Considerations for the Development of a Pragmatic Systems Engineering Process

Tri Gaffney
Pratt & Miller Engineering
New Hudson, MI

ABSTRACT

The term "Systems Engineering" encompasses a large number of 'engineering' tools and processes that all can provide benefit to a program, if used properly at the right time. The objective of this paper is to describe how to navigate the elements of designing the various Systems Engineering tools and process for the scope of the project. Some organizations/individuals can over-use systems engineering tools, to the detriment of project overhead; while others under use the tools at the expense of project quality. There are a few basic tools that can help to justify the magnitude and use of the project.

Introduction

The term "Systems Engineering" encompasses a large number of 'engineering' tools and processes that all can provide benefit to a program, if used properly at the right time. The objective of this paper is to describe the basic thought process of risk/reward and how to navigate the elements of designing a Systems Engineering process tailored for the scope of the project. This paper is not intended to be a technical guide, but rather some viewpoints to allow project managers review their resource decisions.

Background

Projects with larger scopes can benefit from larger systems engineering activities; conversely a simple low optimization project will need less time devoted to systems engineering tools and processes.

Regardless of scope, some project teams embrace or reject Systems Engineering tools and processes.

The paper will further detail how and when the decision to implement Systems Engineering can be of benefit and detriment to a project.

Systems Engineering Tools:

Depending on the author and the audience, the term systems engineering applies to a large number of tools that can be used on a project [1][2]:

- System architecture
  - Architecture Diagrams
- System model, Modeling, and Simulation
  - System Dynamics
  - Systems Analysis
- Optimization and Decision making
  - Pugh Matrix
- Statistical and Reliability analysis
  - DFMEA
  - PFMEA
  - PHA
- Requirements & Document Management
  - Requirements Model
  - Requirements Analysis
  - Requirements Management
  - Stakeholder Map

Project Overhead

All of these Systems Engineering tools require people and infrastructure to support their use. If they are over-used on a project, its overhead can affect the project costs and/or the time devoted to project optimization.

If these tools are underused on a project, the project may not perform to expectations, have un-expected change costs, or other undesirable overhead.

Project scope:

With such large number of tools available to an engineering project manager, each manager must decide which tools to use to manage the complexity and risk on their program.

Several factors that can affect the amount of systems engineering that is applied to a program:

- Complexity
- Controls & Functional Safety
- Production Volumes
- Performance Optimization

Complexity

Complexity can be defined as the point at which an artifact cannot longer be made or comprehended by a single person [4]. Systems engineering tools help to formalize the division of work.

Relatively small projects are manageable with a less rigorous use of systems engineering tools. Larger projects with higher levels of cross disciplinary integration can benefit from systems engineering tools.

Controls & Functional Safety

Although related to complexity, typically adding mechatronic or ‘control’ systems to the project will require an increase in systems engineering tool usage to meet functional safety requirements. As an example, this is driven in the automotive industry by the ISO26262 standard which has rigid processes and requirements.

Production Volume

Smaller “R&D” projects that are not going to production will require a less rigid Systems Engineering infrastructure. However, projects with a very large high volume production program significantly increases the project risk, therefore justifying a larger infrastructure.

Performance Optimization

Projects that require higher level of performance optimization can be benefited by systems engineering to help drive the customer requirements down and optimized them between sub-systems.

Over-use of Systems Engineering

System Engineering tools can be used to the detriment of a project. Typically this occurs when using tools/processes at inappropriate times in the program or with incorrect magnitude. Examples of using Systems Engineering tools to the detriment of a project include:

- Employee Empowerment
- Blind Mandate
- Risk Avoidance
- Blind Mandate
- Risk Avoidance
- Job Positions
- Change Management

Employee Empowerment

If an organization does not believe that it has the right people or process for Systems Engineering it can react with layers of audits, peer reviews, approvals, mandatory meeting attendance, etc. all of which can cripple the engineering process.

Blind Mandate

This refers to instances in which organizations blindly mandate a version of a Systems Engineering process on all projects without regard for the resource burdens that this places on a project or the phases of a project.

Risk Avoidance

This refers to organizational over use of these tools due to risk aversion. An organization can overuse systems engineering tools as a method to avoid all risks, rather than weighing the risk to reward the resource utilization.

For example, the risk vs. benefit of doing fully detailed DFMEA/PFMEAs early in the product development cycle when the design is at a physics level of concept selection with the justification that it was to avoid risk.

Job Position Justification

This refers to individuals, groups, or organizations that try to mandate use of the tools to create workload to increase their billable hours. This can also be seen when a dedicated systems engineering organization is trying to add value to a project; but does not have direct project responsibility to the deliverables in the project. Both of these behaviors can create issues with teams that are resource limited by reducing the resource utilization on the tasks that are key to the project’s success.

Change Management

Organizational use of Systems Engineering tools as a way to limit the number of iterations that happen in the concept phases of a project. Technical departments are often an example of this behavior, overusing tools to decrease changes coming into their area of responsibility. Demanding detailed systems diagrams, interface specifications, and DFMEA/PFMEA documents before the project’s fundamental design architecture has iterated enough to reach this level of detail. This will impact the time required for design concept iteration and thus limit the ability of the team to iterate into the best design architecture for the project.

Underuse of Systems Engineering

System Engineering tools can also benefit a project. Many individuals and organizations downplay the benefit of Systems Engineering tools as they believe that they are un-needed overhead or don’t fit their model of the engineering process. Examples of neglecting to use Systems Engineering tools to their maximum effect include:

- Specialist-Centric Viewpoint
- Opinioneering
- Overhead
- Risk Undervalument

Specialist-Centric Viewpoint

Many engineers work to avoid Systems Engineering processes and tools simply because they believe that their specialty is the most important to the program. In this case, the project misses the opportunity to balance each specialty to meet the overall programs goals. This behavior often happens when the specialists' engineering work is not integrated into the overall project goal.

Opinioneering

Engineers that are subject matter experts may dismiss the Systems Engineering tools because the objectivity and extra work interferes with their ability to 'make-the-call' or "opinion-eer" the solution.

Another usage of this opinioneering is "this is how it has always been done", which can lead to a lack of change.

Increased Overhead

Simply put, the term "System Engineering" invokes images of endless overhead meetings that need to be avoided to do 'real work'. Many engineering teams reject Systems Engineering processes because a 'systems engineer' expert attends the meetings but they are not considered to add value to the program, only overhead. This often happens when the systems engineer does not have an understanding of project deliverables and is not integrated into the team.

Risk Undervalument

Some engineers do not understand the risk that is involved with serial production or projects that can cause hazard to people or the environment around the project. This is especially prevalent in engineers who have only worked in a single discipline or "one-off" products that have not seen the effects of production or end-user variation. They have not seen the un-expected effects of serial production in complex interdisciplinary products which can be resolved with proper use of Systems Engineering tools.

Milestone Viewpoint

Some programs can avoid investing in resources into various systems tools (systems simulation models, documentation, DFMEA, etc) because they are just working ot the next milestone. This often happens when a project is funded in segments, and the team is applying all resources to the next funding milestone. As opposed to investing in tool chains and work products that will reduce the long term workload and risk.

Tools

The project leader or leadership team must understand that Systems Engineering tools provide value at the different milestones in the project. Examples of effective Systems Engineering tools and their application include:

- Workload Models
- Trade Study Cycle Time
- Program Risk Evaluations
- Consequence Education
- Team Members

Workload Models

These models can help to objectively capture the manpower overhead required to complete various tasks. The project team workload applied to Systems Engineering tools must be balanced against time devoted to creatively and efficiently solving problems.

Figure 1 shows a simplified output of a basic workload model that can help illustrate the overhead that systems engineering applied to a program.

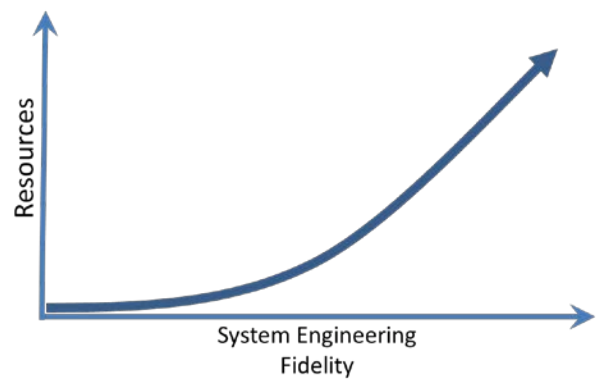

Figure 1 Workload Model

Trade Study Cycle Time –

Early in the concept or architecture phase of a project, the metric of time it takes for each trade study (or concept) iteration which allows the project development team to quickly iterate from one solution, to a better solution, to the best solution.

As an example of design cycle time, the trade study development and decomposition of requirements for racing dampers.

Figure 2 shows a typical V-Chart for a racing damper as applied across a typical program timing. In the upper left corner of the V, the early requirements decomposition and architecture definition (or Trade Studies) that define the product happen.

Multiple iterations between functional disciplines are typically preferred to help optimize the entire product.

The more design cycles that can occur the higher levels of optimization can occur.

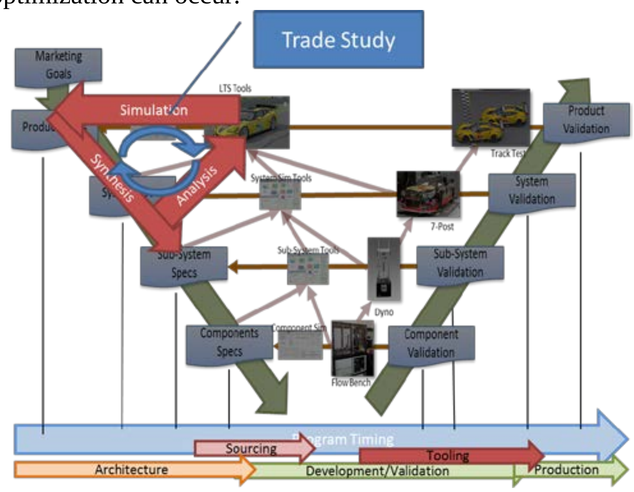

Figure 2 Trade Study Cycle Time

Program Risk Evaluations

These evaluations are a good tool to illustrate the magnitude of risk that many Systems Engineering tools were designed to help mitigate. This concept helps to quantify the risk vs. reward of using Systems Engineering tools. It can help illustrate early in the program that design iteration is more important than highly detailed Systems Engineering documentation.

Consequence Education

This refers to educating a project team with respect to the level of pain (time, investment, etc.) that is required for a team to fix an issue late in the program. This education can greatly increase a team's willingness to embrace or invest into these tools.

Figure 3 shows the costs of change as a program progresses in it timeline. This can be effective in explaining the risks to team members.

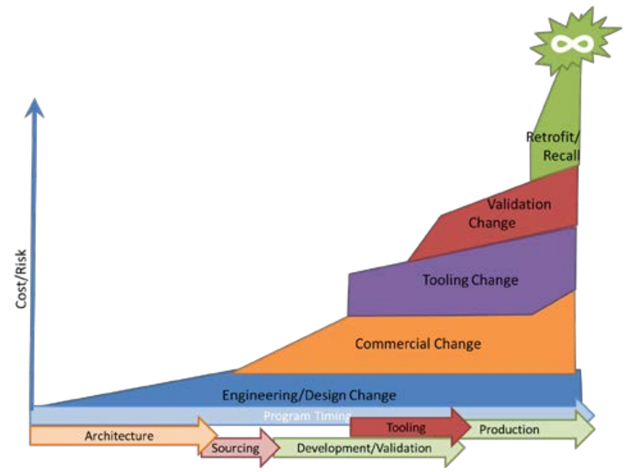

Figure 3 Change Costs

Team Members-

Assessing if all the team members have the proper engagement in the project is a key tool for a successful project. They have to "Get It", "Want It", and have the "Capability" for the project to be successful [1].

Figure 4 shows a typical lean product development graphic, it should be noted that a key leg of the triangle is the skilled people.

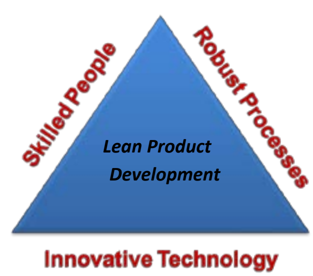

Figure 4 Lean Product Development

Although the skilled of the team is important, they can be better invested in the use of the appropriate tools if there are objective reasons for the decisions on the tools and the magnitude of their use. This can be done with the selection of the right team members and properly educating them as to the magnitude of their task.

Conclusions

In conclusion, the project leadership must determine what Systems Engineering tools are required to meet the objective of the different phases of the project.

There are some tools that can help the project leader determine which tools apply:

- Workload Models
- Trade Study Cycle Time
- Program Risk Evaluations

- Consequence Education
- Team Members

These tools can be used to make decisions, but can also be very useful in getting both management and the project team behind why the Systems Engineering tools and processes are being used on the project.

In the end, the leader must also remain flexible in their willingness to apply these tools at the right levels and time based on the project timeline and objectives.

REFERENCES

[1] System Architecture: Strategy and Product Development for Complex Systems, Edward Crawley, Bruce Cameron, Daniel Selva, Pearson

[2] https://en.wikipedia.org/wiki/Systems_engineering #Managing_complexity

[3] http://blog.eosworldwide.com/blog/95593/traction-thoughts/gwc-work/#axzz4AEd95h5O

[4] Design Rules, C. Baldwin,, K.Clark, MIT Press