Part Four: Getting it Right Every Time

Quality isn't an afterthought; it has to be a constant focus through every stage of the development process. How many times have we heard that? Six Sigma and Lean methodologies have brought a ubiquitous 'Kaizen' philosophy of continuous improvement to development and manufacturing, and Agile project management emphasises iterative testing and incremental development.

Systems engineering has learned a lot from these methodologies and is continually adapting its processes to add value alongside these disciplines.

Using these techniques can only get you so far, and most of these ideas originated in specialised development disciplines. Lean and Six Sigma methodologies add the most value in bulk manufacturing processes; Agile works best with adaptable products like software, with new versions rolling out after weeks rather than years.

Often, complex systems are produced in smaller quantities and can't be changed so easily.

Lean, Six Sigma and Agile methodologies can add a lot of value when you're building the engine or coding a module; not so much when you're building the engine factory or setting long-term strategy for a huge software platform. And then there's all the space in between.

What systems engineering brings to the table is a view of the system as a whole and how its structure generates its behaviour, and this extends to its approach to quality. The outputs of the whole system, in terms of user satisfaction, its operating environment and its interaction with other systems – in short, the value derived from the whole system throughout its life cycle – are at the front-and-centre of quality.

Thinking about quality like a systems engineer is about thinking in terms of a hierarchy of complexity. When designing the system, we start with the broad needs of the client, turn that into specific requirements for the system as a whole, create an architecture at the system and then the subsystem level, and only then produce a detailed design for the individual elements. Ensuring quality of a whole system is about going through that hierarchy in reverse: testing the reliability of individual components or modules against specifications; verifying the performance of subsystems against requirements; then validating the outputs of the system in terms of customer need. This is coupled with a clear recursive process for when standards are not met, to ensure definitions are revisited at the most specific level possible.

Systems engineers work to ensure processes that touch any aspect of the life cycle are capable of meeting the requirements of the project. Because systems engineering success involves establishing scientific and falsifiable requirements, it is also involved with setting the measurement and quality assurance processes. Its focus, therefore, is on ensuring an unbiased and repeatable process, with independence in assurance, and straightforward, repeatable measurement that can be performed frequently.

Verification and validation are the processes of performing these assessments on the project. Rather than see the difference as being the point in the life cycle when these actions are performed – before and after product integration – systems engineering sees them as respectively ensuring that the product has been built right and that the right product has been built. Verification is about checking the system against requirements, architecture and design; validation is about checking it meets client needs. Depending on the system, both of these tasks might need to be performed at any or all of the element, subsystem or system level.

The point of using systems engineering processes in quality is about looking at the system-as-a-whole in the right way. We believe this approach is complementary to a wide range of development methodologies.

A huge part of that is in its scientific approach to requirements engineering, which generates specific, unambiguous and testable requirements using the same method as a scientist uses to generate the hypothesis of an experiment. Black box models of how the elements of a system interact with one another also significantly reduce the risk of an emergent defect not being detected until the system has been integrated. But those are subjects for other articles: articles three and five in this series, in fact.

To discuss how your organisation may use Systems Engineering to accelerate projects, improve quality and reduce costs, contact us via: cet@synthesys.co.uk or call us on: +44(0)1947 821464.

TECHNICAL ARTICLE

About SyntheSys

SyntheSys provides defence systems, training, systems and software engineering and technical management services over a spectrum of different industry sectors. Along with distinct support and consultancy services, our innovative product range makes us first choice provider for both large and small organisations. Established in 1988, the company focus is on fusing technical expertise with intuitive software applications to solve common industry challenges.