The Complexity Gap

Managing Complexity in Rail Supply using Proven Systems Engineering Methods

Even before the COVID-19 crisis, the next few years looked like a period of significant change for the rail industry.

The network is being transformed by massive investment in the digital railway, and technologies like cab signalling and condition-based maintenance require IT integration between systems that never used to directly interact.

In the longer run, greater central control of the network and a set of common strategic priorities set by an independent guiding mind will create pressure for the different components of the network, and suppliers to it, to work together better. Opportunities from decarbonisation, accessibility, Augmented Reality, automation and internet of things technologies only make the technical complexity of supplying to the rail industry seem more formidable on longer time horizons.

Rail systems, in short, from the remotest level-crossing to the network as a whole, are getting much more complicated. Every little piece of infrastructure, every part of a train, increasingly needs to be looked upon as a detail in a big picture rather than as something more discrete, with its own maintenance schedule and separately identifiable requirements.

When industries turn 'smart', when that threshold of technical complexity is passed, it usually requires of engineering to radically rethink its workflow and practices. When it happened to software engineering in the 1990s, Agile was the result. Rail is getting to a similar position now, and the time has come for it to explore options for how it can adapt to this changing environment.

Agile, for all the buzz, is unlikely to translate well to rail systems in most cases. Agile was designed for products with frequent update potential, low integration costs, little dependence on specialised hardware, and with minimal regulatory constraints. In other words, there is little risk associated with getting it wrong the first time.

In rail, the whole thing had better work together first time, not just as an independent system, but in the context of the complex system-of-systems that is the network around it. If there is a defect in the product arising from an unexpected feedback interaction between apparently independent components, it could at best require you to go all the way back to the drawing board, and at worst result in a deadly disaster.

Software engineering wasn't the first industry to reach a complexity threshold which required them to throw out the engineering management playbook; that honour goes to aerospace and defence, which began overhauling its methods in the Second World War, and has continued to develop these engineering management techniques – which came to be known as systems engineering – through the space programme and into the most complex military systems of today.

Rail supply needs an Agile-like revolution in the way it manages engineering complexity, but Agile itself will only work in very specific and limited contexts. Systems engineering has already made considerable headway in the rail industry as an alternative way forward, starting with major infrastructure owners and projects like Network Rail, TfL, Crossrail and HS2. But it is yet to reach all parts of the industry, and as growing rail network complexity reaches further down the supply chain, the argument to adopt systems engineering techniques more broadly only grows stronger.

Processes & Tools

Systems engineering techniques are fundamentally about finding ways to analyse, model and plan the behaviour of a system as a whole and in its context, above and beyond the details of individual components. By having a suite of processes and tools designed to model and anticipate the structure of a system, organisations can have assurance from the start that the right thing is being built in the right way, and that their product will interact appropriately with its context. This drives down cost by reducing the risk of mistakes and unanticipated defects, while simultaneously driving up quality by tying engineering activity more closely to precisely defined stakeholder needs.

Systems Engineering (SE) has developed a wide range of processes and tools for modelling and simulation, requirements analysis, scheduling, and all parts of the life cycle, tailored to better manage the development of complex systems. Of particular interest is how SE thinking has produced a robust and scientific approach to requirements management and verification, a greater focus on the full life cycle of a product, and novel modelling techniques for complex emergent behaviour.

An SE approach to requirements is designed to cleanly and specifically identify ambiguities and gaps in stakeholder needs. Systems engineering treats the requirements engineering process like formulating a scientific hypothesis. If an SE process for generating requirements is followed, it is immediately and specifically visible when individual requirements are not clear, verifiable, functional or minimal, as well as when they are together incomplete or inconsistent. As a result, the question that needs to be answered either by stakeholders or by engineering is specified precisely and robustly.

The context and environment for the system – its basic inputs and outputs – should be understood as clearly as possible while the system as a whole is still being treated as a black box. In a systems engineering process, only then do engineers start to formally investigate the sorts of systems which could solve the stakeholders' problem.

As well as the direct benefits of specifying project objectives with such a high degree of precision, this approach to requirements also enables systems engineers to construct sophisticated models of products, which can anticipate many potential problems before committing to development costs. These models touch on every aspect of the life cycle and are designed to predict the behaviour of a system taken as a whole.

The most fundamental insight of systems engineering models is that by analysing the structure of a system, you can understand aspects of its behaviour that would be missed by focusing only on the individual system elements.

These elements are organised into systems, and then even into a 'system-of-systems', which is a model for systems with very independent components and a function that firmly rests on emergent behaviour, like a railway network or a supply chain.

Using a model like this enables a systems engineer to focus on complex interactions within a system and between the system and its environment, including patterns and trends in how the system changes over time, the impact of time delays in the system's operation, the circular nature of complex cause-and-effect relationships, the problem of where unintended consequences are going to emerge, and the ability of a system to address stakeholder needs.

Systems engineering also provides a much more robust process for integration, verification and validation. By using a scientific approach to requirements management with specifically enumerated constraints on what requirements can look like both individually and as a set, the systems engineering process ensures that verification and validation are conducted in relation to specific, measurable and consistent goals.

Finally, by taking a whole life cycle approach, systems engineering can help to ensure the success of midlife upgrades, prevent the loss of system capabilities during operation and avoid costly compliance failures and other losses during end-of-life disposal.

Systems engineering, in the rail industry and elsewhere, has a proven track record of delivering increased efficiency, capability and adaptability in the development of complex systems. Now could be the moment for far more of the rail industry to adopt the technique.