Requirements management – a case study

J. Lal
Systems Engineering, Alcatel Transport Automation Solutions, Canada

Abstract

Mass Rail Transit projects are usually large, multi-year undertakings that often experience significant changes to the contractual requirements as the work progresses. From a contractor’s perspective, projects that are large in terms of capital investment and duration are usually large from a requirements perspective as well. Experience shows that these requirements must be carefully managed if the work is to be delivered successfully.

This paper describes Alcatel Transport Automation Solutions’ experiences with requirements management with the London Underground Limited Jubilee Line Signal Control System project. The approach described in the case study is, of necessity, practical and obviously successful because it survived the entire lifecycle of the project’s eight-year plus duration. This approach to requirements management benefited all stakeholders in the project, including the customer.

Definitions

Requirement: In the following discussion, the term ‘requirement’ generally refers to the mandatory technical requirements that the project team must address when developing the technical solutions. These requirements can be customer imposed, internally generated, or be any combination of the two.

Requirements management is the process of maintaining an approved, controlled, complete, and up-to-date set of requirements that the project team can reliably use in various ways for delivering the project.

Traceability refers to the ability to link any piece of information from any part of the project’s development artifacts (e.g. design, test, and validation documents, and code) to the source requirements.

Elaboration is the process of determining successively more detailed requirements from a given higher level requirement.
1 Introduction

Mass Rail Transit projects are usually large, multi-year undertakings that often experience significant changes to the contractual requirements as the work progresses. From a contractor's perspective, projects that are large in terms of capital investment and duration are usually large from a requirements perspective as well. Experience shows that a large set of requirements undergoes many modifications due to scope changes, to technology evolution between the times that the projects are bid and when finally implemented, due to clarifications that emerge only after the projects get underway. In extreme cases, the requirements can undergo drastic changes due to fundamental shifts in programs and/or operating philosophies.

The large set of requirements must be carefully managed if the work is to be delivered successfully. While requirements management is not difficult to understand from a theoretical basis, implementation can be thwarted for reasons such as lack of proper management tools, lack of acceptance because of unwieldy implementations in the past, lack of disciplined approaches to requirements-driven project deliveries, or perceived costs.

This paper describes Alcatel Transport Automation Solutions' experiences with the London Underground Limited Jubilee Line Signal Control System project. This large, high profile project exhibited many of the characteristics highlighted above: a large number of requirements, changes in program, and changes in scope. The approach described in this case study is, of necessity, practical and obviously successful because it survived the entire lifecycle of the project's eight-year plus duration. This approach to requirements management helped not only the design, verification, and validation activities with the customer, but also the formal acceptance and contract closeout activities. The payoffs were to the contractor as well as to the customer. This paper describes the sources of requirements, the kinds of changes that occurred, and the approach to devising and implementing a requirements database that served the needs of all stakeholders in the project, including the customer. Tool characteristics and the costs and benefits are also described.

The Customer and the Project

London Underground Limited (LUL) is the overseer of the majority of the transit lines serving the city of London, England. This transit system is one of the world’s largest in terms of number of lines, complexity of service, and size of daily ridership. These characteristics place huge demands on LUL to provide an adequately reliable, and high throughput service.

The Jubilee Line Extension Project targeted expansion of rail transit services into the East End of London. It involved the extension of the Jubilee Line from 17 stations to 28 stations. Massive civil engineering works consisting of new tunnels, stations, depot, and control center were undertaken. Supporting infrastructure such as optical fiber-based communications facilities, and closed circuit television were also provided. The program required new signaling and signaling control for the entire line, and each of these had large dependencies on
works components of other contractors. The fixed-block signaling on the old part of the line was to be replaced also by moving-block signaling. The existing control center at Baker Street, which simultaneously controlled both the Jubilee Line and the Metropolitan Line services, was to be modified so that control over the Jubilee Line could be separated from Baker Street and be transferred to the new control center facilities purpose-built at Neasden. The complexity of the entire undertaking was obvious from the outset. To compound matters further, LUL was required to deliver the extended Jubilee Line well in time for the millennium celebrations sited at North Greenwich.

Alcatel Transport Automation Solutions was awarded the contract by LUL for the new Signal Control System that would control service over the entire line and in the new depot. This system had to interface to many other systems, principally the new moving-block based signaling system, and support systems comprised of CCTV, Radio, Power, Tunnel Ventilation, and Passenger Information systems. Alcatel's contract also called for a Simulation and Development System that would allow operations and maintenance staff to be trained, and software changes to be made and tested before introduction into revenue service. There were many dependencies on other contractors. Failure to deliver on time could precipitate large penalties. The project was classified as safety-related, which meant that the development processes, although not as rigorous as those required for safety-critical systems, nevertheless required highly traceable documentation.

Project Statistics
The size and complexity of the Jubilee Line Signal Control System project can be illustrated via the following statistics (Ko [1]):

a. Extended Jubilee Line Ridership (2001) > 450,000 per day
b. Interfacing External Contractors 4
c. Number of Program Baselines 9
d. Number of Requests for Contract amendments - > 400. Of these over 260 were executed.
e. Number of Software Releases to the field - approximately 50 in 40 months
f. Number of software lines of code - 500,000
g. Project staff -approx 100 peak, spread over Vancouver, Toronto, and London development centers
h. Number of requirements - approximately 2900 initially, 5300 currently

The project was started in December 1993, with an original completion date of March 1996. From mid-1995, a series of program changes were brought about by delays in various contractors’ works, fall-back to fixed-block signaling, and changes in the commissioning approach. With the approaching deadline for the millennium celebrations, further decisions were taken by the customer to focus on the new extension only. Transfer of control from the Baker Street control center and control over the original half of the line were deferred. Alcatel, like all other contractors, had to quickly adapt to these large program changes. It is this climate that really brought out the benefits of the requirements database and the requirements management process.
2 The Requirements Database

When the project started, the engineering group created a simple requirements database (RDB) with a rather modest objective of singling out the true requirements so that they could be allocated amongst subsystems and the various engineering departments for development purposes. However, as the project progressed, and the tool on which it was based became more powerful, the RDB started to be used by virtually the entire project team, including the commercial groups and the customer.

The RDB was created from the following contractual customer documents: the General Specification (GS), the Particular Specification (PS), and subsequently the Engineer’s Instructions (EIs) that provided the contractual variations. The GS contained requirements that were common to all contracts, e.g., documentation and materials requirements. From the GS, only requirements applicable to the Signal Control System were entered into the RDB; others were reported back to LUL for confirmation that they were indeed not applicable to our contract. This step ensured that there were no differences between Alcatel’s and LUL’s understanding of the requirements. Culling the set in this manner with the customer’s concurrence early in the lifecycle meant that our staff did not repeatedly “process” requirements that did not apply to our work. This saved valuable time.

The PS contained requirements specific to the Signal Control System, and all requirements from the PS were entered into the RDB. Contractual requirements such as the terms and conditions were not included in the RDB, as they did not require the same kind of management. The EIs were the formal variations to the required scope and were incorporated as the customer issued them.

Database Structure
Presently the RDB has 5 major inter-linked modules, i.e. Requirements, I&T Test Procedures, I&T Test Results, T&C Test Procedures, and T&C Test Results. These are loosely called the RDB. The following table lists a sample of the attributes being used in the requirements module

<table>
<thead>
<tr>
<th>Item</th>
<th>Attribute Name</th>
<th>Attribute Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Requirement Number</td>
<td>This is a unique number assigned to the requirement by the RDB maintainer.</td>
</tr>
<tr>
<td>2</td>
<td>Sub-Requirement Number</td>
<td>This field is used when a given requirement had several sub clauses that needed to be tracked separately from the main requirement, but yet needed to be grouped with the main requirement. For example min and max values of configurable time-outs could be listed as sub-requirements.</td>
</tr>
<tr>
<td>3</td>
<td>Requirement Description</td>
<td>This field contains the actual text of the requirement. It generally contains a statement with a single ‘shall’ clause.</td>
</tr>
<tr>
<td>4</td>
<td>Requirement Source</td>
<td>This field identifies the parent document in which the requirement originated. The values in this field were typically PS, GS, or EI</td>
</tr>
</tbody>
</table>
### Verification and Baselining

Once the RDB was set up, the QA department verified that the RDB correctly and fully reflected the technical requirements of the contract. This was of utmost importance because the RDB was to be used to drive the design and test work in lieu of the original contract documents, which, though still the contractual specifications, remained relatively unused from then on. Following verification, the RDB was baselined and then made available to other departments for their use. Baselining meant that the general user could not change the critical fields.

### Contract Changes

The customer issued formal contract changes via the EIs. The text of the relevant requirements in the RDB was updated as each EI was received and accepted by our Configuration Control and Review Board (CCRBD). The changes were checked and the RDB re-baselined.

### 3 Stakeholders and the Requirements Management Process

When the RDB was started in the Vancouver office, a requirements-driven development process was new to the company. Neither the engineers nor management knew the costs or what to expect from it. With time, however, the entire project team as well as the customer began to rely on it. The customer was given visibility into the RDB from the start, and once they witnessed the rigor of the process, they accepted it as the main driver of the project’s work. Responsibilities of various stakeholders evolved to the following:

#### 3.1 Systems Engineering

The systems engineering department was the main technical interface with the customer. It was responsible for analyzing and clarifying requirements, and for

<table>
<thead>
<tr>
<th>Requirement Source Ref</th>
<th>Comments</th>
<th>Build Number</th>
<th>SW Allocation</th>
<th>IT Procedures</th>
<th>IT Results</th>
<th>TC Procedures</th>
<th>TC Results</th>
<th>Validation Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>This field identifies the specific section number in the parent document that sourced the requirement.</td>
<td>This field contains any comments that the users wished to record. It usually contained the clarifications provided by the customer.</td>
<td>This field is an identifier for the earliest software version released to the field in which the requirement would be met.</td>
<td>This field identifies the software subsystem that the requirement is allocated to. The field is left blank for non-software requirements.</td>
<td>This field lists the identifiers of test procedures written by the Integration and Test group for tests conducted in the factory.</td>
<td>This field lists the identifiers of the I&amp;T test results that were recorded in the factory.</td>
<td>This field lists the identifiers of test procedures written by the Installation and Commissioning for tests conducted in the field.</td>
<td>This field lists the identifiers of the test results recorded in the field.</td>
<td>This field identifies QA department’s validation of the test results against the requirements.</td>
</tr>
</tbody>
</table>
translating those requirements into the system architecture and the top-level system designs. It was therefore logical that the systems engineering department took on the responsibility for setting up the RDB, and for maintaining its core for the duration of the project. Only designated individuals were permitted to add, delete, or change the text of the requirements. This meant that the RDB could be very tightly controlled, with very little chance of unauthorized changes. In short, the systems engineering department was responsible for ensuring that the requirements were as meticulously maintained as possible. The resulting RDB formed the solid platform that guided the rest of the project team. Details of the RDB processing were as follows:

**Requirements Allocation** The systems engineering group used the requirements database to allocate the requirements to the various hardware and software subsystems. As would be expected, the mappings were one-to-one, one-to-many, or many-to-many. In so doing they examined each requirement, compared it against the statement of compliance that the company had submitted with the proposal, and against any changes that the customer had made when the contract was negotiated. The above process of course meant that each and every requirement was deliberately examined and acted upon. This process greatly reduced the likelihood of requirements being over looked.

Mapping to software and hardware subsystems also meant that the corresponding engineering departments were also automatically assigned responsibilities for these groups of requirements.

**Verification** The systems engineers also identified the most suitable manner in which each requirement was to be verified during factory testing and during field commissioning, e.g. by test, inspection, or analysis. This exercise then assisted the test departments to develop their test plans early in the life cycle. And it also meant that testability was also factored into the designs.

**Clarifications** As the development work began, many requirements required clarification from the customer. Enquiries were then sent off to the customer, and notes were added to the RDB with relevant details, principally the date and the reference number of the correspondence. As the customer responded with clarifications, the texts of the clarifications and other tracking details were recorded in the RDB as comments. The requirement text was altered only when the customer formally issued wording changes that were also accepted by our configuration control and review board (CCRB).

**Elaborations/Derivations** The requirements stated in the initial contract were very detailed and generally did not require extensive elaboration. However, pressures caused by large program changes and slipping schedules began to result in more high level requirements from the customer, and these meant that Alcatel had to elaborate the requirements through detailed interactions with the customer. The exercise was time consuming, and at times resulted in conflicting requirements from different groups. There was also the concern whether the
derived requirements fell outside the scope of the contract. However, the exercise proved to be of great benefit to all concerned. Requirements elaboration meant that the details adequately bounded the functionality. The customer formally approved these detailed requirements before they were entered into the RDB. The elaboration/derivation process was beneficial to the engineers as it defined in great detail what the design was required to meet. Test engineers found it very easy to generate their test procedures from this level of detail. Consequently, the design and test reviews were smoother and generally quicker.

**Release Allocation** As the project management office determined the specific subset of the functionality to be delivered in each release to the field, the requirements were labeled with the specific release number in the RDB. The RDB therefore instructed other departments what they needed to design for the next release.

### 3.2 Hardware Engineering

The Hardware Engineering department was responsible for the detailed design and procurement of all hardware subsystems identified by the systems engineering. It worked from the requirements in the RDB allocated to the hardware subsystems. Design documents and drawings captured their work. When design reviews occurred, the allocated requirements formed the basis of the review.

### 3.3 Software Engineering

The Software Engineering department was responsible for the detailed design, coding, and debugging of the software subsystems. It worked from the requirements that had been allocated to software subsystems. Design documents, and code listings were the principal outputs. Reviews of their designs were also centered on the allocated requirements.

### 3.4 System Integration and Test

System Integration and Test (SIT) department was responsible for verification activities in the factory. They prepared the test procedures and carried out the tests. The test procedures and results were held in two modules separate from the requirements module. Each test procedure explicitly listed all requirements that it addressed, and each requirement pointed to the relevant test procedures. Test results pointed back to the procedures. In other words, all of the key information required for validation was readily available. Approximately 2200 test procedures were created.
3.5 Test and Commissioning

The Test and Commissioning department was responsible for installation, test, and commissioning activities in the field. This group wrote the test procedures specific to testing activities in the field. The test procedures and results were managed in the same manner as were done by the SIT group. They created 2 additional modules for the procedures and results, also linked to the requirements module. Over 500 procedures were created. Since it was unwieldy to manipulate the RDB from the London field site because of inadequate link bandwidth, updates to the database were forwarded to the factory on paper for action.

3.6 Quality Assurance/Auditors

The Quality Assurance department was responsible for validating that the products and processes complied with the customer’s requirements, including the company’s ISO 9001 requirements. Internal and external auditors provided independent checks of all aspects of the development processes. Validation was easy because of the linkages between the requirements, test procedures, and test results. To boot, all of the associated history was also available with the requirements. The QA and auditors were perhaps the most enthusiastic supporters of the requirements management processes because the formality and the rigor made their jobs so much easier. They had the tendency to demand traceability to requirements throughout all stages of the development process. Our experience showed that traceability information was best left to the verification and validation artifacts only.

3.7 Contract Administration

The Contracts Administration group was responsible for handling the contractual aspects of all contract variations. It was supported by the Configuration Control Review Board whose mandate was to review and approve all changes that impacted the contracted scope and price. Approved high level technical requirements were forwarded to the Systems Engineering Department for elaboration and implementation. The elaborated requirements and the information trail recorded in the RDB clearly bounded the functionality, and showed the history. These details were very useful to the contract administrators in their interactions with the customer.

3.8 Customer

The customer’s main concern was that we, as the contractor, were working with the most recent versions of requirements, and that they correctly accounted for all clarifications, and scope changes, and that we did not leave out any requirements. They also required clear links between the requirements and the associated test procedures, test results, and validation reports. Proof of adequacy of test coverage was critical to their acceptance of the system. Knowing that the
RDB had been kept up to date, and that it contained a wealth of historical information, the customer found it an excellent basis of discussion when discontinued requirements and suspended work had to be determined for contract closeout.

4 Tool Characteristics

Our experience showed that requirements management was greatly facilitated if a purpose-built tool is used. When we started the project, requirements management tools were rather crude and expensive. As mentioned earlier, we started by setting up the RDB with Borland’s db4 database software package, and then migrating it to the Microsoft Access database platform. We soon found that the RDB was not stable, did not have adequate multi-user support, and severely lacked features. It was also very awkward to re-create some of the more advanced features available in purpose-built tools. Finally, the RDB was migrated to the DOORS requirements management tool.

For our needs, the important features in a tool were:

a. The tool must allow multiple users to access the RDB simultaneously.
b. The tool must handle several thousand requirements, and the associated linked data such as correspondences, test procedures and results. Also, the tool must allow rapid access and be stable. Slow response times greatly discourages usage.
c. The tool must provide good configuration management features, especially baselining, the ability to track history, and the ability to restrict read/write access as controlled by a tool administrator.
d. The tool must allow forward and backward traceability among requirements and other linked data. These linkages could be one-to-one, one-to-many, many-to-one, or many-to-many.
e. The tool must provide typical data manipulation features such as filtering, sorting, and searching. It must also allow a requirement to be viewed in its larger context (as is in the parent document).
f. The tool should allow importing/exporting of data from/to typical word processing products available in the marketplace. This feature greatly facilitates setting up of the RDB from user-provided soft copies of the contractual requirements.
g. The tool must allow users to link their own material to the baselined RDB. In this way, the testers could link their test procedures and results to the requirements without the risk of altering the requirements per se. Each group of users must be able to control read/write access to its own data.
5 The Costs and Benefits

Costs & Lessons Learned
The RDB evolved as the underlying tool became more powerful, and staff became more accustomed to and reliant on requirements-focused delivery. Several important lessons were learned through this evolution.

a. Management must buy into the process, and provide resources for setting up and continuously maintaining the RDB. This includes purchase, training, and ongoing maintenance costs. It requires that the tool be made available to all project team members who need to use the database. If the RDB is not properly maintained throughout the project’s lifecycle, users will lose confidence in it and will eventually abandon it.

b. It is far better to acquire a commercial purpose-built tool from the outset than to attempt an in-house development.

c. The entire project team needs to accept that work will be done to meet the requirements, no more and no less. This understanding is important especially if requirements creep is to be avoided, and costs controlled.

d. The project team must define a development process that is based on formally controlled requirements, including how the requirements are to be used by each department.

e. The issue of traceability has pitfalls, as we learned. An organization new to the process of requirements management often sees the benefits of keeping traceability information through all phases of a project’s lifecycle, but fails to realize the associated costs. Our experience with adding traceability information to code and design documentation proved to be very costly because the design and code kept changing, with a knock on effect on the traceability information. The well-intentioned efforts soon fell apart as the usual time and cost pressures mounted. If traceability information is not implemented consistently, then the verification process cannot rely on it and it will be of little benefit. It is therefore necessary to determine how much traceability to the requirements is to be carried in the project’s artifacts, especially in the design, code and verification documentation. The guiding principle here is that one must be pragmatic.

f. Postponing the insertion of traceability information until after the design and testing work had been done failed because management was reluctant to add in the “frill”, and the Quality Assurance department was justified in rejecting any alteration of code and/or documentation that had already been tested and released. Hence, the traceability information should be consistently kept up-to-date as the work progresses.

Benefits
The following benefits were distinctly noted:

a. Having the RDB set up as a network-enabled database meant that the full power of computer tools could be used to process the requirements in numerous ways. Also the entire project team could use the RDB.
b. When requirements were allocated to departments and subsystems, the likelihood of overlooked requirements was greatly reduced. Once the allocations were done in the RDB, the design, implementation, and verification teams had early visibility into the next release of the software. In particular the testers could get an early start on test procedure preparation.

c. Because the engineering team was conditioned that the designs must address only the specified requirements, requirements creep was virtually eliminated. A very beneficial side effect of this mindset was that staff began flagging out of scope work to the customer’s attention. These could then be dealt with through the contract change process.

d. Having a single group responsible for maintaining the requirements, the job was done consistently and with high accuracy. This resulted in high confidence in the RDB, and it became a critical resource to the entire project team.

e. Having test procedures and test results linked to the requirements, test coverage could be easily determined. As a result, the QA department could rapidly satisfy itself that software could be released to the field, and it could rapidly generate the release documentation. This shortened the time between testing and actual release to less than a day. Validation was comparatively easy.

f. Given that the RDB contained up-to-date clarifications of the requirements and validation information in addition to the text of the requirements, virtually the entire history was available in a value-added way. A user did not have to wade through paper copies of correspondences to locate discussions with the customer regarding any given requirement.

Customer Benefits
Although not initially envisaged, the customer also benefited from the requirements management process.

a. Requirements elaboration occurred in close consultation with the customer’s own stakeholders, and the functionality was bounded with their concurrence.

b. The customer could readily trace requirements to the verification documentation needed prior to each release to the field. It also helped with their assurance activities that commissioned equipment properly met the requirements.

c. The traceability information greatly facilitated validation, periodic audits, and final customer acceptance.

d. The customer was satisfied that the RDB was kept up-to-date and therefore it could be used in lieu of their original requirement documents and the myriad pieces of correspondence for day-to-day working. They periodically received copies of the RDB for their own information trail.

e. The customer had positive evidence of the process’ rigor.

f. The elaboration process reduced the burden on the customer to sort out the details fully before instructing changes.
6 Conclusion

The Extended Jubilee Line was opened in time for the millennium celebrations and the Signal Control System continues to operate the service reliably on the Extension. As the first Jubilee Line contract with LUL draws to a close after an eight-year run, the requirements database is taken for granted as the backbone that coordinated the entire project team to deliver successfully. Perhaps the most ringing endorsement of the requirements management process is that it is now firmly part of the corporate culture, and one cannot imagine working without it! Our Jubilee Line project has proved that to deliver large, multi-year rail transit projects successfully, a managed, requirements-centred delivery process is the way to go.

Acknowledgement

The author is grateful to the management and staff of Alcatel Transport Automation Solutions for encouraging and materially supporting participation in this and other conferences. A note of thanks is also extended to London Underground Limited and to Infraco JNP Limited for their dedication and support in making the requirements management process a success.

References