



Insights

Systems Engineering 101

What is systems engineering and how can it reduce the risk of project failure?

In 1984, Michael Griffin was a young systems engineer at NASA. While Griffin would go on to become the US space agency's 11th Administrator some 20 years later, his first job at NASA would see him play a pivotal role in the successful deployment of the Hubble Space Telescope.

Part of the Hubble development team, Griffin was tasked with designing a backup guidance sensor for the revolutionary space telescope. As part of the job, Griffin was instructed to perform an independent systems engineering assessment of the entire telescope, where he subsequently found critical interfacing problems existing between the Hubble's various subsystems.

Although each of the contractors involved in the telescope's design and construction could demonstrate that they had fully met the requirements on their side of the interface control

document, it was obvious – even to a young and relatively inexperienced systems engineer like Griffin – that the Hubble would not function properly when launched.

Ultimately, the interface issues were corrected, and the Hubble was launched in 1990. However, if systems engineering had not been a fundamental element of NASA's development approach, it's unlikely the integration problems would have been identified in time, and the infamously troubled telescope may well have ended up being a total failure.

What is Systems Engineering?

While not everyone gets to work on big ticket projects at NASA, if you've ever been involved in a complex project or product development effort, you will know how challenging it can be. From managing stakeholders to designing and testing

components, there are countless variables to consider. That's where systems engineering comes in. In this article, we'll explore what systems engineering is, how it works and how it can help reduce technical and commercial risks in any project, from complex technology developments to mega projects.

So, what is systems engineering? Simply put, it is an interdisciplinary approach to designing and managing complex systems. It is a structured process that focuses on defining and balancing requirements, managing interfaces and verifying and validating systems throughout the system lifecycle. Systems engineers work closely with stakeholders to identify and prioritize needs and then develop and implement solutions that meet those needs.

How is Systems Engineering Performed?

Systems engineering is performed through a series of processes and activities that help ensure the system meets the desired requirements. These processes and activities include:

Concept of Operations (ConOps): an early-stage process that defines the overall objectives, scope and operational concept for the system. It helps to establish the vision for the system and ensures that all stakeholders have a clear understanding of the system's purpose and intended use.

Mission Definition: The process of translating the ConOps into a set of high-level requirements that the system must meet. This process defines the key performance parameters and identifies the constraints that will influence the design of the system.

Stakeholder Requirement Elicitation: This process involves gathering and analysing project requirements from all stakeholders. The role of a systems engineer is to discern each stakeholder's needs and collaborate with the project team to determine which are reasonable, achievable and necessary.

If a requirement can't be met, then a well-supported rationale is needed to satisfy the stakeholders. This is a fluid process, and it is

inevitable that new requirements will materialize at different stages of the project. However, it's highly cost effective and efficient to understand project requirements as early as possible to avoid redesigns and delays.

A simplified comparison of Stakeholder Requirement Elicitation is like cooking a very large dinner for many guests. It is key to understanding everyone's meal preferences and is crucial to know dietary preferences before cooking. However, not everyone may agree on the menu, some might want a vegan meal, others might want steak. Adding or changing a dish in the middle of cooking could throw off your entire dinner plans.

It is therefore important to figure out a menu that satisfies everyone and be able to explain why certain dishes can't be made. This ensures all dinner guests are satisfied with the outcome.

Common areas of friction between stakeholders where good requirement elicitation is key are:

1. Government Agencies vs Environmental Groups: Project completion versus environmental impact.
2. Clients vs Contractors: Conflicts occur regarding project cost, timeline and quality.
3. Regulatory Bodies vs Engineering Teams: Disagreements stem from compliance with standards versus innovative and cost-efficient solutions.
4. End Users vs Project Planners: Tensions over specific project needs versus planned outcomes.
5. Local Communities vs Project Sponsors: Friction often occurs due to concerns about project disruptions versus project goals.

System Lifecycle: This is the definition and management of a system from cradle to grave (conception to retirement). The lifecycle ensures a system's effectiveness, efficiency and adaptability throughout its existence and gives structure to the projects milestones giving points for stakeholder reviews to occur to ensure that the project is ready to move to the next stage.

Investing on the left (early) stages of the lifecycle, ensuring the system is well defined and understood will save large amounts of time and money minimising redesigns and operational costs. A common lifecycle methodology used in systems engineering is the V-Model. An example of these lifecycle stages is illustrated below.

System Architecture: In systems engineering, this is the process of defining a system in terms of its various components, their interactions and the principles that guide its evolution and development. The task of creating systems architectures is a critical part of the systems engineering process and it is covered by international standards like IEC 15288 and guidelines from organizations such as the International Council on Systems Engineering (INCOSE).

Systems architecture in engineering comprises three types: Physical, Logical and Process architecture.

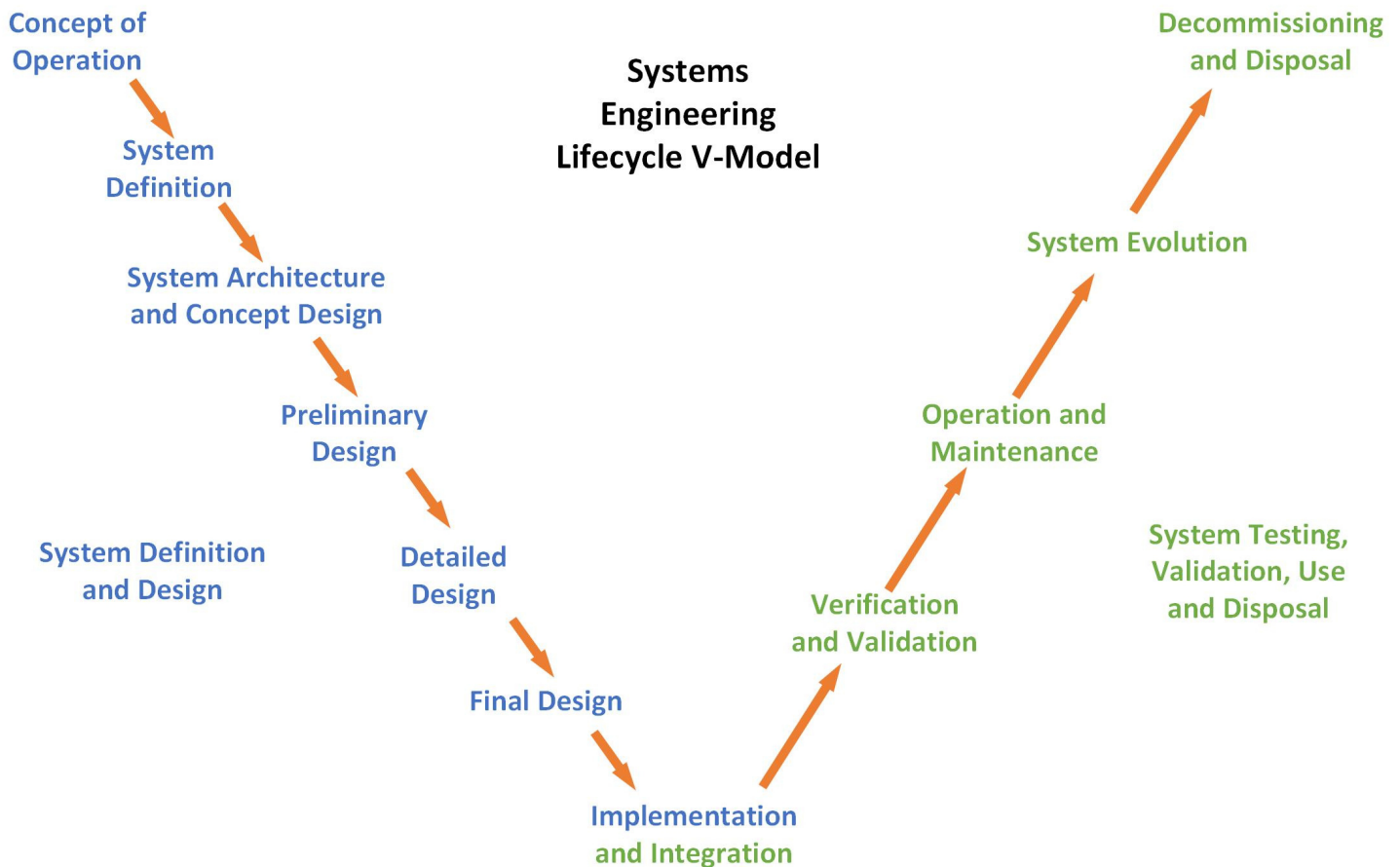
Physical architecture outlines the system's tangible elements and their interconnections. Logical architecture maps system functionalities and their interactions.

Process architecture details the system's dynamic activities, their sequence and control flow. All three provide different perspectives and contribute to a comprehensive understanding of a system's architecture.

System Requirement Management: This process involves the allocation, tracking and control of requirements using the system architecture throughout the system lifecycle previously defined. It includes identifying where requirements are applicable, changes to requirements, managing the impact of changes and ensuring that all changes or non-conformances are properly documented and communicated to stakeholders.

Usually, a requirements management tool such as IBM DOORS or JAMA software are commonly used

Fig. 1: Example of a systems engineering lifecycle V Model





The total failure of Denver International Airport's US\$460m baggage system demonstrates the potential consequences of inadequate SE investment in complex projects.

examples, these will be used to manage a large, interconnected database of requirements throughout each phase of a project.

A requirement can only be closed once suitable verification and validation evidence has been supplied and linked to the requirement. A design should have all its requirements closed before moving to the 'use' phase of its life.

Interface Management: Identifying and managing the interfaces between system components. This process includes defining the interface requirements, managing changes to the interfaces and ensuring that the interfaces are properly tested and validated throughout the project.

As outlined in the introduction, the Hubble Space Telescope incident highlights the value of good interface management along with the potential costs of getting it wrong – or just not doing it at all.

Configuration Management: The process of managing the configuration of the system throughout its lifecycle. This process includes identifying and controlling changes to the system configuration (change management), maintaining the configuration baseline (a reference point in the

development process of a system that has been reviewed and agreed upon, serving as a base for future development and changes) and ensuring that all changes are properly documented and communicated to stakeholders.

Verification and Validation: Testing the system to ensure that it meets all requirements. This process includes developing and executing test plans, tracking and linking test evidence, verifying that the system components work together as intended and validating that the system meets stakeholders' expectations and needs.

What Happens When a Project Goes Wrong?

Even with the best planning and execution, things can still go wrong in a complex project. When this happens, it can be costly in terms of both time and money. Systems engineering can help mitigate these risks by providing active communication and scope control throughout the project.

For example, the Denver International Airport's Baggage Handling System failure was meant to be a revolutionary automatic baggage system that would greatly increase baggage movement efficiency around the new large airport. However, the system cost over double the initially budgeted US\$230m, caused the delay of the opening of the

Denver airport by 16 months, never met its original requirements and was eventually replaced by a manual system.

Some of the failures caused by lack of systems engineering were:

- The system had more than 100 individual PCs that were networked together. Failure of any one of the PCs could result in an outage as there was no automatic backup.
- The distributed nature of the design (with PCs dotted around the different concourses) added to the difficulty of resolving problems when they arose.
- The system was unable to detect jams and as a result when a jam occurred, the system simply kept piling up more and more bags making the jam that much worse.
- During 2005 prior to shutdown maintenance costs were running at \$1M per month.

Poor systems engineering played a large part to this project's failure. The project lacked clear requirements, comprehensive testing and effective integration of system components.

What Can Systems Engineering do to Reduce Technical & Commercial Risk?

Utilising the systems engineering approach to a project following the methodology previously described can achieve the following:

Early identification of issues: By having a well-designed system in place, issues can be identified and addressed before they become major problems. It is often viewed that systems engineering is an unnecessary cost that often is not used properly in projects.

The reality is, by embracing system engineering, initial costs of system engineers will be more than covered by the cost savings to the project saving time and money in the long run. When performed well, the use of verification and validation testing can identify any issues a product may have and allow them to be rectified before it is released for

service, so minimizing risk and downtime during operation while also reducing maintenance costs.

Mitigate risks: Systems engineering processes can help identify and mitigate risks before they become major problems. This can include identifying potential failure modes and developing contingency plans to address them. Again, this will minimise costs and downtime of the project.

Improve communication: Systems engineering processes can help improve communication between different teams and stakeholders. This can help ensure that everyone is on the same page and working towards the same goals avoiding project change and redesign.

Good communication in areas such as interface management is vital. Maintaining functioning interfaces between different subsystems by ensuring design teams understand the different subsystems functions and requirements and communicate changes avoids potential system failures.

Control scope: Systems engineering processes can help control the scope of a project, ensuring that it stays within the bounds of what is feasible and achievable. This can help prevent projects from becoming too complex and unmanageable.

An Example of Effective Systems Engineering

The International Space Station (ISS) program is one of the most successful examples of Systems Engineering.

The program involved numerous stakeholders, including the U.S., Russia, Canada, Europe and Japan and required the integration of multiple modules and systems. Through a well-defined Systems Engineering process, the program was able to overcome numerous challenges and has been in operation for over 20 years.

Conclusion

In a world with ever increasing complexity and automation, the significance and demand for systems engineering cannot be overstated. Systems Engineering facilitates a structured, holistic approach to tackling highly complex



"The International Space Station (ISS) program is one of the most successful examples of systems engineering."

problems, mitigating risk and optimizing efficiency. It encompasses intricate webs of interconnected components, integrating them into a functional whole, while adapting for changes in the environment while accommodating technological advancements such as Artificial Intelligence (AI), Digital Twins and the Internet of Things (IoT).

An example of a technology that we all depend on daily which heavily relies on system engineering is mobile phones. On one side there is the highly capable multifunctional device that can make phone calls, browse the internet, use social media, play music, record videos and play games that requires components and technologies to work together in a system that fits in our hand at a low cost. On the other side, complex technologies such as satellites and the immense mobile phone network allows us to use our devices and have connectivity, data and information all over the country where you can utilise the internet, digital maps and a plethora of different APPs.

Systems engineering brings coherence to chaos, enabling us to navigate and manage the labyrinth of sophisticated systems that permeate our everyday lives with ever-evolving technologies

such as traffic management systems (air and ground) and vehicle automation. Without systems engineering, self-driving cars would be less feasible in the future as they would take longer to create, cost more money due to many iterations of redesign before a functional system would occur and would also be prone to safety issues due to things like design integration issues and life-threatening bugs in code that are present due to lack of verification and understanding of the systems requirements.

Systems engineering can reduce technical and commercial risk, ensuring a successful project while saving time and money. Without system engineering, we would be ill-equipped to understand, let alone handle, the multi-faceted challenges presented by our increasingly complex world.

Matthew Manger | Systems Engineer



**Delivering trusted expertise
to highly regulated
industries**



CONTACT US

+61 (0) 478 814 324
enquiries@acmena.com.au
www.acmena.com.au

Acmena Group Pty Ltd
PO BOX 220
Ashgrove West
Brisbane, QLD 4060
ABN: 37 158 514955
ACN: 158 514 955