

# Railway Systems Engineering in Action

**Dr. Jon Elphick and Mark Irving**

Atkins Rail Ltd, Holgate Villa, 22 Holgate Road,  
York North Yorkshire, YO24 4AB, United Kingdom  
([Jon.Elphick@Atkinsglobal.com](mailto:Jon.Elphick@Atkinsglobal.com), [Mark.S.Irving@Atkinsglobal.com](mailto:Mark.S.Irving@Atkinsglobal.com))

## ABSTRACT

One of the ways that Atkins Rail has recently been delivering real benefits to the rail industry has been through the application of Systems Engineering – this paper describes how, using a number of practical examples to illustrate our successes. One high-profile example is the Concept Design developed for the Victoria Line Upgrade (VLU) Service Control Centre (SCC) that required the deployment of an integrated, multi-disciplined team.

Our approach is based on the rigorous application of a set of basic Systems Engineering principles, and the use of a novel method of managing requirements. Our technique engages the whole team in the **triage** of source documentation and then in **tagging** the location of requirements. We have demonstrated how engaging the whole team in the practice of Systems Engineering results in a deep collective understanding of the problem, produces results very quickly, and we avoid the need to re-write requirements to comply with accepted norms of requirements language.

The paper outlines the benefits of using a requirements management tool (DOORS™) and Atkins iProNET, a web-based collaborative working environment.

Our approach has been employed on projects that were completed successfully, within demanding time scales; they provide a real illustration of the benefits that Systems Engineering can bring to a challenging and complex project.

## INTRODUCTION

Atkins Rail has long been at the forefront of innovation in the rail industry, and Systems Engineering is no exception. Our Systems Integration Initiative has been underway within the company for 4 years; its ambitious aim is to change the culture of railway engineers across the board, by encouraging the use of 18 fundamental Systems Engineering principles. The initiative is supported by a manual of industry best practice, which provides practical examples of how Systems Engineering principles can be applied to everyday projects, by all engineers. It is our strong conviction that we can make the greatest contribution to the rail industry as a whole, if the majority of engineers apply the fundamental principles, as opposed to focusing on a small number of specialists. This paper describes an approach we have developed based on these principles and some practical examples of its use.

The principal example described in this paper is the concept design development for the Victoria Line Service Control Centre (SCC). The SCC will have a critical role in the operational management of the train service, as it will house both critical equipment and the London Underground (LU) staff with responsibility for strategic, service related decision making. The challenge was to collate and assess a large and disparate array of requirements, and develop them into a Concept Design for the SCC – the scope included everything within the boundary fence of the new SCC. The functions provided by the SCC include:

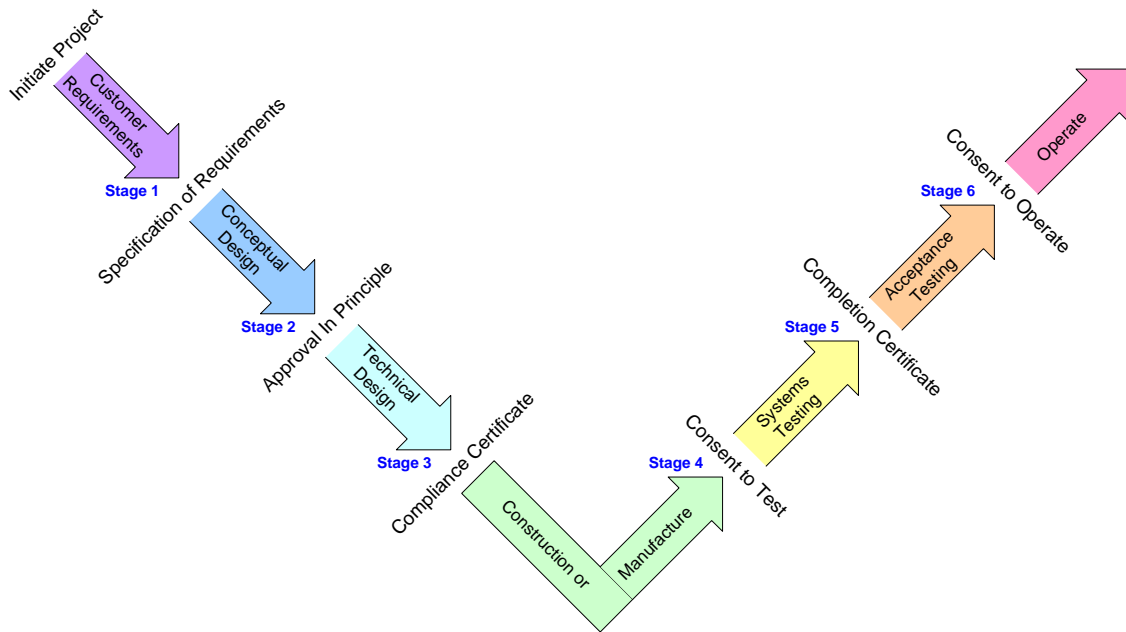
- Automatic signalling and train control, supervised by control room staff
- Voice communications between control room staff, platform staff, train drivers and other railway staff
- The provision customer information across the Victoria Line

Although the Victoria Line is just 21 km in length and serves only 16 stations, more than 500,000 passenger journeys are taken on it each weekday, making it a key artery into the capital. This, combined with the challenging timescales for the overall Victoria Line Upgrade Programme, meant that it was of critical importance to deliver an SCC Concept Design that would meet the tests of necessity and sufficiency, and minimise future re-work.

The Victoria Line Upgrade Programme falls under the jurisdiction of the London Underground Systems Engineering standards and specifically the E1008 standard, which controls the introduction of new and changed assets onto their network (London Underground, 2002). E1008 describes the overall system lifecycle (reproduced in Figure 1), and defines the Conceptual Design or Approval in Principle (AiP) stage, which takes the output of the user/customer requirements specification stage and develops it into an outline or concept design. The Approval in Principle stage, which is analogous to preliminary design, must meet two key objectives:

- Develop the outline or concept design to a point where it supports a design and build tender package.
- Support the concept design with an assurance case that demonstrates how all the relevant requirements have been appropriately considered.

The end of the Conceptual Design stage is controlled by the Approval in Principle stage gate; at this point the concept design and associated assurance case is



**Figure 1 The London Underground System Development Lifecycle**

reviewed by the Infrastructure Owner's Asset Engineer and approved.

Atkins Rail has undertaken many projects at this conceptual or preliminary design phase and in our view it is always worth reinforcing the distinction between conceptual and detailed design. Engineers are naturally drawn into detail; this must be resisted as it leads to delays in the concept design stage and will commercially disadvantage the purchaser by restricting the scope of the design/build contractor. The approach described herein includes strategies that are specifically designed to delay the onset of design work, thereby providing increased time to understand the whole problem prior to the onset of more detailed thinking, whilst not increasing the overall project delivery timescale.

The scope of the VLU SCC project is to build a new SCC and equip it to meet the operational and maintenance needs of the railway for the next 30-40 years. The concept design had to provide a conceptual solution to the whole problem, which could be subdivided into packages of work for further design and development. The source of the business requirements for the project is the PPP contract. This is a performance contract that defines the principal driver for the line upgrade as an increase in line capability. The requirement is specified as a required Journey Time Capability (JTC), which is measured in minutes. It is a theoretical measure of the average journey time of the average passenger, augmented with a number of additional factors. JTC can be related to traditional railway engineering measures such as signalling headway and carriage (car) design features. However, in respect of the SCC these requirements essentially flow down to a requirement to provide SCC functionality by a certain date; they do not provide measurable targets for SCC systems.

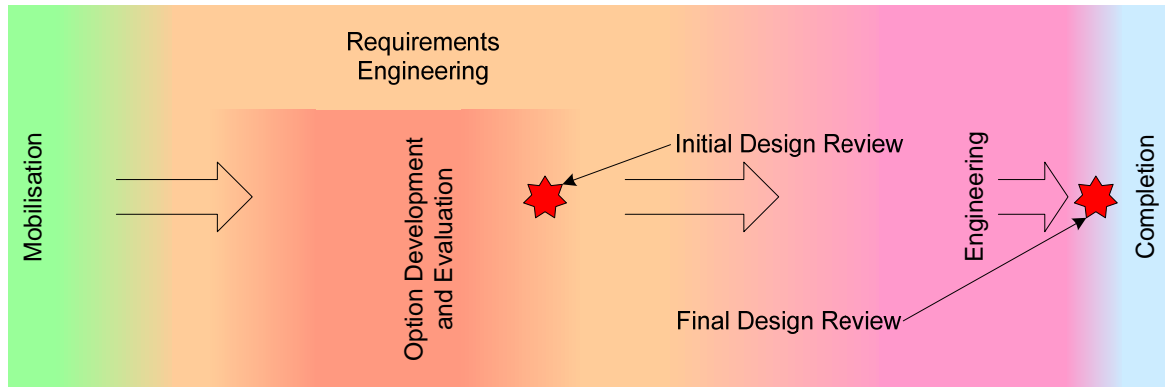
In addition to the capability measures, the PPP also mandates performance in respect of availability of assets and longevity of operational life. These requirements can more easily be related to the SCC.

This paper also references a similar project that Atkins Rail has undertaken for Tube Lines, where we were responsible for developing the concept design of the communications systems for the Jubilee and Northern Line Upgrade Project SCCs. While the scope of this project was more constrained, the same systematic approach was adopted and it fulfilled exactly the same lifecycle stage as in the VLU SCC Concept Design.

The final example that we will use is the development of a detailed design for a very large computer-based supervisory control and data acquisition (SCADA) system. The system in question provides a business critical function for one of the world's largest airports. This final example demonstrates that the approach we have developed is applicable to systems outside the railway industry and to different lifecycle stages.

The structure of our paper mimics the overall make-up of the approach itself, which is shown diagrammatically in Figure 2. The approach starts with a **mobilisation** phase, which is designed to ensure that all the necessary procedures, templates, and tools are in place, and to ensure that all the team members are given time to familiarise themselves with the project and the approach.

The second phase is **requirements engineering**. This phase engages the whole team in identifying and classifying the requirements; this phase is crucial in the approach of embedding understanding of the project's objectives within the team. Depending on the nature of the scope, the **option development** phase is employed to a greater or lesser extent. The



**Figure 2 The Generic Systems Approach**

principle objective of this phase is to analyse the business requirements, and to identify and compare alternative solutions. The **engineering** phase is where the chosen option(s) is(are) developed to produce a solution that is compliant with the whole spectrum of requirements and domain knowledge. The final phase facilitates the completion of the work and the production of the client deliverables.

The paper concludes with our reflections on the approach, its successes and failures to date and where it goes from here.

### MOBILISATION

The key objective of the mobilisation phase is to customise the generic approach to suit the specific application, and to ensure that this is documented and that all the necessary tools and templates are easily available for the team. The primary vehicle we have adopted for this is a document called the Design Management Plan (DMP); this might also be called a Systems Engineering (management) plan, an assurance plan or (perhaps 10 years ago) a quality assurance plan.

The DMP must be simple and straightforward; the authors strongly advocate that clear, easy-to-understand plans and procedures will facilitate sound decisions. The documents we have developed to date have generally included the following:

- Project remit and background
- Project organisation and team member responsibilities
- Project activities during each project phase and a list of deliverables for each phase
- Level 1 programme (a structured work breakdown structure of 30-50 activities)
- Descriptions of the specific management processes that will be employed for the project; e.g. design review process, issues management process, configuration control, document management, etc. Appendices support management processes as necessary, in order to keep the main body of the document to an easily manageable size (we suggest less than 30 pages). This also facilitates reuse of procedural, project independent material.

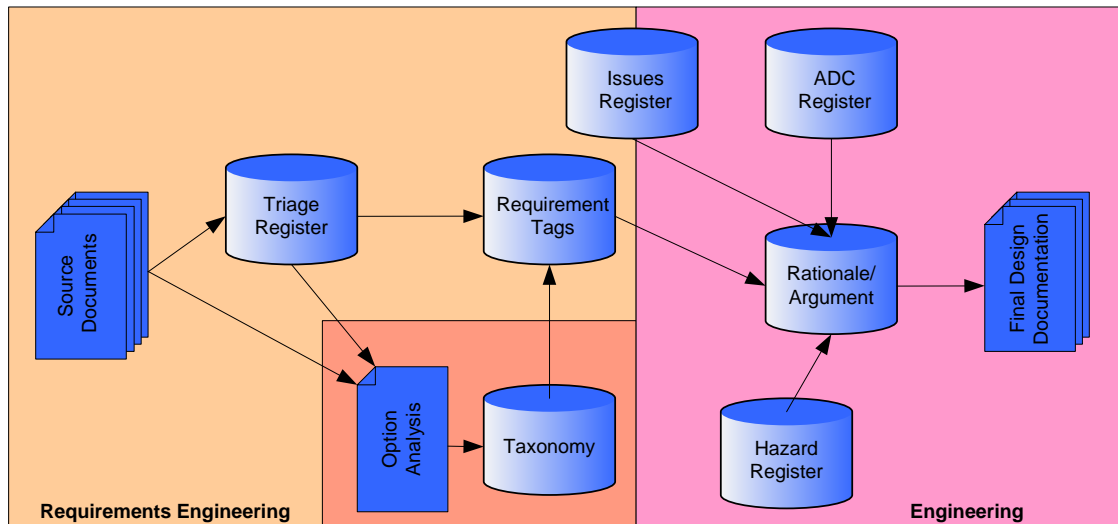
The amount of material and the detail provided will vary between projects, depending on project complexity, project novelty and, most importantly, the staff involved.

From analysis of past projects, we have identified four pillars on which a successful systematic approach is built. It is our belief that many projects fail because one or more of these pillars is either weak or flawed in some way. We have developed procedures, tools and templates that provide solid foundations in these four areas and expedite the mobilisation phase. Our four pillars are:

**Planning and Integration** – the Design Management Plan is the key planning tool, but to be effective it must be a ‘living’ document. In the projects we are presenting here, the DMP was used throughout the project as a reference for the team. We believe that one of the key drivers for this was that it combined processes, practical procedural instructions and project specific milestones that were actively measured by the project management team. This caused all members of the team to need to refer to it on a regular basis.

The DMP also forced an integrated approach between engineers, human factors specialists, operations and maintenance experts. This was achieved by insisting that there would be one plan listing all the key project phases and deliverables; all those involved were forced to use the same language and to explain their tasks in the context of this framework.

**Information Management** – the design management plan sets out an overall information model (see Figure 3), which is used to explain the relationship between all the information collected and produced. This is critical to enabling consistency of understanding within the team. The model is supported by a number of simple, effective procedures that are employed to manage all documents and information received and issued. It is also important that appropriate tools are used to support the management of information. The projects presented here successfully used Atkins iProNET, a web based collaborative working environment, to manage all documentation passed between parties; the tool supports configuration management and



**Figure 3 Example Project Information Model**

provides a single source of truth for project information.

We also used Telelogic's DOORS™ requirements management database to store all the requirements information and to link it with information and documentation generated by the project. DOORS™ has proven very effective in the capture and management of the link information, which is then employed as a basis for the assurance case for the final design. It has also delivered cost savings as a result of the speed and accuracy it affords the process of editing and reissuing documentation.

**Communication** – is an obvious requirement of any project, and yet not maintaining a consistent understanding of the current position of the whole project is a recurring cause of failure. Our approach to communications during the projects presented here was two fold.

Firstly, to establish and monitor clear project targets; this started with the overall deliverable milestones in the DMP and was supported by additional metrics and targets that were developed during the projects.

Metric collection and analysis were key drivers during a weekly telephone conference, which was structured to allow all team members an opportunity to describe progress made, what problems they had encountered and what they were planning to do next. This forum provided an ideal opportunity for team members to share ideas. This encouraged teamwork and helped to avoid the isolationist (silo) mentality that often leads to problems.

Secondly, the projects employed a number registers to formally track issues, technical queries, and ADCs (assumptions, dependencies and caveats). These simple tools were shared by all team members and reviewed regularly, providing formal impetus for individuals to communicate subjects of mutual interest.

**Risk Management** – we followed Yellow Book (UK Rail Industry, 2000) principles for managing both safety-related and project related risk, employing a simple hazard register in which we recorded both safety and non-safety hazards. The process is anchored with two formal design risk workshops, but hazards can be identified at any time in the project. Hazards are managed in DOORS in the same way as requirements, allowing us to demonstrate the design steps that have been taken to mitigate hazards, thus supporting the analysis of residual risk

### REQUIREMENTS ENGINEERING

The Requirements Engineering phase of any project must organise the morass of project requirements into an ordered set. Traditionally this phase takes a great deal of time and produces a set of well formed, atomic requirements – the foundation for the engineering to come. This work is often undertaken by specialists and frequently runs in parallel with early design work, justified either on the basis of programme, or because of attitudes such as 'this is the way it has always been done' and 'we know the solution the systems engineers will arrive at'. We have taken a different approach.

We consider it essential to engage all of the engineers in a process that allows them to begin to become familiar with the requirement source documentation – with all its foibles – and tease out the fundamental requirements that will drive the design. In the projects presented here, we encountered between 50 and 1500 requirement source documents, including standards, user requirements, operational concepts, operational procedures and interface specifications. We also needed to dissuade the team from jumping to solutions and at the same time identify where we needed to collect additional material; be that requirements or domain knowledge. To address these objectives we have developed a

process called **Triage**, and a concept we call a **Requirements Tag**.

### Triage

Triage is the name we have given to the process of initially reviewing the source material, classifying and prioritising it (the name derives from the medical procedure for the same purpose, albeit that the 'material' is human casualties!). This process is undertaken by the whole team, and we have found that it is best kicked-off in a workshop environment where all the team members are physically co-located. Undertaking this initial exercise together helps the team bond, ensures consistency and encourages cross-discipline questioning. There are two key pre-requisites for the triage process:

- The **scope** of the problem/design/brief has to be clearly stated and understood by all participants.
- A **system lifecycle** has to be agreed and all participants have to clearly understand where the scope is positioned on the lifecycle.

The Triage process asks five questions about each document (or triage item):

- Does it contain **Business Requirements** (viz. what the client wants the system to achieve) that relate to the defined scope – Answer Yes/No?
- Does it contain **System Requirements** (viz. what the system must actually do) that relate to the defined scope and lifecycle phase – Answer Lots/Some/None?
- Does it contain **Process Requirements** (viz. how the project team should deliver its remit) that relate to the defined scope and lifecycle phase – Answer Lots/Some/None?
- Does it contain **Domain Knowledge** (viz. relevant facts about the environment into which the system must integrate) that relate to the defined scope – Answer Yes/No?
- Does it contain requirements or domain knowledge that relate to a **later lifecycle phase** – Answer Yes/No?

This is all the information we collect during triage; be aware that there is a great temptation to collect more, but our experience has shown that additional information often proves worthless, as one does not really know what other questions to be asking at this initial stage.

This process requires that there is a strong document management process in place to register and track all incoming documents. All documents are subject to triage by at least one team member; we have employed a process of randomly peer reviewing work to ensure we maintain consistency and accuracy. The Triage process is relatively quick, as it does not require the reviewer to read and digest every document in detail; the objective is to understand the content just enough to be able to answer the five

triage questions. The process lends itself to the use of metrics, which can be generated relatively easily to track progress; our experience is that an individual can Triage between 5 and 20 documents a day.

The documents identified as containing business requirements are routed straight to the lead systems engineer/systems architect, as these will be the key drivers for the Option Development phase. Process requirements are passed to the Project Manager, as these are requirements of the process we adopt to develop the design. In some cases the material will raise questions or issues; these are recorded and where necessary investigations undertaken. When additional material is uncovered it is fed into the Triage Process. Importantly Triage avoids abortive time spent reviewing in detail irrelevant documentation.

### Requirement Tags

The concept of a requirements tag has evolved from the observation of common practice among engineers. When reading requirements source material we noticed that the most engineers employ a method of visually tagging important sections of the document; methods vary but include corner folding, highlighting and post-it notes. The problem with these methods is that they are typically not accessible to other team members. The concept of a **requirement tag** is a traceable, standard tag that is accessible to all and which conveys an agreed set of information about the material to which it relates.

An important aspect of the requirement tag is that its scope is left to the engineer to define, based on the material being tagged. That is to say that one requirement tag may relate to one specific, well formed, atomic requirement; another requirement tag may relate to a whole group of requirements. The former tag may point to one sentence, the latter to a whole document. The engineer is trusted to decide the level of granularity that is appropriate, in order for the requirement tag to sufficiently convey the totality of that to which it points. However, in order to achieve consistency between engineers, we have found it necessary to employ a template (or boilerplate), which mandates the structure of each tag.

Initially, the business requirements are tagged and classified using an appropriate taxonomy; again it is very tempting to launch into an exercise of system requirement tagging, which can show great progress in terms of measurable targets. However, our experience is that it is better to wait. In order to design an appropriate classification for the system requirement tags (and domain knowledge), it is necessary to have a clearly defined and stable model of the system under design. This will come from the Option Development phase, which develops the system architecture based largely on the business requirements. **Do not start tagging system requirements until an appropriate taxonomy can be agreed based on the evolving system architecture.**

The system requirement tagging process will require the engineers to re-visit material that has been triaged. This is a necessary and deliberate step in the approach. It may be that material is redistributed to a more appropriately skilled individual, but in many cases the same engineer who performed the triage will undertake the tagging. The approach forces engineers to become more familiar with the source material and helps to ensure that designing does not commence until all relevant information is known – this leads to improved quality of the resultant design.

In the case of the SCADA system design, initially it seemed that the Requirements Engineering phase would be straightforward, as the client had presented a very structured set of project requirements. However, our approach soon uncovered a host of additional and contradictory requirements, which were tagged and managed.

### **OPTION DEVELOPMENT**

The approach to this phase will vary significantly depending on the scope and lifecycle phase of the work in question. The objective of this phase is to identify solution options and identify the one that will best deliver the required business requirements. However, successful evaluation also requires the appropriate consideration of domain knowledge, key system requirements, human factors and safety.

The options developed at this stage are necessarily conceptual – they must be expected to evolve as more is understood about the requirements and the domain. The purpose is to develop an outline, which will guide and facilitate the design – a straw man. It is important not to get bogged down in detail at this stage.

Atkins Rail has employed this approach on a number of projects in the last two years, and each has taken a different approach to the option development phase. The VLU SCC project was guided by a very high level description of London Underground's operational concepts. We modelled these concepts using UML use-cases to provide a simple overview of the services required. This overview was validated via peer review and then used as the basis for a conceptual model of the SCC. Through a process of workshops with key experts, the conceptual model was used to develop a number of architecture options. We evaluated the optimum solution using a process of systems analysis; the analysis considered system safety, human factors and the business requirements.

In 2004/05 we developed a concept design for the Jubilee and Northern Line SCCs – a project that was very similar to the VLU project in terms of scope. However, the option development phase was very different. In this case a clear vision of the required systems architecture was already in place, and the dominant business requirement was the migration from the existing systems to the new systems. In this case, we created a suite of simplified schematics of the existing systems and developed these to establish solution options with a demonstrable migration path. Once again these were

evaluated to select the optimum solution.

Finally, in the development of a detailed design for a computer-based SCADA system, where we were introducing new technology, we augmented the systems analysis with trials to evaluate specific aspects of performance that were critical to the option selection.

In each case, as the solution options became clear, it was also possible to develop an appropriate model, which was used to design the requirement tag taxonomy. The process of revisiting the source requirements documents to tag system requirements and domain knowledge reinforces the design team's understanding and provides the basis upon which the assurance case was built.

The option development phase is formally completed with a design risk review. This process is designed to achieve compliance with the UK Railway Engineering Safety Management standard Yellow Book 3 (UK Rail Industry, 2000), employs a workshop to consider the selected design option and identify both safety and non-safety hazards that are associated with the design. The hazards are noted and, where appropriate mitigating actions taken in the Engineering phase.

### **ENGINEERING**

The Engineering phase is the engine room of the approach – this is the stage where the majority of the work is undertaken and when the main project deliverables are produced. The actual work undertaken is not significantly different to that which might be done if a more traditional approach were adopted. However, following a systematic methodology ensures that the design work is based on solid foundations.

The necessary engineering processes will be determined by the scope of the project; in recent projects this phase has included specialist engineering/design; human factors analysis and assessment; interface design; investigations and surveys; prototype development and evaluation; safety integrity level (SIL) assessment; migration planning; and reliability, availability and maintainability (RAM) analysis.

The objective of this phase is to develop the selected option which best satisfies the identified and tagged system requirements. The approach is structured to allow the design to be subdivided at this stage, based on the requirement tag taxonomy employed. Typically the team will review the totality of the requirement tags and domain knowledge that relate to a particular subdivision, and begin to develop the necessary design. Additional work, such as a RAM or SIL analysis, is undertaken where necessary and the results fed into the overall system requirements.

We used DOORS™ to provide formal configuration control of the design and, crucially, it allowed the design to be linked to the source requirement tags. The method we successfully adopted for linking employed an 'argument' clause,

<p><u>Consider the following Source Requirement Tags:</u></p> <ul style="list-style-type: none"> <li>• <b>Tag 1 (from a standard):</b> Communications shall be provided (audio and data) from all Train Control Centres to other Control Centres, and with emergency services and trains. Such communications shall also be recorded.</li> <li>• <b>Tag 2 (from client contract):</b> All incoming and outgoing telephone calls are recorded and the recording tapes are retained.</li> <li>• <b>Tag 3 (from User Requirement document):</b> Recording method must provide safe, secure protection for the recorded data with no ability to modify, delete, edit the material except under formal authorised methods.</li> <li>• <b>Tag 4 (from User Requirement document):</b> For post incident review and operational training purposes, all communications in and out of the Control Centre shall be recorded and the ability for selective playback provided. Conversation within the Control Room shall also be recorded.</li> <li>• <b>Assumption:</b> It is assumed that recorded material will be retained for a minimum of 28 days.</li> </ul>
<p><u>The following compliance argument can then be constructed:</u></p> <ul style="list-style-type: none"> <li>• “Facilities are to be provided to record all communications in and out the Control Centre control room. This will include conversations within the control room.</li> <li>• Assumption: It is assumed that recorded material will be retained for a minimum of 28 days.</li> <li>• This leads to Fully Compliant Requirements for an audio recording system with interfaces to Auto Telephone system, which can provide the following facilities: <ul style="list-style-type: none"> <li>- selective playback</li> <li>- authorised access only</li> <li>- no modification, editing or deletion possible”</li> </ul> </li> </ul>
<p><u>Subsystem Specification Clauses linked to this argument include:</u></p> <ul style="list-style-type: none"> <li>• “All voice/audio messages received shall be recorded by the audio recording service.”</li> <li>• “All auto telephone service calls into the Control Centre will be recorded by the audio recording service.”</li> <li>• “All conversations within the control room of the Control Centre shall be recorded via a system of microphones.”</li> <li>• “Recorded material shall be stored for a minimum of 28 days.”</li> <li>• “The recording method must provide safe, secure protection for the recorded data.”</li> </ul>

**Table 1 – Compliance Argument Example**

which provided rationale and aided understanding. As with the requirement tags we mandated that each argument follows a “boilerplate” to ensure that all the arguments were consistent in structure. Each argument began with a sentence that referred to the requirement tags which it supported – in some cases the requirement tag was self-explanatory and this initial sentence was straightforward; in other cases it was necessary to provide more details of how the requirement tags had been interpreted. The body of each argument explained any assumptions, dependencies or caveats that were implicit in the argument. Each argument stated if it was sufficient to provide full compliance, or if it was just partially compliant and why. Finally, the argument provided an outline of the elements of the design to which it referred, thereby providing a comprehensible link to the actual clauses in the design. An example of a requirement tag, argument clause and design clause is included in Table 1.

The development of argument clauses for each requirement tag involved significant effort, but we found it to be a very valuable exercise, which improved the quality of the engineering and provided a robust assurance case. Project team members took a number of approaches to the completion of design documentation and argument clauses; our experience suggests that the best approach is to first develop an outline of the design documentation based on a

review of the relevant requirement tags (system requirements and domain knowledge). The design documentation is then drafted, using expert knowledge. Once this draft is complete the team return to the argument clauses and complete them as appropriate. A separate exercise is then undertaken to review all of the requirement tags relevant to each design subdivision to ensure that they are all appropriately considered in the design documentation. This approach produced surprises for a number of experienced engineers who discovered aspects of the requirements that they had overlooked. Although it is not possible to quantify the full benefit associated with the identification of these omissions so early in the design, it is clear evidence of the advantages of a systematic approach.

The quality of the assurance case is dependant on the time spent in developing the argument clauses. Similarly, it has been shown that the quality of the design also improves with effort spent on the argument clauses. However, in our view the limit of this latter relationship is reached well before the former. The team must estimate the value of improving the assurance case, beyond the point where significant design improvements are available, in order to decide how much effort to expend on the argument clauses. In the case of the projects we are presenting here, it is unlikely that the assurance case will be used beyond the Approval in Principle project

stage gate, therefore the decision was taken to allow multiple argument clauses to link to each design clause and to each requirement tag. In some cases this makes the reading of the assurance case complex as it comprises multiple arguments. However, this was the most expedient route given that we had teams working in parallel on different design subdivisions, which in some cases had common requirement tags. Overall, in our assessment, we feel that the small increased effort involved in reading the assurance case is warranted given the time saved in writing it.

A more straightforward assurance case could be generated if it were mandated that each requirement tag was only to be supported by one argument clause. An argument clause would still be able to support more than one requirement clause and could refer to multiple design clauses. This rule would force greater discipline in preparing the argument clauses, but would also add to the time taken to develop the design, as it would necessitate closer co-ordination between team members.

The Engineering phase is concluded with a final design review. This design review considers the overall design that has evolved from the design option; it also reviews the hazard mitigation included within the design, defines and qualifies the residual risk and provides an opportunity to identify any additional hazards.

### COMPLETION

The completion phase of any project will be concerned with the production, review and approval of the final client deliverables. This is a key phase that is often compressed in terms of programme, as earlier phases slip. It is our experience that any short term cuts at this stage generally lead to longer term disbenefits. It is necessary to plan for proper drafting and review processes, providing ample time for comments to be collated and considered. It is also vital to have a formal approach to capturing comments and the actions taken to address them. As with much of our approach, this is a straightforward exercise, but it is surprising how often it is deemed expendable. Curtailing time and effort spent on this phase invariably reduces the client's 'buy-in' to the content of the deliverables.

The projects we have undertaken recently, specifically the VLU SCC project, also demonstrated very clearly the benefits of having the source material held in a strongly configuration managed environment such as DOORS™. When it came to updating and manipulating the content, we found that the speed and the accuracy with which changes could be made was significantly improved over traditional word processor based editing. The VLU SCC project completion phase produced eight system specifications, containing over 2700 system requirements in total. These were supported by an overarching concept design report and the assurance case, which consisted of 353 compliance arguments, 50 ADCs and 1781 source requirement tags. These deliverables were produced by a team of engineers

working in parallel from different offices. DOORS™ enabled all of this documentation to be collated, updated following comments, and re-issued easily in a matter of days.

### CONCLUSIONS

The paper describes an approach that has been developed over the last 3 years within Atkins Rail, based on the practical experience of a number of systems practitioners and building on the knowledge gained from using the approach on real projects. The VLU SCC project is a very large, multi-disciplined project with very tight timescales and complex requirements. The techniques employed were largely successful and, while the overall VLU SCC project is still in its infancy, the feedback Atkins Rail has received about the Concept Design work undertaken has been very positive. The SCC Concept Design was successfully delivered to Metronet, who are in the process of moving to the next lifecycle stage of the project.

The VLU SCC project was built on the successful use of requirement tagging for the Tube Lines SCC Communications Systems Concept Design project. The VLU SCC project exploited DOORS™ to a much greater extent and derived significant benefits from doing so. We also employed templates (or boilerplates) to great effect and this improved the consistency across the assurance case greatly.

The most significant learning gained from the VLU SCC project, which was applied to the SCADA design project, was the need for triage teams to be initially co-located and the dangers of compressing the completion stage. SCADA design project demonstrated the speed and flexibility the approach affords, as it allowed a team of six engineers to complete the mobilisation, requirements engineering and option development stage in ten weeks. In this time, 75 documents were triaged, 1132 requirements tags were generated, and a 100 page specification was generated (from DOORS) that presented 1200 consolidated system requirements in a structure based on the architecture option selected. This specification was supported by an assurance case, founded on the argument clauses which were linked back to the source requirement tags.

The requirements triage and tagging approach that we have developed is novel and we have found it effective in two key areas. We believe that it engages the whole team in the exercise of requirements engineering, without overburdening individuals with Systems Engineering theory or terminology. This leads to a more integrated team, delays the onset of engineering and results in a better end product. Secondly, the requirements triage and tagging approach is quick. We recognise that this speed is at the expense of the elegance of the resulting compliance argument and assurance case, but in our view it is likely this will be acceptable in most railway scenarios.

The authors believe that the combination of

rigorous application of basic principles and our requirements triage and tagging approach, renders Systems Engineering squarely within the reach of all railway projects and all engineers. We have seen how it empowers teams with greater knowledge and understanding on the problems they face and how motivating it is when they recognise the scale of the improvement that such simple steps can deliver.

We plan to continue to develop our approach as we are faced with new projects and new challenges, but we are confident that we have a simple and accessible basis on which to propagate the use of Systems Engineering in the rail industry.

of system changes to Axle Counter systems on the UK's West Coast Main Line. He has delivered presentations to the IRSE (Institution of Railway Signal Engineers) & INCOSE (International Council on Systems Engineering) in England, Europe & the U.S.

### REFERENCES

- London Underground Limited Chief Engineer's Directorate, *New or altered assets – approvals prior to bringing use*, E1008 A4 Cat 1 Standard, Oct 2002
- UK Rail Industry, *Engineering Safety Management Yellow Book 3*, 2000

### BIOGRAPHY

#### **Dr Jon Elphick BEng DPhil CEng MIET**

Jon works with many of Atkins key clients providing engineering expertise and strategic advice. Within the last 4 years he has acted as lead systems engineer and systems architect within projects for Network Rail, BAA, Metronet and Tube Lines; he has undertaken this role in the three projects referred to in this paper. Prior to this, he has worked for MetalBox, Westland Helicopters and Procter and Gamble, and as a project manager delivering industrial IT and process control systems. Jon specialises in the implementation of Systems Engineering in the rail industry, and delivering business benefit from the accelerating convergence between the IT and engineering.

#### **Mark Irving BEng (Hons) CEng MIET AMIRSE**

Mark is a Railway Signalling & Systems Engineer. Since completing a Graduate Training Scheme with British Rail's Signalling & Telecommunications department in 1995, he has worked as a Signalling Systems Engineer on mainline & Light Rail projects. He initially specified, designed, tested & commissioned Electronic Signalling Control Systems on UK heavy rail infrastructure. He has been involved in the transfer & development of German/Swiss train borne & trackside signalling technology into the UK environment. Recently, he has undertaken Requirements Management activities on mainline & underground railway infrastructure projects. He spent 18 months project managing the delivery of Safety Case activities associated with the introduction