Systems Engineering Lessons Learned for Class D Missions

Kristina Rojdev[1] and Irene Piatek[2]
NASA – Johnson Space Center, Houston, TX, 77058

Josh Moore[3] and Derek Calvert[4]
NASA – Marshall Spaceflight Center, Huntsville, Alabama, 35812

One of NASA's goals within human exploration is to determine how to get humans to Mars safely and to live and work on the Martian surface. To accomplish this goal, several smaller missions act as stepping-stones to the larger end goal. NASA uses these smaller missions to develop new technologies and learn about how to survive outside of Low Earth Orbit for long periods. Additionally, keeping a cadence of these missions allows the team to maintain proficiency in the complex art of bringing spacecraft to fruition. Many of these smaller missions are robotic in nature and have smaller timescales, whereas there others that involve crew and have longer mission timelines. Given the timelines associated with these various missions, different levels of risk and rigor need to be implemented to be more in line with what is appropriate for the mission. Thus, NASA has four different classifications that range from Class A to Class D based on the mission details. One of these projects is the Resource Prospector (RP) Mission, which is a multi-center and multi-institution collaborative project to search for volatiles in the polar regions of the Moon. The RP mission is classified as a Class D mission and as such, has the opportunity to more tightly manage, and therefore accept, greater levels of risk. The requirements for Class D missions were at the forefront of the design and thus presented unique challenges in vehicle development and systems engineering processes. This paper will discuss the systems engineering process at NASA and how that process is tailored for Class D missions, specifically the RP mission.

Nomenclature

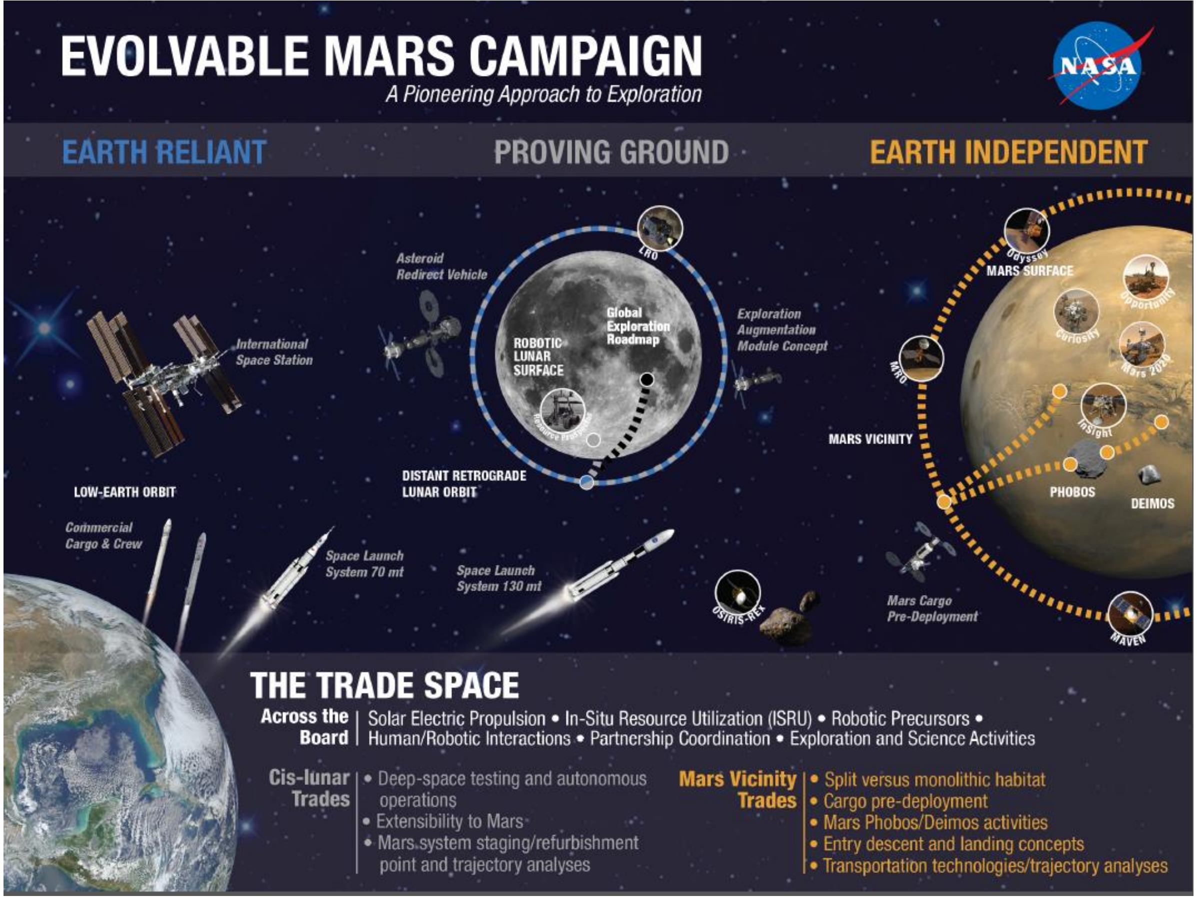
CAD	=	Computer aided design
IMS	=	Integrated master schedule
ISS	=	International Space Station
LEO	=	Low-Earth orbit
MEL	=	master equipment list
MOP	=	Measures of performance
RP	=	Resource Prospector
SEMP	=	Systems engineering management plan
TPM	=	Technical performance measures
WBS	=	Work breakdown structure

I. Introduction

ONE of NASA's strategic objectives is to "expand human presence into the solar system and to the surface of Mars to advance exploration, science, innovation, benefits to humanity, and international collaboration[1]." To accomplish this objective, NASA has developed a three-pronged approach to expand into the solar system known as the Evolvable Mars Campaign (Figure 1). In this campaign, there will be several projects devoted to mastering long-duration spaceflight for humans aboard the International Space Station (ISS), as well as working with commercial companies to provide access to low-Earth orbit (LEO). These projects are part of the "Earth Reliant" stage. The next

[1] Aerospace Engineer, Systems Engineering & Test Branch, 2101 NASA Parkway, MC EA531, and AIAA Member.
[2] Aerospace Engineer, Asset and Process Management Branch, 2101 NASA Parkway, MC EA541.
[3] Aerospace Engineer, Project Engineering Branch, MSFC/ES11.
[4] Aerospace Engineer, MSFC/ES13.

stage is known as the “Proving Ground” where there were will be several human and robotic missions to cis-lunar and lunar space to prove out various technologies needed for living and working on Mars. Finally, the last stage in the campaign is focused on human and robotic exploration of Mars, and is known as the “Earth independent” stage.


EVOLVABLE MARS CAMPAIGN
A Pioneering Approach to Exploration
NASA
EARTH RELIANT
PROVING GROUND
EARTH INDEPENDENT
International Space Station
LOW-EARTH ORBIT
Commercial Cargo & Crew
Asteroid Redirect Vehicle
LRO
ROBOTIC LUNAR SURFACE
Global Exploration Roadmap
Resource Prospector
DISTANT RETROGRADE LUNAR ORBIT
Exploration Augmentation Module Concept
Space Launch System 70 mt
Space Launch System 130 mt
OSIRIS-REX
Odyssey
MARS SURFACE
Opportunity
Curiosity
Mars 2020
MRO
InSight
MARS VICINITY
PHOBOS
DEIMOS
Mars Cargo Pre-Deployment
MAVEN

THE TRADE SPACE

Across the Board | Solar Electric Propulsion • In-Situ Resource Utilization (ISRU) • Robotic Precursors • Human/Robotic Interactions • Partnership Coordination • Exploration and Science Activities

Cis-lunar Trades
- Deep-space testing and autonomous operations
- Extensibility to Mars
- Mars system staging/refurbishment point and trajectory analyses

Mars Vicinity Trades
- Split versus monolithic habitat
- Cargo pre-deployment
- Mars Phobos/Deimos activities
- Entry descent and landing concepts
- Transportation technologies/trajectory analyses

Figure 1: Summary of the Evolvable Mars Campaign.[2]

In this summary (Figure 1), there are several different types of missions planned as part of the Evolvable Mars campaign with varying degrees of complexity. To optimize the work involved in developing the variety of these missions, different levels of risk and rigor need to be implemented to match the level that is appropriate for the mission goals. Thus, NASA has four different categories to differentiate missions appropriately based on the following qualifications: priority to agency and acceptable risk level, national significance, complexity, mission lifetime, cost, launch constraints, in-flight maintenance needs, research opportunities, and mission success criteria. The classification for a mission is set early on in the formulation process and is based on the considerations set in Table 1.

Table 1: Summary of NASA risk classification considerations.[3]

CHARACTERIZATION	CLASS A	CLASS B	CLASS C	CLASS D
PRIORITY	High	High	Medium	Low
NATIONAL SIGNIFICANCE	Very high	High	Medium	Low to medium
COMPLEXITY	Very high to high	High to medium	Medium to low	Medium to low
MISSION LIFETIME	Long (> 5 yrs.)	Medium (2-5 yrs.)	Short (< 2 yrs.)	Short (< 2 yrs.)
COST	High	High to medium	Medium to low	Low
LAUNCH CONSTRAINT	Critical	Medium	Few	Few to none
IN-FLIGHT MAINTENANCE	N/A	Not feasible or difficult	May be feasible	May be feasible and planned

ALTERNATIVE RESEARCH OPPORTUNITY/RE-FLIGHT OPPORTUNITY	None	Few or no alternative or re-flight opportunities	Some or few alternative or re-flight opportunities	Significant alternative or re-flight opportunities

The Resource Prospector (RP) mission is one of the pre-cursor robotic missions as part of the lunar proving ground in the Evolvable Mars Campaign. It is a multi-center and multi-institution project focused on investigating the lunar polar regions for volatiles. The vehicle is comprised of a lander, a rover and a payload. The lander is the spacecraft that shuttles the vehicle from the Earth to the Moon and lands the vehicle on the lunar surface. The rover then transports the payload to various locations on the lunar surface to determine the type and quantity of volatiles in those locations. In addition, the payload will then use those volatiles to demonstrate water production. Demonstrating the capability to mine surface resources and then use those resources to produce water, oxygen, or propellant will enable paths to multiple strategic objectives and new mission architectures to Mars (Figure 2).

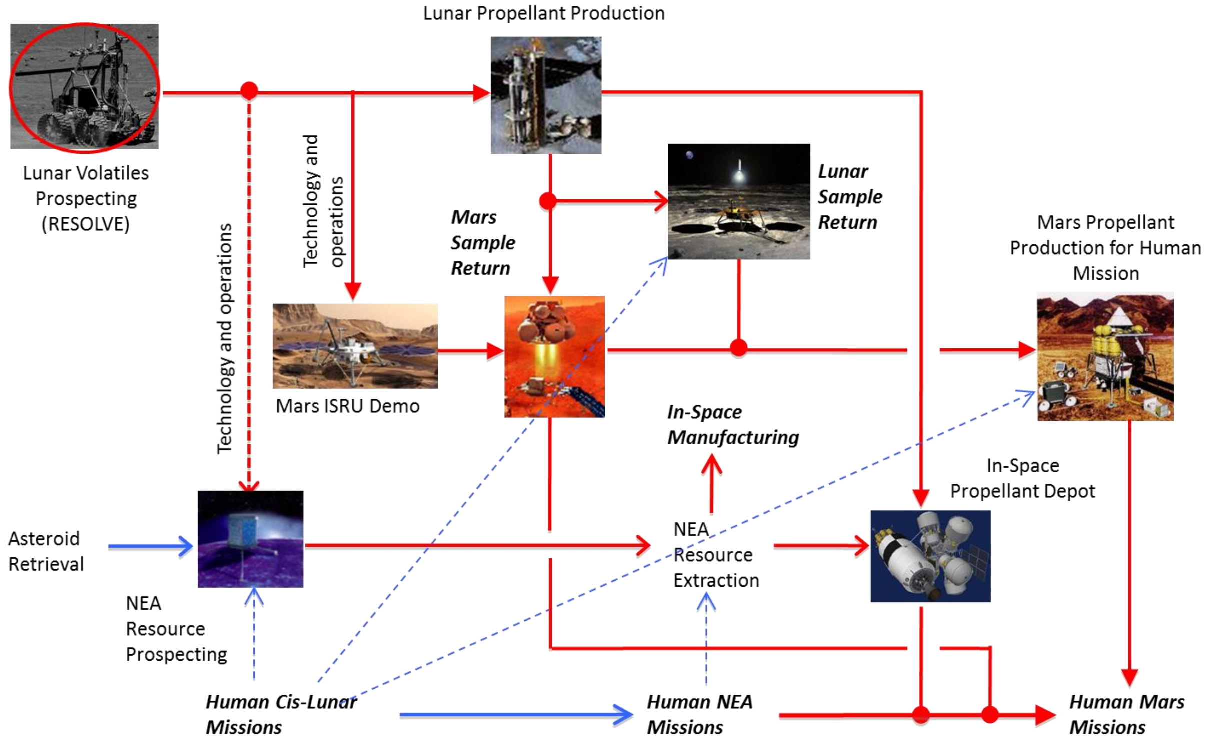


Figure 2: The paths to meeting various objectives of Evolvable Mars Campaign due to the RP mission.

The mission is rated Class D and as such, has the opportunity to more tightly manage, and therefore accept, greater levels of risk. Given this classification, the mission must have a simple design to reduce complexity, a short mission duration, and low cost. Currently the RP project is in the formulation stage with a concept for an approximately ten-day mission, including a nominal five-day direct Earth to Moon trajectory. Additionally, the design of the lander has primarily been driven by the project requirements and following a low-cost approach. This classification level, along with its potential for enabling future paths to Mars, provided unique challenges from the systems engineering standpoint. The following sections will discuss the typical systems engineering process executed at NASA and how that process was implemented for the RP lander. Since the lander is still in the formulation phase, the discussion will be limited to the section of the systems engineering process that has been implemented to date.

II. Background

NASA's typical systems engineering process has seven major phases that begin with concept studies in pre-formulation and end with closeout of the mission. In each of these phases, there are major reviews to evaluate the status of the project and to provide valuable feedback on the technical data from those outside the project. At the end of each phase, there is a key decision point to determine whether the project is of sufficient maturity to continue into the next phase. Figure 3 shows a summarized visual of this project life-cycle process and its various components.

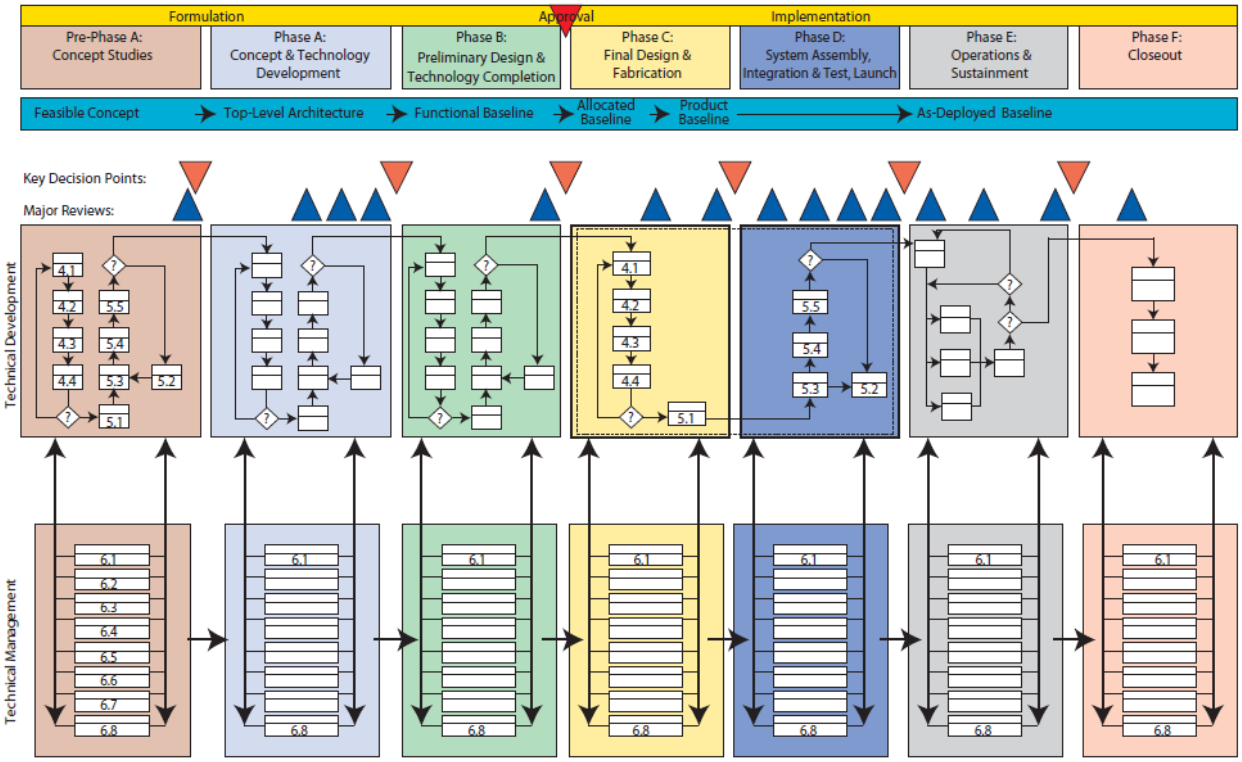


Figure 3: Summary of NASA's life-cycle process.[4]

The systems engineering process discussed here will only focus on details relevant to the Pre-Formulation phase, or Pre-Phase A. The purpose of this phase is to develop various ideas and mission concepts that are feasible, produce high-level requirements, model or simulate the concepts, and verify and validate that the concepts would meet those requirements. Three main processes occur to accomplish this: system design, product realization, and technical management. The system design and product realization processes are needed for the physical development of the product, whereas the technical management process is the set of tools that enable the product development. Figure 4 shows the interconnections between these three processes.

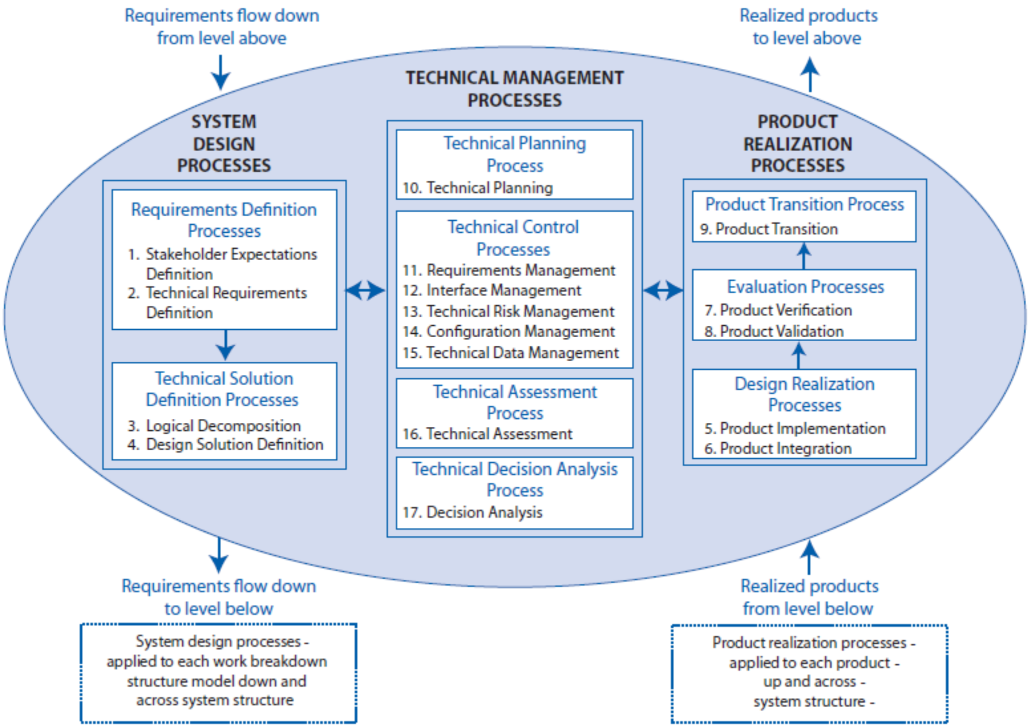


Figure 4: The systems engineering processes and their interconnections.[4]

A. System Design Process

Within the system design process, there are four main components. These constituents are the definition of stakeholder expectations, the technical requirements, the decomposition of those requirements, and the design solution.

The first step is defining the expectations of the stakeholder for the product. These expectations are key in the validation process and can include operator and user interfaces, skills, system and human performance criteria, standards and regulations, safety, quality, reliability, or constraints. Additionally, the expectations include candidate concepts of operation and measures of effectiveness.

The second step is transforming the stakeholder expectations into a set of quantitative and measureable requirements. These requirements should include a method of verification so that there is a clear manner in which to prove the requirement has been met by the product. Additionally, candidate measures of performance (MOP) are developed to satisfy the measures of effectiveness. The MOPs are comprised of functional and performance requirement combinations. Furthermore, technical performance measures (TPM) are defined to measure the technical progress. The TPMs are selected from the MOPs, are measurable, and can be compared with a projected progress profile from historical or previous test data. If the project does not meet the TPMs, then it will be at risk in terms of cost, schedule, or performance.

The next step focuses on decomposing the requirements to lower levels to provide higher fidelity and detail to subsystems, as well as define the relationships between the requirements. The requirements are also analyzed by function, time, behaviors, data flow, objects, states and modes, and failure modes and effects. The products developed during this process include functional flow block diagrams, timelines, data control flow, states and modes, behavior diagrams, operator tasks, or functional failure modes.

The final step in the system design process uses the requirements to develop several trade studies. Each trade study is analyzed until a preferred alternative is selected and fully defined into a final configuration that meets all the requirements.

B. Product Realization Process

Following the system design process is the product realization process. This process has five steps to develop the design into a product or multiple products that vet the design to the level of detail appropriate for the phase of the project lifecycle.

The first step in the process is the product implementation whereby a specific product is developed by satisfying the design solution defined in the requirements. For the Pre-phase A part of the lifecycle, the products could be reports, models, simulations, mockups, prototypes, or demonstrators. After the various subsystem products have been developed, they are assembled and integrated into the final product.

Once the final product is constructed, if necessary, the product undergoes verification and validation. Verification confirms that the product meets the requirements, typically via analyses, demonstration, inspection, or test. Validation proves that the product functions properly in the intended environment and meets the stakeholders expectations. Finally, the product is delivered to the customer, along with any pertinent documentation.

C. Technical Management Process

The technical management process occurs concurrently with the system design and product realization processes, as shown in Figure 4, to manage the work. This process consists of eight pieces: technical planning, requirements management, interface management, technical risk management, configuration management, technical data management, technical assessment, and decision analysis.

The technical planning maps out the cost estimates, schedules and resource needs to complete the work. A systems engineering management plan is put into place and methods by which to assess the progress of the technical work are defined.

The requirements management process includes defining and baselining the requirements to which the design must adhere. Defining the level of detail for the requirements, providing bidirectional traceability to the top requirements, and managing the changes to the requirements throughout the lifecycle are important aspects to the management of the requirements.

Given that the product is divided into subsystems and there are various groups that then work on those subsystems, it is important to define and manage the interfaces between the groups. Interface management includes definition of the interfaces, negotiation of which party is responsible for which aspect at the interface, and the integration at the interface.

Managing the technical risk involves defining, tracking, and mitigating any potential performance, mission success, safety, cost, and schedule impacts associated with the design. Managing the configuration of the product throughout the lifecycle is important so that each member of the team is working from consistent data when performing analyses. Finally, defining the location of the technical data and the process by which the team is expected to manage the technical data is key to keeping the team in sync and optimizing the work flow.

Near the end of the current phase in the lifecycle process, a technical assessment is required to move forward to the next phase in the design and development of the project. This assessment is usually in the form of a review where outside personnel are invited to review the products, critique the work, and provide recommendations for proceeding with the project. Additionally, if there are any alternatives to the current design that are being considered, decisions on whether to pursue those alternatives are made at this time.

D. Implementation methods

Each of these processes are meant to be customized based on project details such as scope, risk tolerance, system size and complexity, impact on other systems, safety, etc. In addition, the implementation method for these processes should be agreed upon early in the project and documented in the systems engineering management plan.

Historically, the implementation plan for large-scale, complex projects has been document driven. Thus, with each component of the systems engineering process, a document would be developed to detail the work completed in that process. Multiple people would work on these documents and one person would then have to collate all the information into a clean format for presentation to stakeholders. Additionally, the systems engineering products are interconnected and showing those connections is very challenging in a document-driven implementation process.

Thus, with the advent of technology, NASA is evaluating new methods of systems engineering implementation to optimize and streamline the systems engineering process, especially as projects have grown in complexity and become more distributed. These methods are a database driven method and a model driven method. A database driven method has the technical data in tabular format with the capability to link several pieces of information together. This method provides the capability to work more collaboratively and store information in a centralized location. Furthermore, information can be visualized and manipulated quickly in tabular format. One tool that can be used to document systems engineering products in a database form is Microsoft SharePoint. A model driven method, such as SysML, is pictorial and object oriented, using block diagrams to represent the data visually and to show linkages between the data. This method is advantageous when there are many layers of interconnecting, complex information.

Each method of implementation has benefits and shortcomings. The project as a whole must decide which method to use as the primary implementation strategy and where to apply the other methods when the primary strategy may not work. The next section provides details on the implementation of these processes for the RP lander.

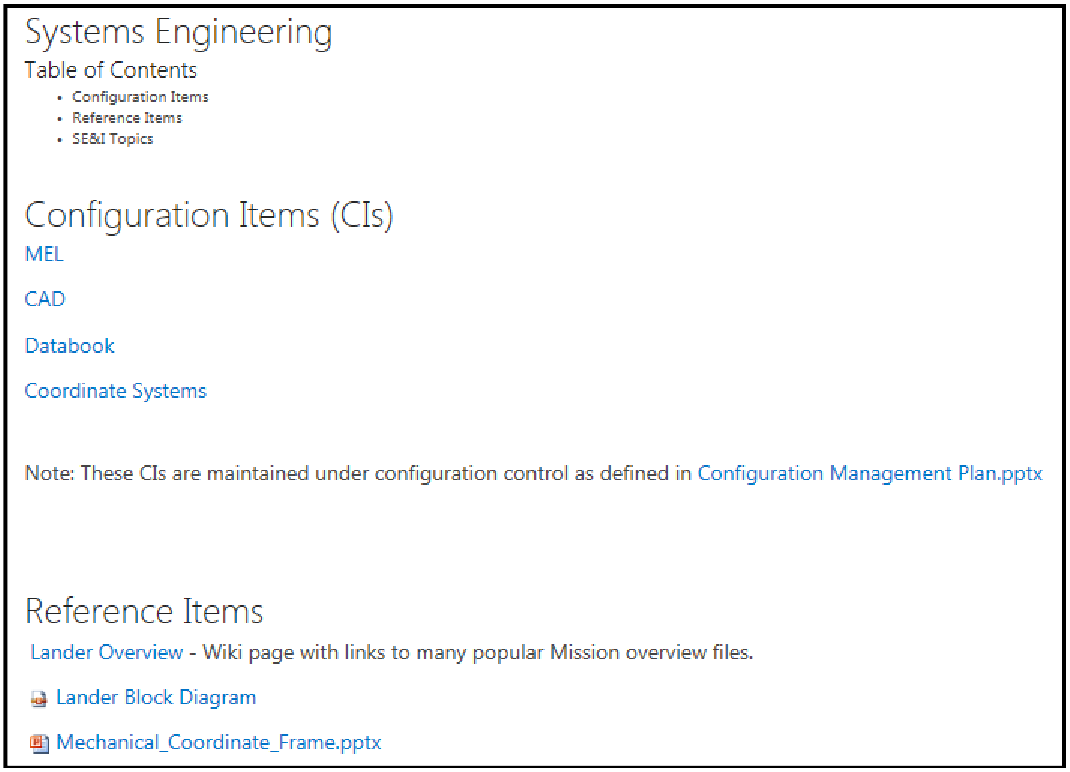
III. Systems Engineering for the Resource Prospector Lander

A. RP Lander Implementation Method

Even though the RP project is in the formulation phase, it has been necessary to implement NASA's systems engineering processes in a formal manner to maintain a high level of productivity within the project. As with all space flight projects, this activity was unique in the implementation of the process. However, even though the implementation was a divergence from the historical approach, we designed it to fully meet the procedural requirements of NPR 7123.1B[5] with no formal tailoring. It was important to limit the tailoring since institutional waivers would then have been required, adding complexity to the execution.

The implementation approach we selected was still primarily document-driven to remain in sync with the document-driven approach favored by the project leadership. However, in an attempt to optimize the systems engineering process, we started incorporating some of the database-driven approaches to alleviate the challenges of having a distributed team that needed to work collaboratively on products. We primarily focused on limiting the documents that needed to be managed to only those that were essential in getting the work accomplished. Additionally, we did not require an explicit format to the data and specifically attempted to make the data as compact as possible. For example, depicting the information using bullet points in a single slide PowerPoint presentation was preferable to a Word document that required one page of text to get the same information across. Furthermore, we experimented with capturing information in a tabular format, more akin to how databases manage information. By using this tabular format, the team could quickly access and interpret data, rather than having to read extensive text documents to retrieve the data. This method also provided opportunities to map future products to existing data, such as verification of requirements. We also investigated methods of linking data between products to make multiple pieces of information more accessible.

The tool we used to execute on this collaborative approach was Microsoft SharePoint. We set up a specific construct within SharePoint to organize the team, provide ease of collaboration, and promote communication. Capturing and disseminating all the project information was essential, even for a small team. The home page for the project was set up like a wiki page where the team could access the most recent updated information, the latest news on the project, and important items that required high visibility. Each subsystem also had their own wiki page to provide overview information on their subsystem and quick links to access subsystem specific documentation more effectively. We included a dedicated systems engineering wiki page as well with select information (Figure 5). While this page included everything from mission overviews to requirements, the most important sections were the configuration items that we will discuss in more detail below. In addition to the wiki pages, we set up a folder library as a repository of data where each subsystem could organize, manage, and distribute their data collaboratively with the team.



Systems Engineering

Table of Contents

- Configuration Items
- Reference Items
- SE&I Topics

Configuration Items (CIs)

MEL

CAD

Databook

Coordinate Systems

Note: These CIs are maintained under configuration control as defined in Configuration Management Plan.pptx

Reference Items

Lander Overview - Wiki page with links to many popular Mission overview files.

Lander Block Diagram

Mechanical_Coordinate_Frame.pptx

Figure 5: Example section of the SE Wiki Page

In addition to the structure we were able to develop within SharePoint, there were unique capabilities in the tool that we wanted to exploit. For example, we wanted the team to be able to collaboratively develop and edit documents, reducing the overhead on one individual whose responsibility would be to collate all the information from the team and put the document together. To implement this process effectively, we used versioning in SharePoint whereby an individual team member could open a document in SharePoint to make their updates and SharePoint would automatically save the update as a new version of the document. The person making the updates can provide a summary of the edits as documentation and SharePoint documents who made the edits. If someone made a mistake, it was easy to restore an old version of the document in SharePoint. Furthermore, an individual can set alerts on a product or document such that any time it is updated, the individual receives a notification. The capability to receive automatic alerts reduces the overhead on an individual to have to check the website and products constantly to see what has changed.

With any customization, it is important to communicate the implementation to the team. The typical NASA approach to sharing a project's approach is through the Systems Engineering Management Plan (SEMP). In keeping with our implementation approach, we captured the principles of the plan in a single table rather than a verbose document, providing future advantages in mapping our verification that we met the requirements of NPR 7123.1B[5] with our systems engineering approach. The table explains in a few words how we will meet the intent of the requirements.

B. RP Lander Technical Management Process

The systems engineering team used the implementation method described above throughout the technical management process. However, given that the RP lander was still at an early stage of development, the products resulting from the technical management process were not fully realized nor fully detailed.

The technical planning was mostly completed at the project management level with some support from the systems engineering team. The primary products developed were the work breakdown structure (WBS) for the lander element, an integrated master schedule (IMS), and the lander budget and resource allocations. The WBS was used to organize the lander element by subsystem, which set up the construct for how data and information was managed in SharePoint. The IMS was developed using Microsoft Project to enable linkages throughout the schedule showing the dependencies along the critical path. The IMS was also organized via WBS to stay consistent with how we managed the data. The technical planning also encompassed partnership development with other collaborators on the project.

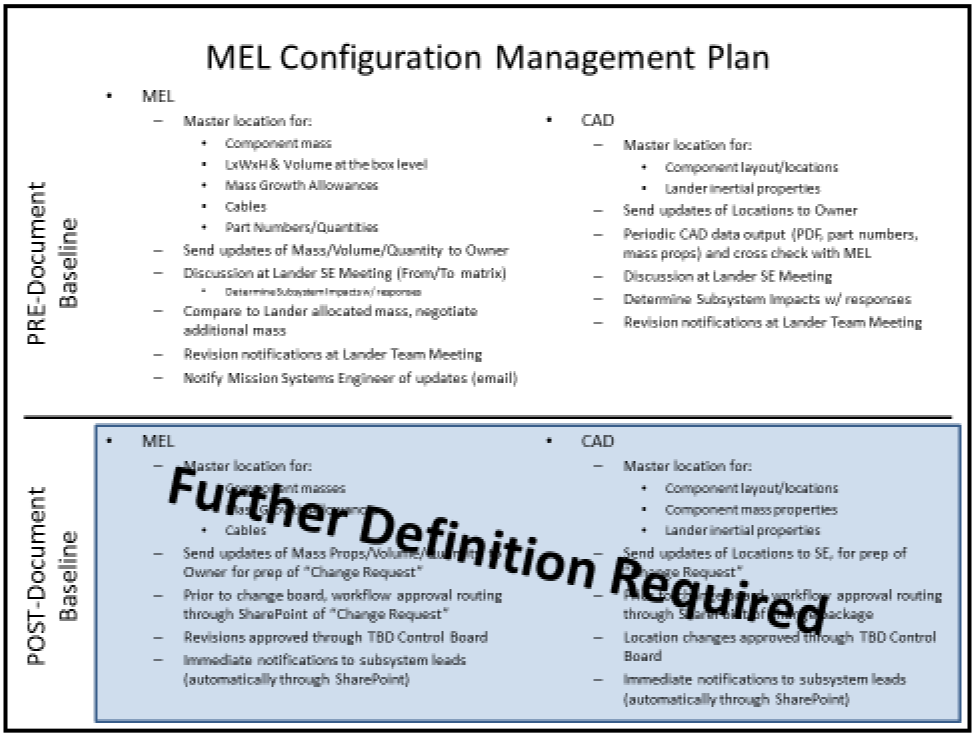
The requirements were developed using an Excel document with one point of contact for managing that document. The level one requirements are mission level requirements and the level two requirements are lander system requirements. These were developed by the project and flowed down to the lander team. The lander team then developed the level three requirements, which consisted of lander subsystem specific details.

- Interface management
    - ○ Was there a process implemented for this?

Once the design was described in detail, there was the ability to capture the inherent risk associated with certain design decisions. With any Class D project, risk must be continuously watched and communicated to properly set expectations. The RP lander was expected to track and manage all risks that were specific to the Lander element. If the consequence of the risk reached outside the element, those risks were then escalated up to the mission-wide risk system. For consistency across the project, both the Lander element and the mission risks were captured in the NASA tool ePort. This database tool allowed easy tracking, ranking, and categorization of the risks to the system.

Configuration management early on in the project was essential due to the nature of this type of project. Inevitable turnover in the team and the fast pace of development made it easier to lose track of the latest vehicle state, its subsystems, and components. In keeping with the implementation plan of limiting overhead, we wanted to find a path that required a minimum number of products to be kept under configuration control. This plan was described in a single page PowerPoint slide (Figure 6) which detailed the products under control, by whom, how they could be updated, as well as the method of communicating the changes to the entire team.

The configuration control process was flexible to accommodate the changing level of maturity. Early on, documents changed rapidly and required an agile change process. When the design matured to an appropriate level, more rigor was required. This break-over point was coined “baselining.”



MEL Configuration Management Plan

PRE-Document Baseline

- MEL
    - Master location for:
        - Component mass
        - LxWxH & Volume at the box level
        - Mass Growth Allowances
        - Cables
        - Part Numbers/Quantities
    - Send updates of Mass/Volume/Quantity to Owner
    - Discussion at Lander SE Meeting (From/To matrix)
        - Determine Subsystem Impacts w/ responses
    - Compare to Lander allocated mass, negotiate additional mass
    - Revision notifications at Lander Team Meeting
    - Notify Mission Systems Engineer of updates (email)
- CAD
    - Master location for:
        - Component layout/locations
        - Lander inertial properties
    - Send updates of Locations to Owner
    - Periodic CAD data output (PDF, part numbers, mass props) and cross check with MEL
    - Discussion at Lander SE Meeting
    - Determine Subsystem Impacts w/ responses
    - Revision notifications at Lander Team Meeting

POST-Document Baseline

- MEL
    - Master location for:
        - Component masses
        - Mass Growth Allowances
        - Cables
    - Send updates of Mass Props/Volume/Quantity to Owner for prep of “Change Request”
    - Prior to change board, workflow approval routing through SharePoint of “Change Request”
    - Revisions approved through TBD Control Board
    - Immediate notifications to subsystem leads (automatically through SharePoint)
- CAD
    - Master location for:
        - Component layout/locations
        - Component mass properties
        - Lander inertial properties
    - Send updates of Locations to SE, for prep of “Change Request”
    - Prior to change board, workflow approval routing through SharePoint of [illegible] package
    - Location changes approved through TBD Control Board
    - Immediate notifications to subsystem leads (automatically through SharePoint)

Further Definition Required

Figure 6: Configuration Management Plan for a Configuration Item.

Given that the RP Lander is in the formulation phase, leading up to the Mission Concept Review, only three items were under configuration control. These are the Master Equipment List (MEL), CAD model, and design data book, which, when combined, can fully describe the lander design. The MEL captures all the components of the vehicle, down to the part level. Even early on in the design cycle, it was essential to provide as much information as necessary to identify a specific part, including any available model numbers. Additional information was also captured in this same location, including part quantity, mass, and approximate length, width, and height details. The approximate sizes were useful data to the team, as they allowed for a more intuitive sense of how parts would be incorporated into the vehicle configuration. We used this approach so that data could be collocated and simple to find.

To then reduce duplication of configuration-controlled information, the detailed dimensions and mass properties were only captured in the CAD model. The CAD model was maintained in a strict configuration management system, which directly integrated with the CAD software. This allowed multiple designers to work on the same vehicle and maintain versioning as the design matured. The generation of a 3D PDF instantiation of the model allowed the whole team to view and manipulate the detailed model. The provided the team an opportunity to gain an understanding of the system-wide interactions between their parts without needing specific CAD software installed or familiarity with any specialized software. While the CAD shows the physical implementation of the parts and the MEL provides the physical details, additional information about the system design was still needed to communicate with the team.

The chosen method for capturing this detailed design information is the Design Data book. Any additional information that required configuration control was captured here. The basic outline included the following details (Figure 7), repeated for each subsystem.

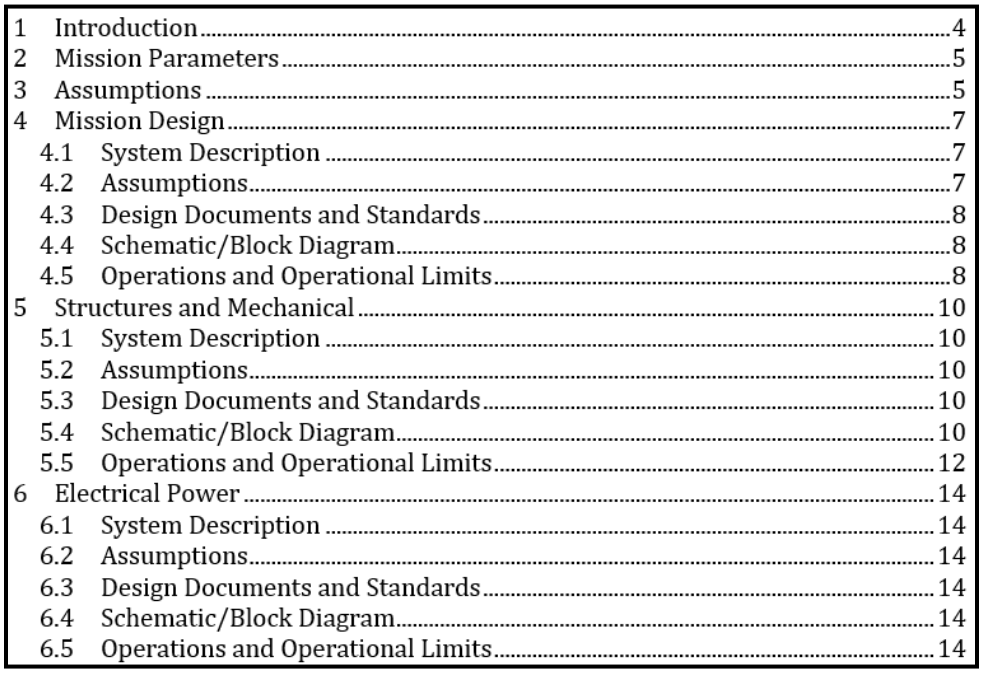

1 Introduction 4
2 Mission Parameters 5
3 Assumptions 5
4 Mission Design 7
4.1 System Description 7
4.2 Assumptions 7
4.3 Design Documents and Standards 8
4.4 Schematic/Block Diagram 8
4.5 Operations and Operational Limits 8
5 Structures and Mechanical 10
5.1 System Description 10
5.2 Assumptions 10
5.3 Design Documents and Standards 10
5.4 Schematic/Block Diagram 10
5.5 Operations and Operational Limits 12
6 Electrical Power 14
6.1 System Description 14
6.2 Assumptions 14
6.3 Design Documents and Standards 14
6.4 Schematic/Block Diagram 14
6.5 Operations and Operational Limits 14

Figure 7: Excerpt of Design Data book outline

The management of this technical data was provided through the construct developed on SharePoint, as well as the design data book mentioned above. The technical assessments were accomplished through technical reviews with primary stakeholders included. Rather than providing a compendium of several documents for review along with the presentation, the important technical details that warranted more in-depth description were singled out and provided as backup slides to the main presentation.

IV. Implementation Challenges Experienced and Lessons Learned

Technical challenges of implementing this process, both in setting up the construct and team engagement with using it, was evident throughout the project. The SharePoint system was new to the leadership team, and initially presented challenges in setting up effective filing structures and correct permission groups. The initial approach we used for permissions involved providing all files to all members of the team, allowing team members to find pertinent information that they needed. In addition, we implemented a routine whereby any time one subsystem requested information from another subsystem, the request was posted to a public folder and a link was sent to the subsystem that was to provide the information. This open approach to data sharing was implemented in an attempt to foster communication among the distributed groups and to ensure information was available to new members of the team.

Personnel changeover is always challenging to small projects. In this case, we had a few unique challenges each time a new team member was added to the lander team. Many of our team members have experience with previous lander studies, providing them with a good knowledge base of lander technologies. However, given the varied expertise, when new team members joined the project and reviewed the technical details from their experience-base, questions arose that caused old decisions to be revisited. This often slowed down the forward progress in the technical design. Furthermore, given the implementation method of limiting documentation and using SharePoint as the primary source of data management and collaboration, there were challenges in bringing new team members up to speed, which increased the time necessary to bring one's knowledge to a point where they are then productive. Simple instructions on how to use SharePoint were provided on the site, but more-depth training and practice with using the tool would help to ease the shock to new team members.

Another challenge we faced was in the configuration control process. Even though we were at an early stage in the design life cycle, we needed to implement some level of configuration control due to the rapid and iterative approach we had to design. As mentioned above, we used a somewhat informal process to keep configuration control of three main products essential to describing the design. We found that the small number of configuration items caused us to execute the change process infrequently. Thus with large gaps of time between when the configuration items were reviewed caused the team to lose track of the process and the overall vehicle configuration. In the future, a more formal schedule of reviewing configuration controlled items will alleviate this challenge.

V. Roadmap for future Class D missions

From our experience with implementing the systems engineering process for a class D mission and the challenges we faced, we recommend the following approach for future class D missions and teams of this nature. In our case, the element teams for the project were distributed and worked somewhat independently. The lead for the entirety of the project implemented a traditional document driven approach to the systems engineering process, which made it difficult for the lander team to implement too many innovations. Thus, discussing new implementation and collaboration approaches with a project lead and all the element leads at the outset of the project may achieve buy-in for new implementation methods.

In addition, achieving the right mix of implementation methods is key. Each implementation method has advantages and disadvantages. Thinking through which implementation is best for a specific product or process will help to optimize the systems engineering process as a whole.

Furthermore, agreement across all factions on the tools to be used and collaboration methods should occur early in the project development such that everyone is using the same method. This consistency in the tools and methods across all elements will help to reduce overhead in converting products to meet requested formats.

Once the implementation and collaboration tools have been agreed upon at a higher level, there needs to be clear communication in disseminating that information to the element teams. To increase engagement and buy-in from the element team, we recommend providing sufficient training on the tools the team will use and providing them an opportunity to get familiar with the tools. To facilitate engagement with the tools, it helps to have a construct/template in place that the team would then use for data collection, collaboration, product development, and reporting.

In addition to the implementation methods, this fast-paced, higher-risk, low-cost type of mission requires a team with leaders and members that are agile, self-reliant, organized, and constantly communicating. The team size for this type of project should be kept to a minimum to reduce organizational complexity and facilitate communication that is more frequent. Team members should have knowledge of multiple aspects of their subsystem and know when to pull in an expert on a specific task, rather than having a large subsystem team with multiple members that only have expertise on one specific component. Having a small team with team members that contain general systems knowledge and expertise in specialized areas provides the ability of the team to constantly look at interfaces at the lower levels and higher levels of the subsystem and recognize risks from an integrated perspective.

Furthermore, a smaller team that can be co-located is more efficient than a large, distributed team. From a cost and schedule perspective, having a co-located team allows simple questions and discussions to occur easily and quickly. Reducing the organizational complexity provides for quick turnaround in reviewing and accepting changes that impact the system, as well as disseminating that information to the team as a whole.

VI. Summary

In this paper, we reviewed the typical NASA systems engineering process and implementation methods. We then discussed how this process was implemented for the Resource Prospector lander team, and the challenges and lessons learned that resulted. From our experience, we provided some recommendations that other Class D projects might want to employ as they consider the framework and implementation of their systems engineering process. In our future work, we will continue to experiment with new implementation methods or optimizing the current implementation methods to better refine the systems engineering process, increase productivity, and reduce overhead.

References

[1]NASA, “NASA Strategic Plan,” 2014.
[2]Crusan, J., “Evolvable Mars Campaign,” 2014.
[3]NASA, “Risk Classification for NASA Payloads,” NPR 8705.4, 2004.
[4]NASA, “NASA Systems Engineering Handbook,” SP-2007-6105, 2007.
[5]NASA, “NASA Systems Engineering Processes and Requirements,” NPR 7123.1B, 2013.