SYSTEMS INTEGRATION BEGINS AT THE END

This article explores the development of system-level requirements to bring about desired rail program outcomes.

The defining characteristic of the systems integration (SI) approach is to begin with the end in mind. This core feature[1] means understanding what the operational outcome of a project should be and what benefits the users and stakeholders of the system desire. It is often easy to articulate the top-level goals of a project, whether it is a railway, a building or a product. However, in most cases it is impossible to deliver a solution based only on stakeholder expectations and user needs. So, the expectations and needs must be broken down into pieces that form deliverable packages of work. Applying a systems methodology to the breakdown, structure and management of these expectations and needs—in the form of requirements—can provide the difference between delivering on time and on budget, or not delivering at all.

Encountered Issues

A common approach to breaking down a program into deliverable work packages is to create a concept design followed by a plan for delivery (by describing the packages of work and when they need to be delivered), and then move straight to defining the contract requirements. This process often leads to encountering one or more of the following five issues:

1. **Over specifying or underspecifying what the subsystem or contract needs to deliver –** Little traceability from the contracted requirements up to the user needs means it is often hard to find out how a contract-level requirement contributes to the operational outcome.

2. **Subsystems or contracts that do not integrate to meet the overall operational outcomes –** This means diminished benefits for the end user or additional time needed to align the delivered products to make them work together.

3. **Sections of work are missed or have overlapping boundaries –** The concept design may not be complete, or the way the subsystems or contracts have been defined means there are gaps in the overall system. Conversely, two separate subsystems may assume responsibility for the same aspect of delivery.

4. **Missed opportunities for innovation –** When defining requirements at the contract level, the main deliverables of the system have already been established; this means lost opportunities for the contractor to meet user needs in an innovative or a new way.

5. **Contradictions and confusion in requirements for the work package –** If all the system requirements are defined at the contract level, this can lead to specifying needs that are not clear or too generic so that the contractor cannot fully meet them.

[1] This core feature was introduced in the article "Requirements Done Right" by Nassar Majothi.

Starting Right With System-Level Requirements

Within the systems engineering and systems integration discipline, requirements engineering (also known as requirements management) aims to avoid these pitfalls by developing upfront a set of clear, complete and correct system-level requirements and providing them to those contracted to deliver the work packages (or subsystems). The intention is to ensure that when delivered those packages come together to deliver the desired operational outcomes.

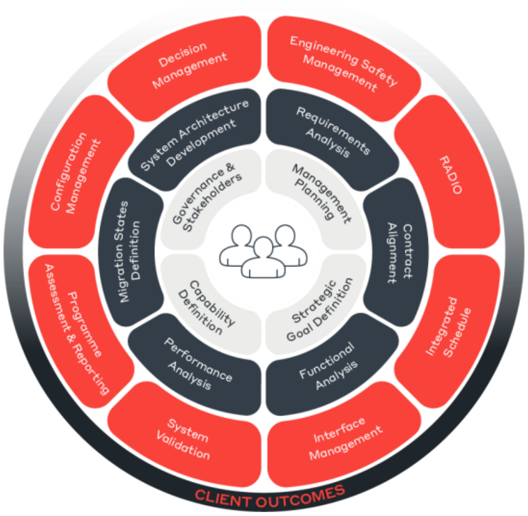

Figure 1 - WSP's SI:D^3 framework brings together a set of processes used to manage the development of a complete set of specifications to deliver the unique strategic goals of each program.

Developing the requirements is done by understanding the capabilities (functions and performance) that represent stakeholder expectations and user needs and determining how the whole system will meet those expectations and needs. Then, an assessment is made to specify the work packages (or subsystems), what those work packages are required to achieve and how those packages must integrate to deliver the capabilities using SI. This holistic process culminates in defining the requirements at the contract level. Each level described here may have multiple sublevels.

A simple example is shown below in Figure 2 and explained below.

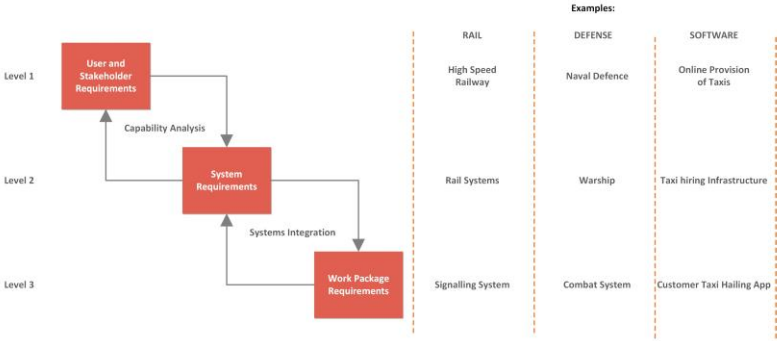

Figure 2 - Requirements Structure

At the top level, the user and stakeholder requirements are captured. These are written to reflect the desired operational outcomes, user benefits and, where appropriate, constraints. The intention of these requirements is to set the scope of what is to be delivered. It is key to avoid too much detail at this level, as at this point the system should be considered holistically.

After capturing and analyzing the user and stakeholder requirements, a set of system requirements is established at the system level. One way of doing this is to create operational concepts outlining, at a high level, how the system is expected to behave to meet the user and stakeholder requirements. These concepts describe the functions that the system must perform and how the system must perform them—collectively referred to as the capability of the system. The system-level requirements are linked to the user requirements, showing how each of the system functions will deliver the desired operational outcomes. As the definition of the system becomes clear, the user requirements may be updated based on the system-level requirements. Updating can happen for a variety of reasons and is an expected iterative step as the system definition explores the realities of achieving what the user wants within the program constraints.

Once the system-level requirements are understood, the subsystems/work packages can be defined. Defining subsystems can involve the creation of a concept design and the use of a work breakdown structure to govern how to divide the work. The work packages will still likely be distributed primarily by technical areas (such as civil engineering), geography or phases of delivery; however, the division is driven from the system requirements by considering what work packages are required to meet which capabilities and, crucially, how those work packages will interact to deliver the operational outcomes. While the work is being apportioned, how the work packages are going to integrate is captured in the form of interface definitions (e.g. interface control documents). This is key because by definition a system provides more capability than the sum of its parts, and understanding how the work packages will come together before defining them in detail will save significant integration cost in the long run.

Once the work packages or subsystems have been determined, the contract requirements are derived and decomposed from the system-level requirements that are assigned to each contract, in line with the agreed-upon interfaces. This process ensures that each work package will meet the needs of the system (and, through traceability, the needs of the users and the expectations of the stakeholders) and integrate with other work packages. The requirements created in this way are then traced back to the system requirements, which might bring change at the system level—expected, as the detailed understanding of the work package will inform the system-level design.

In a perfect world, every work package requirement would be derived from the system level. In reality, there will be some gaps and outstanding requirements for the work packages/ subsystems that should be developed by domain experts. At this level, the flowed-down requirements may be complemented by the development of additional requirements to reflect the specific functions of the subsystem. This is done by utilizing expertise, modelling and design.

Delivering Desired Outcomes

By adopting the systems integration process and dedicating time to develop each level of requirements, a clear path from user needs and operational outcomes to the work package requirements is created. This approach helps ensure a clear, complete and correct set of requirements describing what is truly needed and confirms that the separate work packages will integrate correctly. Crucially, it puts in place from the start a structure to ensure that the contracts deliver what stakeholders want and users need.

Authors

James Spink

Principal Engineer, Systems Engineering & Integration,
United Kingdom

James.Spink@wsp.com

Genevieve Edwards

Associate, Systems Engineering & Integration, Australia

Genevieve.Edwards@wsp.com

About WSP

WSP is one of the world's leading professional services consulting firms. We are dedicated to our local communities and propelled by international brainpower. We are technical experts and strategic advisors including engineers, technicians, scientists, architects, planners, surveyors and environmental specialists, as well as other design, program and construction management professionals. We design lasting solutions in the Transportation & Infrastructure, Property & Buildings, Environment, Power & Energy, Resources and Industry sectors, as well as offering strategic advisory services. Our talented people around the globe engineer projects that will help societies grow for lifetimes to come. wsp.com