, such errors can be no less costly in terms of time and money for less critical applications.

Constructing a Requirement

The most basic construction of a requirement can be expressed as an actor (the thing of interest), performing some act (the action to be taken) upon a target (the focus). So in the preceding example the *system* is the actor, the act is to *ignore* and the focus is the *all anomalies*. The requirement also has a constraint applied, in that the act can only occur in the 20 seconds prior to shutdown.

There also needs to be an imperative that identifies the requirement as just that; a mandatory requirement. In the preceding example the word shall is used. Although different organisations have different styles and rules, you should always clearly identify what term signifies a requirement and which terms are discretionary. For example, should (for a goal) and will (to indicate future intent). One should avoid the terms *may* and *must* however as they grant permission and may be (mis)read as to intent.

Requirements versus design criteria versus constraints

When we write requirements we need to be aware that there are various classes of requirements.

A design requirement is, as noted previously, a specific statement of need that has a design context. A design criteria on the other hand can be thought of as a general (what) principal that has broad application. For example, a (safety) design

criteria for a train door might be that it remain closed when the train is in motion (the what). However the derived design requirement (the how) contained within a design specification for a plug door may be very different to the design requirement for a sliding door satisfying the same criteria.

Finally, there are design constraints. These are requirements that impose a bound on the possible set of design solutions. Constraints may be imposed by the customer (safety, outline gauge, axle limits) or emerge from system resource limitations in the design (power, space constraints and so on). One way to distinguish between a (goal) requirement and a constraint is to ask if the need could be satisfied by not operating or delivering the item. For example one can always satisfy safety by simply not operating the system.

REQUIREMENTS QUALITY ATTRIBUTES

A good requirement should be written clearly enough that you can implement it, and you will know when it has been done correctly.

Over the years the following attributes have been developed as measures of the quality of a requirement:

- Complete (and necessary)
- Consistent
- Traceable
- Concise
- Correct
- Modifiable
- Unambiguous
- Understandable
- Verifiable
- Valid
- Implementation free
- Annotated
- At an appropriate level of abstraction
- Achievable, and
- Context Free

The objective of such a list is to ensure that a requirement is written clearly enough so that you can implement it, and you will know when it has been done correctly.

Completeness and Necessity

As noted previously, canonical completeness is one of the most difficult quality attributes to evaluate. However there are some things that can be done to help in this evaluation. One can look at the set of requirements to see if a requirement is inferred, for example if you specify that a train must accelerate then logically it also has to brake. One can also walk through the sequence of operations (a train door goes through a defined operational duty cycle) or use design standards and checklists to see if anything is missing.

Another aspect of completeness is more administrative: are all the references provided, are all voids in the specification filled in and are there any of those traditional To Be Confirmed (TBC) requirements (much beloved by engineers).

Conversely, we should not include unnecessary requirements. For example do we really need all those design standards? If so, do we need to invoke all of it or just a part? Have we mixed a number of standards together? If so beware the eclectic mixing of standards.

Consistent

In English class we're taught the rule of elegant variation, but for specification writers this is poison. Each requirement should read the same. A specified item must always be called the same thing.

Likewise requirements should not conflict (nor referenced design standards). For example the requirement that an alarm should be 30 dB should not be contradicted by another requirement for 65 dB of noise attenuation elsewhere in the specification!

Traceable

Traceability is an enabling attribute that allows one to determine the global effects of a local change within a specification. At the most basic traceability relies upon the correct referencing of parent specifications, numbering of paragraphs and sections and requirements clauses.

For larger projects and specifications the use of a requirements management tool or database has almost become standard practice and allows the traceability from one individual requirement to another. Where such a tool is used, traceability of requirements to the verification results, or to the rationale for that requirement, can be easily provided.

Concise

One could sum up conciseness as 'don't waffle'. For example:

- "Maximum use of distributed intelligence, under the control of a Primary TOS Unit, is required to minimise traditional wiring and to provide the greatest degree of flexibility and redundancy" *Bad*
- "A TOS controlled distributed intelligence is required" *Better (well at least more concise)*

Remember, specifications are read by people and the briefer you are the more likely they are to read the requirement. A personal recollection by one of the authors was when an experienced engineer was asked to summarise the technical specification within a customer tender; responded

with the best response he could muster, “well it’s 325 pages”.

Correctness

We should accurately and precisely identify the need. In practice we often need to make assumptions and in turn these need to be documented and eventually cleaned up. For example:

> “…the distance travelled shall be considered as 120,000 km per year.”

On a first read this seems clear enough, but is the 120,000 km a maximum or the average? In this case the unstated assumption turned out to be that it was a maximum.

Modifiable

In writing a specification it should be easy to make changes. To do so we need to group like concerns, separate unrelated terms, express requirements separately and *eliminate redundancy*. By clustering like issues we minimise the *distant context* problem e.g. the interpretation of one requirement being contingent upon another requirement in a distant part of the specification, which may be missed.

In practice, developing a template for specifications with standard sections can serve this structural purpose. Likewise the use of paragraph numbering and indented numbering (e.g. section 3.1, sub-section 3.1.1 and sub-sub-section 3.1.1.) can be used to group requirements that are related, to make their later modification easier. However, if you have a thousand requirements all at the same level, then you are likely to have a problem!

One area where we can depart from eliminating redundancy is referencing constant or standard values throughout a specification. While having a list of constants in a single table is fine, referencing the table (and not the value) is unwieldy. In this case including a suffixed subscript label against each value allows us to globally search for and replace such values should they change.

Unambiguous

As Alice remarked to Humpty Dumpty, words in the English language can have several meanings (Carroll 1871), and such multiplicity breeds ambiguity. For example, “There shall be less than 5 amps measurable between *any* two of the three test points”. In this case what does *any* mean? Does it mean all possible pairs, or just one pair?

As a practical aide there does exist a recognised set of ambiguous words that should be avoided as a matter of course, including:

- any, all, and, or,
- affect versus effect,
- host,
- include,
- consist of,
- not limited to,
- comprise, and
- at a minimum.

The word support deserves a separate mention due to it’s wide, and incorrect, usage. Support is a proper term if you want to specify a structural load. But support is ambiguous and incorrectly used if you are stating that a system will *support* certain activities. The usual fix is to simply delete it or replace with the word ‘provide’.

Syntax can also introduce ambiguity, for example, “shunting locomotives may be hazardous”, in this case is the act of shunting a locomotive or shunt locomotives themselves hazardous? We can usually solve these problems with punctuation but that in turn may introduce problems. What does the following sentence mean?

> “The flange shall be fastened by three round-head screws, three flat-head screws and three fillister-head screws all of grade eight”

The more complex the syntax of a sentence the more likely we are to have built in some ambiguity. The use of auxiliary verbs like ‘will’ or ‘should’, the application of modifiers to two or more nouns (actors or targets) and multiple conjunctions are all problem areas.

Another, and very common, source of ambiguity is to introduce an explanation (rationale) with the requirement. The problem is that this can introduce two ways to interpret the requirement. That being said, one should always try to record the rationale for a requirement somewhere as without it assessing the impact of changing requirements at a later date can become problematic.

Understandable

Sometimes we have to express complex requirements and if we’re not careful we can end up in a tangle of concepts.

> “Reliability data for communication on all serial links is to be recorded. Appropriate systems must be implemented for handling of data errors. The error handling systems must include without limitation logging of total data transmitted and received as well as total data errors, a link should not be considered failed before 100% data loss occurs.”

In this case structuring the requirements using sub paragraphs and indenting can make your points more clear.

> 3.1.1 Reliability data for communication on all serial links shall be recorded.
>
> 3.1.2 The TOS system shall handle data errors.

3.1.2.1 Data error handling shall include logging of total:

a) data transmitted and received, and

b) data errors

3.1.2.2 A data link shall not be considered failed before 100% data loss occurs.

The reworked second example above contains all the requirements of the first monolithic statement, but here we can see more clearly how requirements relate to each other.

Verifiable

In quality circles there's an old maxim that 'you only get the quality that you test for'. In systems engineering this may be restated as the principal that, 'if you can't verify a requirement it's not a requirement'. For example on one contract the customer stated this requirement:

"The contractor shall deliver software with a reliability of 10E-9 per hour"

Unfortunately given the extremely low probability such a requirement is not verifiable by any finite and cost effective means. In these circumstances how does the supplier demonstrate contractual compliance and acceptability? Could they say "of course we comply, prove we don't?" And how could the customer prove otherwise?

An organisation's legal position in disputes over acceptability is usually only as good as the demonstrable facts. Opinions, however learned, don't cut it so as a customer it is generally a good idea to specify for particularly critical requirements a test or demonstration as part of the specification to ensure clarity as to acceptability exists.

Another good way to guard against unverifiable requirements is to eliminate the use of ambiguous or qualitative terms such as; maximise, minimise, support, adequate, but not limited to, user friendly and so on.

Operationalism

A little understood aspect of verification amongst engineers is the principal of operationalism, where in some cases we define an item by the process required to make it. For example we specify a cake not by the ingredients list but rather by the recipe.

Specification by operationalism often crops up when dealing with environmental specifications. For example the following track access EMC requirement:

- F1. (Communication system) Vehicles and trains shall generate no energy capable of interfering with the [Access Provider] signalling and communications equipment (covers cable-borne & radio communications).

This EMC requirement is actually defined by the associated test:

- RSU 296 3.1 Interference tests shall be conducted in accordance with the requirements specified in RailCorp Signalling Standard SC 00 18 00 00 SP Rolling Stock Signalling Interface Requirements, C3 Signalling Interference.

Verification in this case is a fundamental part of the requirement and the requirement would be operationally incomplete without its inclusion.

However sometimes specifying an operation can be inappropriate. For example, the statement, "equipment X shall be stowed in the vestibule during train operations" is specifying an operation not a requirement for the equipment. In this instance what was meant was the requirement, "The X equipment shall be designed for the environment of the vestibule cab.". The danger in stating an operation is, as this example illustrates, that the intent may be misunderstood and determining how to verify can be a problem.

Valid

The attribute of requirements validity goes to the question of 'are we building the right system?' If the requirement doesn't express a valid stakeholder requirement, nor can it be traced back to a valid stakeholder requirement, then its validity should be questioned. Traceability is one way to ensure that the requirements set does not grow (or creep) as lower tiers of requirements are derived.

Implementation Free

If we are acting as a customer then we should strive to specify the need not the solution. For example in one contract the following was specified as a requirement:

- "… the Couplers shall incorporate energy absorption devices, which consist of gas hydraulic capsule, deformation tube and tear off elements" (*Implementation*)
- "…the couplers shall absorb a crash energy impact of XXX …" (Need Only)

In the first requirement, the first part is clearly the need while the second is pure implementation, which may or may not work. If the supplier develops the design and finds that implementation does not work, then what part should he or she comply with? The second example requirement expresses the need and is not open to such ambiguity.

The basic rule (and one that is also regularly broken) is that if you are not the design authority don't specify an implementation. But if the contractor doesn't have the competence to assume that role then you as the customer may be forced into doing so.

Annotated

Annotation of requirements is often found in tender documents where essential versus desirable features are specified by a customer. Annotation

can also be used when different variants of a product share a common specification whose requirements may be annotated to indicate effectivity to a specific mark or model.

Appropriate level of abstraction

Abstraction level is dependent on the function of the specification for example:

- …The locomotive shall provide a dead-engine function... *(Loco performance specification)*
- 4.3.3. Dead engine fixture. The brake system shall use brake pipe pressure to charge the main reservoirs. This shall be accomplished by connecting brake pipe to main reservoir through an isolation valve. The Isolation valve shall be labelled 'dead engine' with the valve being labelled 'IN' *(Brake design specification)*

While both these requirements describe the same requirement the level of detail of the requirement reflects the phase of the system development. An attempt to provide the latter requirement level of detail too early in the project exposes the designer (and acquirer) to the risk of premature design commitment.

Achievable

Specifications that are not doable are not really specifications, they're science fiction. This is especially critical if we accidentally write conflicting requirements. This is one of the more common reasons for suppliers seeking contractual relief via legal action.

Another cause of unachievable requirements is failing to consider the allowable tolerance and the cost of specifying tighter than needed tolerances or no tolerances at all.

Context Free

Some requirements rely on others for context. Near context (in the same section) is not so bad, but far context (buried somewhere else), can be very bad. For example:

- The vigilance timer shall have a maximum time to alarm of 65 seconds

The near context was that a fixed duration timer was selected in the preceding paragraph, but the far context was a Driver and Observer Operational concept which dictated a certain vigilance timing sequence.

The antidote for contextual problems is to group or cluster requirements using the structure of the specifications and to stick to a 'separation of concerns' approach to writing the specification. That is, only talk about door requirements in the door part of the specification and not in the crew cab section also.

Unfortunately even with this approach contextual issues tend to creep into large multi-author specifications. In these circumstances locking everyone in a room to do a single read through for context clashes, or appointing an overall 'editor in chief' may be the only true solution. Again, for large specifications, this is where requirements management tools like DOORS™ come into their own as requirements can be cross referenced to the system architecture as well as requirements taxonomies or break down structures[1] to support such reviews.

REVIEWING THE RESULTING SPECIFICATION

Having prepared a specification the final step is to review the specification to confirm that it is complete, correct and fit for purpose.

Will it be understood?

Any specification is a method of communication, so the author should always consider the intended audience and whether the specification will be understood. Simple statements using well understood common terms are rarely misunderstood. If you have to use acronyms and special terms then have you defined what they are?

Is it written at the right level?

Having written the requirements with clarity, it should be considered whether the requirements are written at the right level of abstraction. For example; is it appropriate to provide a high level operational goal type requirement to a lower tier equipment supplier? Or would a more detailed technical design specification be appropriate?

When looking at the question of level of abstraction, consideration should be given to whether each section or group of requirements within the specification adheres to the right level of abstraction. If there are problematic areas that appear to be underdone, the requirements analysis has probably not been completed satisfactorily. This is the trap of 'specifying what's easy to specify'.

A quick check is to count the number of requirements written for a specific section and look at the level of indenture of sub-sections. Deep indenture and many requirements indicate a high degree of detail. Another quick check is to ask 'are we stating both requirement and a solution at the same time?' If so, there is a mixed level of abstraction that needs to be resolved.

Is it complete?

Confirming the completeness of a specification is probably the next step in reviewing a specification. While most basic completeness checks are fairly straight forward (e.g. are all references included?), how should it be determined whether the set of

[1] Requirements taxonomies or breakdown structures identify (in a tree or indented list form) all the specific types of requirements needed to define a product.

requirements is itself complete? Unfortunately there is no easy answer, as the effect of 'problem framing' can cause reviewers (as much as authors) to overlook holes in the specification.

Specification templates and checklists can be used to ensure that all recognised content is included. One basic technique is to require authors to fill unused sections with 'Not Used' rather than deleting them, this serving as a discussion point in reviews. The following is a checklist of requirements types to consider:

- Functionality and Performance,
- Interfaces,
- Operability,
- Reliability and Maintainability,
- Environment,
- Safety,
- Support and Maintenance Facilities,
- Transportation and Storage,
- Security and Privacy,
- Deployment,
- Training and Personnel, and
- Design constraints.

Another technique is to develop the verification requirement in parallel with the technical requirement. Cross walking between verification and technical requirements can often lead to the identification of missing technical requirements as (for example) we consider how the specified item will be tested or operated under test to verify the design.

Consider other perspectives

Most engineers think that specifications are read by other engineers. Engineers are, by and large, a helpful lot, but specifications should always be looked at from other perspectives. For example place yourself in the shoes of a supplier who is in dire financial straits. In this circumstance, the specification reader is not going to care about being helpful. Their objective will be to satisfy the letter of the contract and hopefully avoid bankruptcy.

When looked at from this perspective, requirements that have been written with some assumed charity of interpretation become less firm and more open to interpretation.

Reviewing for quality

Having prepared a specification, the author, or authors, should also review the requirements set to confirm it meets the detailed quality criteria outlined in the preceding sections.

Specific detailed review techniques include:

- Use checklists to identify the use of problem words and phrases,
- Parse out problem requirements into the actor, target, constraint & action form,
- Ask 'why this requirement?' several times until you get to the root reason (which may be the real requirement),
- Ask 'is the requirement necessary?' Who will complain if it is deleted? If the answer is no-one then it's probably requirements creep,
- Ask 'can this requirement be satisfied in more than one way?' If not, then an implementation has probably been specified, and
- Ask 'can it be verified?'

CONCLUSION

Specifications are a critical part of the development of any but the simplest items of supply within the railway industry. The ability to clearly define the purpose and achieve the required level of specification quality is a critical skill for any organisation that is trying to outsource elements of its design activities or acting as the customer in acquiring a product.

The authors believe that the good specification practices identified in this paper will become essential as the industry transforms itself in the decades ahead. If your requirements are incorrect then it doesn't matter how good your design practices are!

REFERENCES

Brooks, F.P., The Design of Design: Essays from a Computer Scientist, Pearson Education Inc., Boston, 2010.

Carrol, L., Through the Looking Glass, And What Alice Found There, 1871.

Leveson, N.G., Safeware, System Safety and Computers, Addison Wesley, 1995.

Leveson, N., Intent Specifications: An Approach to Building Human Centered Specifications, IEEE Trans. On Software Engineering, Vol 26, No.1 2000.

Pahl, G., Beitz, W., Engineering Design: A Systematic Approach, Springer-Verlag, Berlin, 2007.

Smith, G.F., Representational Effects on the Solving of an Unstructured Decision Problem, IEEE Trans. Systems, Man, and Cybernetics, pp. 1,083-1,090, vol. 19, 1989.