Spring 2023

Tailoring a System Engineering and Management Plan for a University Satellite Team

Jacob D. Anderson
Missouri University of Science and Technology

TAILORING A SYSTEM ENGINEERING AND MANAGEMENT PLAN FOR A UNIVERSITY SATELLITE TEAM

by

JACOB DANIEL ANDERSON

A THESIS

Presented to the Graduate Faculty of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

in

AEROSPACE ENGINEERING

2023

Approved by:

Dr. Henry Pernicka, Advisor
Dr. Daoru Han
Dr. Steven Corns

ABSTRACT

A Systems Engineering and Management Plan (SEMP) helps enable a successful engineering project or team. All university teams working on satellite development encounter similar obstacles that are not commonly seen in industry. As a result, university teams must determine new approaches to solve these unique and difficult problems. A SEMP can mitigate risks, solve problems, and provide streamlined solutions. A modified SEMP is proposed along with guidance for implementing it into a new team and an existing team.

The adopted SEMP provides a road map for the technical execution and program management of a university team. The SEMP should be periodically updated as the team develops, when activities move from plans to historical facts, when known risks are mitigated or altered, when new risks are identified, when new tools and technologies are adopted, and as a myriad of other factors suggest needed adjustments to teams' overall technical approaches. It is expected that an SEMP will evolve over time and be revised as needed.

Missouri S&T Satellite Research Team's (M-SAT) SEMP is provided as an example that can be tailored for other teams to adopt. Included in this thesis are methods based on lessons learned and successful practices implemented by the M-SAT team. Lessons learned include improvements to program management and system engineering practices. Successful innovations include the creation of a SEMP, the implementation of a human resources team, and a new member program among others.

ACKNOWLEDGMENTS

There are numerous people in my life who supported me and enabled me to be successful in my life. Without the ones listed below, my collegiate career would have been dramatically different, and I would not have been as successful or been offered the unique opportunities.

I would start with Dr. Hank Pernicka and I thank him for his choice of recruiting me for the Chief Engineer position in the summer of 2021. This decision led to numerous paramount connections and the opportunity to write this thesis. I thank Dr. Daoru Han and Dr. Steven Corns for their willingness to be on my committee and their support. I am thankful to Kartikeya Veeramraju for his selflessness in mentoring me in the beginning of my time with M-SAT, which enabled me to be successful in multiple areas in the M-SAT Team.

I genuinely thank every student who interacted with M-SAT, whether or not you decided to join or stay on the team. All those students provided the feedback and experience necessary to write this thesis. Specifically, I would like to thank Joseph Nguyen, Ian Williams, and Collin Gentry. These members of M-SAT, and close friends of mine, supported me in being a leader for M-SAT and enabled me in successfully doing so.

I thank my family for their unwavering support through my college years. Lastly, I would like to show appreciation to my girlfriend of more than two years, Yordie Schnieder, for her amazing and unwavering support of my academics, sports, and anything else about which I am passionate.

TABLE OF CONTENTS

Page

APPENDICES

LIST OF ILLUSTRATIONS

1. INTRODUCTION

In the last decade, the space industry saw tremendous growth in activity and interest. Globally, many nations have been investing more in space, and the results are evident. In fact, the number of satellites launched globally has increased exponentially, as seen in Figure 1.1[1].

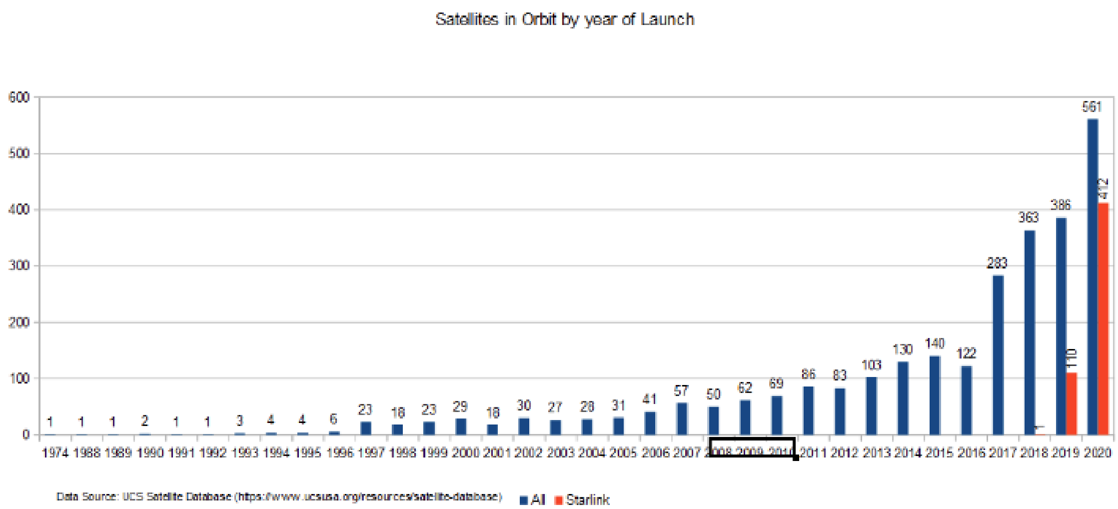

Figure 1.1. Satellites in Orbit by Year of Launch.

Related to the growth is an increased investment made by the United States government in space. Figure 1.2 illustrates the growth of the global government expenditures in space compared to the United States expenditures since 1990, along with forecasted growth through 2033[2]. This growth is evident in the increasing number of satellites launched by the United States. Figure 1.3 shows the total small satellite launches per year and by what entity[3]. A noticeable increase is seen in the number of university teams' launched satellites. This is an impressive feat for all of these teams, but this did not come without complications. University teams are continuing to struggle to produce space-qualified satellites. All university teams are subjected to the ruthless process of competing in real-world environments with fewer resources and experience than aerospace companies in the industry. Many of these universities struggle with the same challenges unique to these teams and not seen in the industry. This thesis study explored those challenges and proposes a system to avoid the common challenges and mitigate risks with unobtainable solutions.

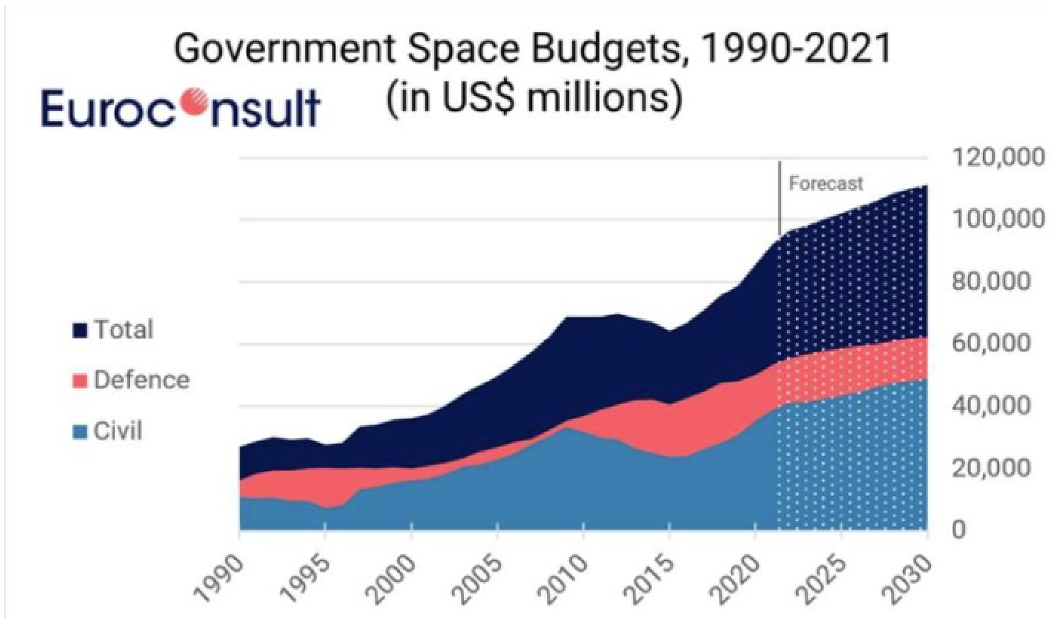

Figure 1.2. Global Government Space Budget.

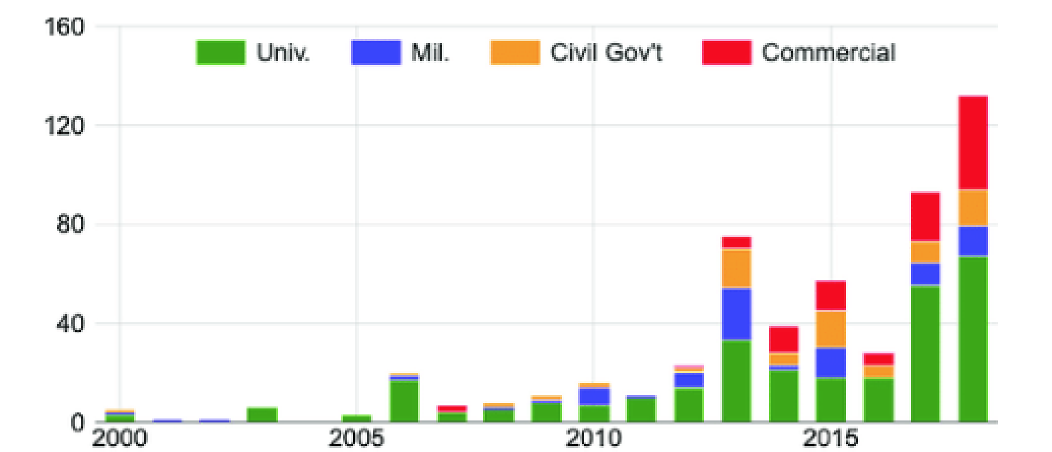

Figure 1.3. United States Small Satellite Launch by Type of Entity.

1.1. PURPOSE

This thesis provides guidance for any university team who wishes to create or improve the management of its members by creating a Systems Engineering and Management Plan (SEMP) tailored specifically for their team. The discussion in this thesis provides guidance for a university team with aspirations to build and launch a small satellite into space. However, many of the foundational principles discussed in this thesis could be applied to most university engineering teams.

1.2. MISSOURI S&T SATELLITE RESEARCH TEAM

The Missouri S&T Satellite Research Team, M-SAT, is a university satellite team that designs and builds small satellites to enhance state-of-the-art technologies. The team was founded in 2001, and since then has actively participated with the Air Force Research Laboratory in their University Nanosatellite Program (UNP) and NASA with their Undergraduate Student Initiative Program (USIP). Refer to Sections 2.1.1 and 2.1.3 for more information about UNP and USIP, respectively.

Since M-SAT's establishment, the team has supported three missions: MR and MRS SAT, M^3, and APEX. APEX is no longer supported and M^3 is in flight integration and manifested for a launch in January of 2024; MR and MRS is scheduled for a pre-integration review in April of 2023 with flight integration following a successful pass of the review. The MR and MRS SAT mission was developed to test a stereoscopic imaging system and cold gas propulsion system using refrigerant, R-134a, as the propellant. MR and MRS SAT was accepted into the NS-8 competition in 2015 with UNP, and APEX was a part of the NS-10 Competition. M^3 is sponsored by USIP and the mission will demonstrate and characterize the electric mode of a multi-mode propulsion system.

1.3. AUTHOR'S INVOLVEMENT

At the time of this writing, the author has been an M-SAT team lead for nearly two years. After joining M-SAT in April 2021, the author immediately started training to be the Chief Engineer for the MR and MRS SAT mission. For the next two years, the author served as Chief Engineer and supported many subsystems, primarily Command and Data Handling (C&DH) and Communications. The author additionally created a new entity called the ”Resource Management (RM) Team”, and served as the RM Team lead, ”Student Director”. Outside of the MR and MRS SAT mission, the author contributed to the M^3 mission as a technical advisor by supporting many efforts in both hardware and software development. As the overall M-SAT team lead, the author engaged with numerous faculty and directed and supported more than 200 students during his time on M-SAT.

As Command and Data Handling lead, the author supported both the hardware and software sub-teams. With hardware, the author designed and improved an in-house designed motherboard and integrated equipment into a microsatellite flat-sat layout for signal processing and verification testing, including Raspberry Pis, microcontrollers, Sun sensors, IMUs, GPS, torque coils, magnetometers, pressure transducers, and solenoids. While supporting the software team, the author led the development of a multi-threaded algorithm that handled the communication, control, and state of a small satellite with autonomous space situational awareness. While supporting the communications subsystem, the author conducted a trade study in collaboration with the Communications lead for a complete redesign of the communications system including ground station equipment, satellite radio equipment, and link budget. He additionally led the hardware redesign and software development for the new communications system. As Student Director, the author created and supervised the RM Team and structured new program management systems and systems engineering guidelines that the RM Team uses to support the active M-SAT missions.

2. LITERATURE REVIEW

2.1. UNIVERSITY SATELLITE TEAMS

University satellite teams are groups of students, faculty members, and industry partners who come together to design, build, and launch small satellites for scientific research, educational purposes, and/or technology development. Successful teams are generally interdisciplinary, with members from various fields such as engineering, physics, computer science, communications, and more. The teams can vary in size from five students to more than one hundred. Regardless of size, the discussions in this thesis will hopefully help improve and streamline the management process and systems engineering of a university satellite team.

University satellite teams are an important part of the space industry by providing valuable learning opportunities for students that are unique and life-changing experiences. This allows students to ”hit the ground running” once they join the industry. University satellite teams must overcome numerous technical and logistical challenges to achieve their objectives. They must select appropriate mission objectives, design and test their satellites, secure funding, obtain regulatory approvals, and coordinate with launch providers. They must ensure that their satellites can communicate with ground stations and withstand the harsh conditions of launch and the space environment. Thus, students who participate have far greater knowledge and respect for program management and technical disciplines because the opportunities and challenges faced by these teams are not typically taught in the current university curriculum [4]. As a result, students are more competitive when applying for internships, co-ops, and full time jobs. Employers benefit from recent graduates having real-life experience in the space industry with previous failures and successes with satellite development.

2.1.1. University Nanosatellite Program. Established in 1999, UNP was the first federally-funded program dedicated exclusively to university participation in spacecraft development [5]. With a history of success, UNP can be a valued sponsor for university teams. UNP's mission is to train the students by providing a real-life experience and clear-cut system that teaches student satellite development while focusing on proper systems engineering principles and practices [6]. Through a competitive and rigorous program, getting into the program's Phase A and Phase B can secure funding and guidance for the teams. Forty universities are a part of UNP and fifteen missions have flown with them since the establishment of UNP [5].

2.1.2. CubeSat Launch Initiative. The CubeSat Launch Initiative (CSLI) is a program sponsored by NASA that provides launches for university satellites. If a university can secure funding for a CubeSat, accepted universities will receive full funding from NASA for their launch and be provided access to NASA's environment testing facilities. Since its beginning, CSLI has launched over 150 CubeSats [7]. CSLI announces requests for proposals on a yearly basis offering a consistent alternative to UNP if a team is able to secure funding for satellite development elsewhere.

2.1.3. Undergraduate Student Initiative Program. USIP is sponsored by NASA, and the program management is managed by NASA's Goddard Space Flight Center and Wallops Flight Facility in Virginia. USIP provides hands-on flight opportunities to enhance science, technical, leadership, and project skills of undergraduate student teams [4]. M^3 made Missouri S&T one of 47 universities to join the program in 2016. Of those 47 universities, 22 other programs were similarly building CubeSats [4]. With $8 million in funding to provide the 47 universities, the program has seen 38 of those missions fly or have manifested launches as of 2019 [4].

2.2. PROGRAM MANAGEMENT AND SYSTEMS ENGINEERING

Systems engineering and program management are two interdependent and highly complementary disciplines that are essential for the successful completion of complex projects. Both systems engineering and program management play a critical role in ensuring that a project is completed on time, within budget, and to the required level of quality. Program management and systems engineering are critical disciplines of university satellite teams and most multidisciplinary engineering teams. In the author's experience, students often do not clearly understand the difference between the two nor how to implement proper principles from either into a team, resulting in poor team management and increasing the risk of a satellite mission.

The interaction of systems engineering and program management is crucial for the successful delivery of satellites. Systems engineers work to ensure that the subsystems and components of a system are integrated effectively, and that the system as a whole meets the requirements of the stakeholders. Program managers work to ensure that the various subsystems that make up the project or program are coordinated effectively and delivered on time and within budget. Systems engineers work closely with program managers to ensure that the design of the system is aligned with the overall goals of the program, and that any changes to the system do not negatively impact the program as a whole. Program managers, in turn, work closely with systems engineers to ensure that the various projects within the program are integrated effectively, and that the overall program meets the requirements of the stakeholders.

There are three primary interrelated aspects of the management of a multidisciplinary team: managing technical aspects, managing the project team, and managing the cost and schedule [8]. Typically, systems engineers manage the technical aspects, program managers manage the project team, and they both collaborate on managing the cost and schedule. Without a dedicated systems engineer and program manager, a satellite mission will likely not be successful [6].

2.2.1. Systems Engineering. Systems engineering is a discipline that deals with the design, development, and operation of complex systems [9]. Systems engineering is conducted in a way that implements logical thinking in order to operate a multidisciplinary team to achieve a specific goal [8]. Students typically do not have system engineering knowledge because of the lack of inclusion of multidisciplinary systems in most engineering curricula [10]. This does not mean students cannot be systems engineers, but they must learn to be critical thinkers and gain multidisciplinary knowledge. Students with systems engineering knowledge can see technical aspects of the team much clearer, provide quality feedback, and prevent issues before they arise [11].

The goal of systems engineering is to ensure that the final product meets the needs of the customer, while also being efficient, reliable, and cost-effective [12]. There are three primary aspects of systems engineering; system design processes, product realization process, and technical management process [8]. System design processes involve the preliminary design process that requires compiling stakeholder expectations into mission requirements and designing a mission around those system requirements [8]. Product realization process involves product implementation, integration, verification, validation, and transition [8]. Technical management handles the technical planning, requirements management, risk management, technical assessment, and decision analyses [8].

A systems engineer is responsible for the practice, planning, conducting, and management of systems engineering during all phases of a project [9]. System engineers are also commonly called Chief Engineers, Lead Engineers, and Program Engineers. The system engineers are in charge of managing conflicts that arise throughout the life of the project while maintaining an acceptable level of risk [9]. It is not uncommon for there to be multiple systems engineers for large and complex problems [8].

2.2.2. Program Management. Program management is the process of overseeing the execution of projects or programs, with the goal of achieving strategic objectives. Program management typically involves identifying the program's objectives, defining the scope, developing a program plan, and identifying the resources needed to successfully execute the program [13]. The program manager must also manage the relationships with stakeholders, project managers, team members, sponsors, and customers, to ensure that everyone is aligned and working towards the same goals.

One of the key outcomes of program management is that it helps organizations to achieve their goals more effectively by aligning multiple groups toward a common goal. By managing a program, an organization can ensure that all of the groups within the program are coordinated and integrated, thus reducing the risk of duplication or conflicting efforts [13]. Additionally, program management enables organizations to optimize their resources by prioritizing and allocating them across the various groups within the program [13].

Effective program management requires a set of skills and competencies that are often different from those required for a typical engineering curriculum. Program managers need to be skilled in strategic thinking, problem-solving, communication, leadership, and relationship management [13]. They must be able to manage ambiguity and complexity, make informed decisions, and work effectively with multiple groups [13].

2.3. RISK MANAGEMENT

Risk Management is the implementation of specific procedures or principles to eliminate, minimize, and understand the risks of engineering projects. Risk and risk tolerances vary from project to project. For example, the construction of bridges has extremely low-risk tolerances and low risk of failure [14]. While satellite missions typically have low-risk tolerances and a high risk of failure [14]. The combination of low-risk tolerance and high risk of failure results in risk management being an important aspect of satellite missions.

NASA's approach to risk management is divided into two methods, Risk-Informed Decision Making (RIDM) and Continuous Risk Management (CRM). These are complementary processes and CRM typically follows RIDM [15]. Overall, NASA's risk management plan at the simplest is the identification of risks then applying RIDM and managing known risks with CRM. There are five critical roles in the risk management process as follows: stakeholders, risk analysts, subject matter experts, technical authorities, and the decision maker [15]. Stakeholders are those who have interests invested in the program, risk analysts are those who handle the risk management, subject matter experts are those with the technical experience and wisdom to provide support via technical knowledge, technical authorities are those whose opinions or declarations hold decision influencing power, and the decision maker(s) are who have the authority and responsibility with making a decision. All five of these roles can affect the outcome of a decision, and therefore all five roles should be involved in the risk management processes discussed in the following sections.

2.3.1. Risk Informed Decision Making. RIDM is typically done in the first phase of the mission as a way to identify risks and find solutions to eliminate or minimize risk, or to mitigate risk moving forward [15]. RIDM is split up into six steps, as follows:
RIDM Process:

1. Understand the stakeholder expectations and drive performance measures
2. Compile feasible alternatives
3. Set the framework and choose the analysis methodologies
4. Conduct the risk analysis and document the results
5. Develop risk-normalized performance commitments
6. Deliberate, decide on an alternative, document the decision rationale

Starting with step one, the stakeholder expectations form the top-level requirements in the RVM. The top-level requirements are typically vague or complex. There are multiple approaches to obtaining tangible requirements to include in a risk management plan. The first is an examination of the lower-level requirements in the RVM that are applicable to the risk and scenario. Another method is to consider budgets, schedules, or other documentation to determine constraints. Another source that is quantifiable is safety constraints. Aside from constraints, performance measures should be defined. A performance measure is a way to identify if a constraint is met [15]. For example, a typical constraint for a satellite is cost. If the budget is $100K, the constraint is $100K, but the performance measure is the project cost. Overall, performance measures should be compiled to accurately track the risk based on constraints, which enables progress in the risk management process [15].

The second step, compiling feasible alternatives, is simply searching for possible alternatives while minimizing decision traps [15] (see Section 2.3.4). During this step, feasibility must be identified. However, most of the analysis and examination are done in the following steps. It is pertinent that multiple alternatives are identified to ensure the best solution and to have a list of alternatives if the decided-upon solution does not prove to be effective.

In step three, set the framework and choose the analysis methodologies, the types of analyses are determined [15]. The analyses selected must support the decision-making process [15]. Therefore it is critical to ensure all analyses are considered during this step to ensure nothing is missed; satellites make this a multidisciplinary process. For example, a guidance, navigation, and control analysis can be dependent on the propulsion analysis. Redoing the propulsion analysis would require revisiting the GNC analysis as required by proper systems engineering practices. This requires traceability within the team, which is why risk management processes should be communicated and documented. The previous example is a technical analysis, but there are other analyses that should be considered including safety, schedule, personnel power, reliability, and operations. Structuring the analyses is also a part of this step; configuration control should be set up to ensure all parties are working with the same information prior to moving into step four [15]. Additionally, analysis outputs should be defined to understand what information is needed from the analyses to provide traceability and a sense of the bigger picture to everyone [15].

Once the risk analysis framework is established then step four, conducting the risk analysis and documenting the results, can be completed. The goal of step four is to conduct any analyses and quantify performance measures [15]. Once quantified data are in hand, one should also model the uncertainty in the data to support the decision-making process [15]. Though qualitative information is important to consider in the decision-making process, if there is a way to model qualitative information into quantitative data confidently and accurately then one should do so. If this is not the case then one should be prepared to assess and weigh the qualitative information during the decision-making process. Additionally, an assessment of the credibility of the risk analysis results should be considered during this step. [15].

Step five is developing risk-normalized performance commitments. A performance commitment is a performance measure that is compared to the ideal performance [15]. For example, comparing a sensor's cost to the budget allocated for it. The risk normalized approach means that the performance measure be ordered for priority and ideal performance is the same for all alternatives [15]. This allows for a uniform comparison between alternatives. One should also determine risk tolerances to create acceptable ranges within the performance measure range. The risk tolerances should still satisfy the requirements. The relationship to the composed constraints and the priority level of the performance measure should be considered when establishing risk tolerances [15].

The last step is to deliberate, select an alternative, and document the decision rationale. Prior to deliberation, two objectives must be met. First, the members of the deliberation group should be determined and they should only be those with functional roles; see Figure 2.1 for more on function roles[15]. Next, a Technical Basis for Deliberation (TBfD) should be created. This documents steps one through five so that the deliberation team has access to the same information and other future purposes [15]. The TBfD also serves as a guidebook on discussion during the deliberation. Deliberation may be an iterative process because new risks, analysis flaws, and other information can arise during the deliberation which may result in changed baselines and required updates to the TBfD prior to a final decision [15]. Once a final decision is made, it will be documented in a Risk-Informed Selection Report that outlined the deliberation process and the reasoning for an alternative selection.

RIDM Functional Roles*

Stakeholders - A stakeholder is an individual or organization that is materially affected by the outcome of a decision or deliverable; e.g., Center Directors (CDs), Mission Support Offices (MSOs).

Risk Analysts – A risk analyst is an individual or organization that applies probabilistic methods to the quantification of performance with respect to the mission execution domains of safety, technical, cost, and schedule.

Subject Matter Experts – A subject matter expert is an individual or organization with expertise in one or more topics within the mission execution domains of safety, technical, cost, or schedule.

Technical Authorities – The individuals within the Technical Authority process who are funded independently of a program or project and who have formally delegated Technical Authority traceable to the Administrator. The three organizations who have Technical Authorities are Engineering, Safety and Mission Assurance, and Health and Medical. [11]

Decision-Maker – A decision-maker is an individual with responsibility for decision making within a particular organizational scope.

*Not to be interpreted as official job positions but as functional roles.

Figure 2.1. Functional Roles.

2.3.2. Continuous Risk Management. CRM is done throughout the mission cycle to manage known risks. CRM consists of knowing the scenario, the likelihood of the scenario, and the consequence or outcome of the scenario [15]. An overview of the CRM process can be seen in Figure 2.2[15].

The CRM starts with the identification of the scenario in the form of a risk statement. The risk statement should minimally contain the ideal condition, the departure from the ideal condition, the asset under risk, and the consequence of departure [15]. The risk statement's purpose is to provide an understanding of the scenario and risk. The analysis step consists of an analysis of departure from the ideal condition and the magnitude of consequence it would cause [15]. Next, the plan step involves either accepting the risk or taking action to mitigate the risk. Alternatively, one could decide to elevate the decision-making process to someone higher in the program or organization, such as a CE, PI, or even the sponsor. Mitigating the risk can be done by creating contingency plans or by researching risk drivers and preventative measures [15]. Once the plan of action is defined for the risk and scenario, then tracking should begin. This entails the collection of any data or monitoring of the situation and the risk drivers, then taking the proper course of action defined in the planning phase [15]. The last step is control, where the outputs of the tracking step determine the course of action. If the data or observations show that the risk is properly mitigated then a control document should be made. If the data or observations shows deficiency in the plan of action then the planning step must be revisited and revised. Regardless of the outcome, the process should be formally documented so that the CRM process is formally recorded and communicated to the necessary personnel.

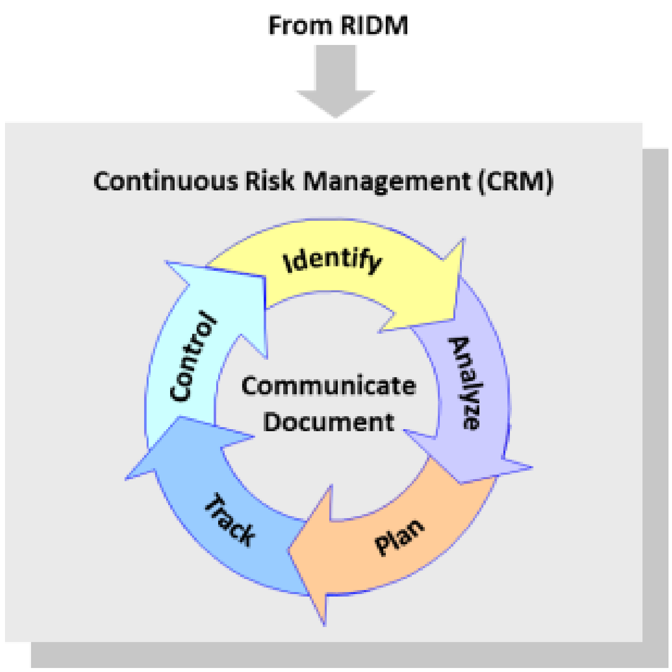

Figure 2.2. CRM Process.

2.3.3. RIDM and CRM integration. RIDM decides how to meet the requirements or constraints and CRM handles the management of the risks associated with implementation. As shown in Figure 2.3, the results of the RIDM process create a scenario and risk statement for the identification step of the CRM process[15].

During the planning phase of the CRM, newly identified risks or scenarios could be discovered and may result in a re-baseline of the RIDM process [15]. Prior to the re-baseline, the severity of the new information should be considered to determine how to adjust the original RIDM documentation. It is rare for the RIDM to be completely redone, so the goal of the baseline is to minimize the changes while still conducting a quality update. Once the re-baseline is completed, then one should move forward again in the CRM process starting with step one.

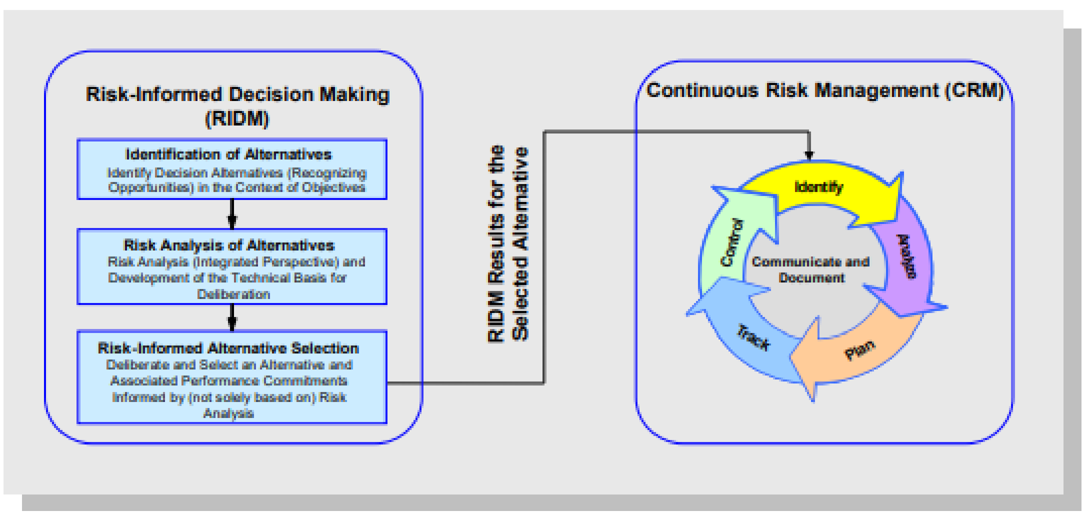

Figure 2.3. RIDM to CRM Integration.

2.3.4. Decision Traps. Decision traps are the risks that come when risk management and making quality risk-informed decisions [15]. The best way to avoid decision traps is to be aware of them; see below for a non-comprehensive list.

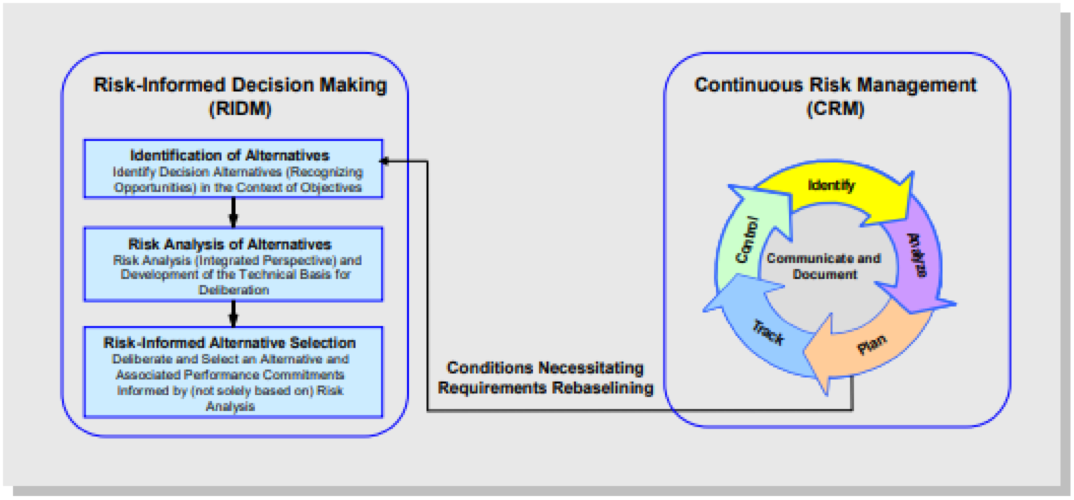

Figure 2.4. CRM to RIDM Rebase.

Sample of Decision Traps:

- Anchoring - This trap is the tendency of decision-makers to give disproportionate weight to the first information they receive [15]. It is related to a tendency for people to reason in terms of perturbations from a baseline perception, and to formulate their baseline quickly and baselessly [15].
- Status Quo Bias - There is a tendency to want to preserve the status quo in weighing decision alternatives, meaning that people tend to avoid doing things outside norms during risk assessment and decisions [15].
- Sunk-Cost — This refers to the tendency to throw good money after bad: to try to recoup losses by continuing a course of action, even when the rational decision would be to walk away, based on current knowledge [15].
- Confirmation Bias — This refers to the tendency to give greater weight to evidence that confirms prior views and even to seek out such evidence preferentially [15].
- Framing — This refers to a class of biases that relate to the human tendency to respond to how a question is framed regardless of the objective content of the question [15].

- Overconfidence — This refers to the widespread tendency to underestimate the uncertainty that is inherent in the current knowledge [15]. While most experts will acknowledge the presence of uncertainty in their estimates, they tend to do a poor job of estimating confidence intervals [15].

2.4. SYSTEMS ENGINEERING AND MANAGEMENT PLANS

According to NASA, "systems engineering is defined as a methodical, multi-disciplinary approach for the design, realization, technical management, operations, and retirement of a system" [8]. As such, a Systems Engineering and Management Plan (SEMP) documents systems engineering practices and requirements addressed throughout the mission life cycle. An SEMP should be created during mission concept development and will help guide engineering in that program [8]. Once created, the SEMP should be continuously synchronized with master plans, schedules, and documentation to ensure proper systems engineering and program management are being followed [8]. If the creation of an SEMP is delayed, then complications can arise that would have otherwise been prevented or mitigated with a proper SEMP.

An overview of a conventional SEMP derived from the ISO/IEC/IEEE International Standard for Systems and Software Engineering is outlined below. The provided outline of the conventional SEMP is meant to be tailored to any program. This will be compared to the proposed modified SEMP, outlined in Section 3, and compared in Section 3.2.

1. Technical Project Summary
   (a) Purpose, Scope, and Objectives
   (b) Assumptions and Constraints
   (c) System Description
   (d) Schedule and Budget Summary

2. References

3. Definitions

4. Program Organization

5. Technical Processes

   (a) Mission Analysis

   (b) Requirements Management

   (c) Design Definition and Traceability

   (d) Systems Analysis

   (e) Implementation

   (f) Integration

   (g) Verification and Traceability

   (h) Operations

6. Technical Management Processes

   (a) Project Planning

   (b) Project Assessment and Control

   (c) Decision Management

   (d) Risk Management

   (e) Configuration Management

   (f) Information Management

   (g) Measurement

   (h) Quality Assurance

7. Organizational Management

(a) Life Cycle Management

(b) Resource Management

2.4.1. Technical Project Summary and Program Organization. A technical project summary describes the SEMP's purpose and use. The program summary provides an introduction to the project the SEMP is governing. A References section is provided to reference any information used in the SEMP varying from unpublished program references to peer reviewed papers. Regardless of the reference, a clear direction on where it can be found and how it was used is provided [16]. The Program Organization section should be explicit and leave no assumptions in responsibility. This section should provide information on roles, responsibilities, authority, and accountability [16]. Explicitly stating this information provides the program with streamlined management. Additionally, other important personnel such as supporting organizations or key suppliers should be discussed here.

2.4.2. Technical Processes. The technical processes should discuss the systems engineering practices for the program. It is broken into eight subsections: Mission Analysis, Requirements Management, Design Definition and Traceability, Systems Analysis, Implementation, Integration, Verification and Traceability, and Operations. Mission Analysis defines the problem the mission is solving and how. This includes a description of the implementation and the mission ConOps. The Requirements Management section should discuss the stakeholder requirements and decomposition of requirements. It should also specify the control mechanisms for measuring, reporting, and controlling changes for the requirements and assessing the impact of requirement changes [16]. Design Definition and traceability must guide the program through the design process and cover topics such as trade studies and control documents. Systems Analysis should describe an overview of the approach and methods planned for design architecture to satisfy requirements [16]. Implementation should include procurement, prototype fabrication, software coding, final production, and operator training among others. Integration includes what must be integrated into the program to make it successful. For example, support from outside the program and facilities required for the program should be documented in this section [16]. Verification and Traceability describe the scope, strategy, tools, techniques, and responsibilities for verification and validation of requirements [16]. How analysis and simulation results should be interpreted into satisfying requirements should be discussed in this part. Operations determine what training, manuals, procedures, assessments, and facilities are needed during the operations phase of the program. Overall, this section lays the groundwork for the entire life cycle of the project and ensures the program is prepared for every step in this process.

2.4.3. Technical Management Processes. The technical management processes section should describe the program management information necessary to have a successful program. The project planning process determines the scope of the project management and technical activities; identifies process outputs, tasks, deliverables, establishes schedules for task conduct, including achievement criteria, and required resources to accomplish tasks [16]. The purpose of the Project Assessment and Control process is to assess if the plans are aligned and feasible, determine the status of the project, technical and process performance, and direct execution to ensure that the performance is according to plans and schedules, within projected budgets, to satisfy technical objectives [16]. Decision management is meant to provide a structured and analytical framework for evaluating decisions [16]. Decision management supports risk management by outlining how to make risk-informed decisions with proper decision rationale. Refer to Section 2.3 for more on risk management. Configuration management manages and controls system elements and configurations over the life cycle and manages consistency between a product and its associated configuration definition [16]. Information management oversees the documentation process to verify that the program is documenting information. It should validate that documentation is unambiguous, complete, verifiable, consistent, modifiable, traceable, and presentable [16]. The purpose of the measurement process is to have procedures defined for the collection, analysis, and report of data and information [16]. These data and information can be technical, which supports the product or non-technical and assists the management team. Lastly, quality assurances are meant to verify the effective application of the organization's quality management [16].

2.4.4. Organizational Management. Organizational management typically outlines the responsibilities of those outside the program. For example, life cycle management could have policies and procedures imposed by those with authority over the program to stay in compliance. These can come from within the program organization or by external organizations such as the federal government. Life cycle management also includes program assessment and continuous improvement policies [16]. Resource management defines personnel management, such as training, infrastructure, and quality management.

2.5. COMMON CHALLENGES AND PITFALLS OF UNIVERSITY SATELLITE TEAMS

University satellite teams face various challenges in designing, building, testing, and operating satellites. Many of these challenges are different from those encountered in industry. This is due to the lack of resources university teams have compared to professional organizations who have a full-time work force committed to satellite development [17]. Nearly all members of the university teams are students committing part-time effort to the mission while attending the university full-time and perhaps also balancing other extracurricular activities. University teams commonly encounter either technical or systems engineering and program management issues [11]. Technical problems tend to be university-specific because every university team has a different satellite and payload. In the SmallSat industry, empirical evidence suggests that 80% of mission failures are due to poor designs that could have been prevented with effective systems engineering practices [18]. Many of the systems engineering and program management problems are frequent among university teams. The following list details specific issues that are either program management or systems engineering related:

1. Program Management

    - Lack of stable personnel resources [11] [19]
    - Student turnover [11] [6] [20] [21] [22] [19] [17]
    - Lack of university support [11]
    - Lack of internal reviews [11] [23] [24] [25]
    - Lack of program management experience [6] [21] [17]
    - Lack of recruitment ability [20]
    - Lack of multidisciplinary recruiting [20]
    - Poor new member on-boarding [20] [21] [17]
    - Difficulty meeting external safety requirements [20]
    - Lack of faculty oversight [22] [24]
    - Lack of stable funding [11] [6]

2. Systems Engineering

    - Lack of systems engineering knowledge [6] [21]
    - Lack of risk management practices [21]
    - Lack of quality assurance [21] [24]
    - Poor software design and management [11] [21] [19]
    - Poor documentation standards [21] [23] [19] [25] [17]

3. PROPOSED MODIFIED SYSTEMS ENGINEERING AND MANAGEMENT PLAN

Traditional SEMPs are not appropriate for a university satellite team that has multiple missions and other internal teams because traditionally they are developed for individual missions. A university team could create one for each mission they start, but the scope would only apply to that mission and would not address all issues that face a university team as a whole. The proposed modification here of a SEMP has additions that apply to an entire team, regardless of the number of active missions, and ”live” outside the scope of individual missions. Topics such as systems engineering, risk management, and program management are introduced in the SEMP, but the proposed SEMP is not intended to be a complete guide for these topics. Rather the SEMP should serve as an introduction with guidance on how to implement these topics, specific rules, and procedures the team should follow.

3.1. PROPOSED STRUCTURE

The purpose of this SEMP is to establish a reference for managing the team and to provide education on systems engineering and program management principles. After a student reads a university SEMP they should understand how the team is run on a day-to-day basis and learned some systems engineering and program management fundamentals that can be used to support the team.

University SEMP Structure:

1. Introduction
2. Mission Statement
3. Human Resources Team
4. Mission Management

5. Enterprise Software
6. Design Processes
7. Testing Processes
8. Software Management
9. Mission Life Cycles
10. Risk Management Plan
11. Alumni Engagement
12. Managing a Successful Team
13. Conclusion

3.2. CONVENTIONAL VS MODIFIED SEMP

A conventional SEMP provides a plan for implementing the selected system life cycle processes across the full life cycle of the project [16]. The modified SEMP provides a plan for program management of the entire team, guiding individual missions through the system life cycle processes and systems engineering procedures throughout the life cycle of a university team. It does so by taking the conventional SEMP and subtracting sections that change from mission to mission. Sections added cover responsibilities and information that fall outside the scope of the mission and which are used by the program's governing organization. Some sections were modified to tailor to university teams' needs. The following sections explain modifications; Chapter 5 discusses the purpose and implementation of the individual sections.

3.2.1. Sections Removed. One goal of the modified SEMP was to be simple and straightforward to allow students to easily understand the SEMP with no prior knowledge of systems engineering, program management, or the university team. This is meant to encourage inexperienced students to read and review the SEMP. Additionally, the SEMP is meant to provide general guidance to the team, not just for one specific mission. Thus many of the sections that discuss mission-specific information were removed or modified, including mission analysis, implementation, and operations in the technical processes section. The configuration management and measurement sections are also meant for specific missions and were removed.

3.2.2. Sections Added. The first section added was the mission statement for the team. The conventional SEMP has a section on the purpose of the satellite mission, and generally, mission statements are reserved for organizations, not individual missions. The modified SEMP is meant to govern the entire university team, not an individual mission, hence the mission statement section was added.

The enterprise software and software management sections are not explicitly provided in a conventional SEMP. The insertion of these sections serves to address some of the common challenges for university teams. The enterprise software section is meant to serve as a control device to provide a consistent team environment to effectively increase productivity and new member onboarding. The software management section is meant to address common software design and management issues faced by university teams.

The alumni engagement section was added to help address the most common and challenging obstacle for university teams: student turnover. Alumni engagement, if used effectively, can provide a solution to nearly every challenge a team encounters, whether it be program management, systems engineering, or something specific to a mission. Therefore, a specific section in the SEMP was been dedicated to discussing processes and methods to utilize when engaging with alumni.

Another section added specifically for ”defense” against common challenges that face university teams is dedicated to practices, tips, and advice to managing a successful team. This section is tailored to the team with the goal of filling in any gaps left in the SEMP to facilitate the teams’ success.

3.2.3. Sections Modified. Program Organization was modified to be more straightforward by dividing it into two sections. The Human Resources Team and Mission Management sections divide the team into two entities that could be further detailed and replace the program organization section in a conventional SEMP. A conventional SEMP is meant to be applied to an individual entity and has no need for this division. The two new sections explicitly state what each entity's responsibilities are and how they support one another. Other subsections found in a conventional SEMP are divided into these two new sections. For example, project planning, assessment, and control were categorized under mission management, and resource management and life cycle management were categorized under the HR Team. The quality assurance section from a conventional SEMP was integrated into both the mission management and human resources team sections.

Both the design and testing processes sections are more specific than the technical processes section discussed in the conventional SEMP. These two sections are sufficiently broad to provide guidance for all missions regardless of the mission objective. Principles from the requirements management, verification, traceability, and information management sections are discussed here and they provided a streamlined and simplified version dedicated to satellite missions. Additionally, a section dedicated to requirements management was included.

The mission life cycles section's purpose is to educate and guide team members. Many students who join the university satellite teams have little experience with satellite mission life cycles, or any engineering life cycle. Thus, this section replaces many of the technical processes subsections in a conventional SEMP with a discussion on satellite development life cycles.

A section dedicated to the risk management plan is included to introduce risk management procedures and principles. The decision management section was dispersed and merged with risk management, human resources team, and mission management sections to provide a simpler SEMP.

4. PRECURSORS TO CREATING A SYSTEMS ENGINEERING AND MANAGEMENT PLAN

There are some planning tasks and logistics the team should complete prior to the creation of an SEMP. Addressing these action items will more easily allow for the development of a SEMP. Not taking care of these action items prior to the SEMP or prior to the start of a new mission can lead to poorly managed missions and even failed missions.

4.1. RECOMMENDED UNIVERSITY TEAM STRUCTURE

A team should define an organized structure prior to the creation of the SEMP. The organization of the team lays the groundwork for the SEMP and discussion within it. For example, M-SAT has organized its team into two or more entities, one for the Human Resources (HR) Team and another for each mission. The HR team and Principal Investigator oversee and support each mission, as shown in Figure 4.1.

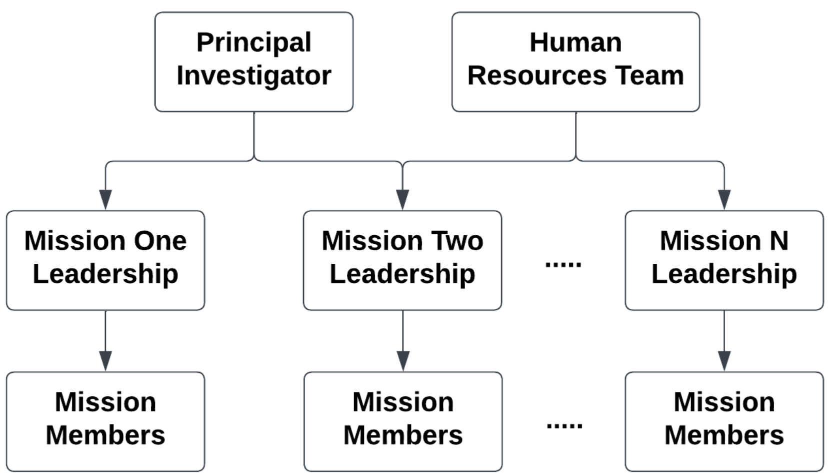

Figure 4.1. University Team Organization Chart

4.1.1. Human Resources Team. M-SAT's HR Team handles the human resources aspects and supports the program management of each mission with the program management. As discussed in Section 2.5, the majority of the challenges the team faced were program management issues. The HR Team is a separate entity from each mission with the purpose of resolving these program management issues. M-SAT calls this team the ”Resource Management Team” and Figure 4.2 outlines the organization of the team. The Student Director oversees this team and assists every manager with their responsibilities. The Team Managers oversee the day-to-day operations such as recruiting, new member onboarding, and more. The Lab Managers are in charge of lab cleanliness and safety and provide training to the team to ensure all members are educated and knowledgeable with lab protocols and skills. The Proposal Manager assists the Principal Investigator with external proposals and assists students with funding opportunities. Finally, the external relations managers handle engagement with non-university organizations such as potential sponsors or partners, alumni, and launch providers. See Section 5.3 for further discussion on how the HR team supports the missions.

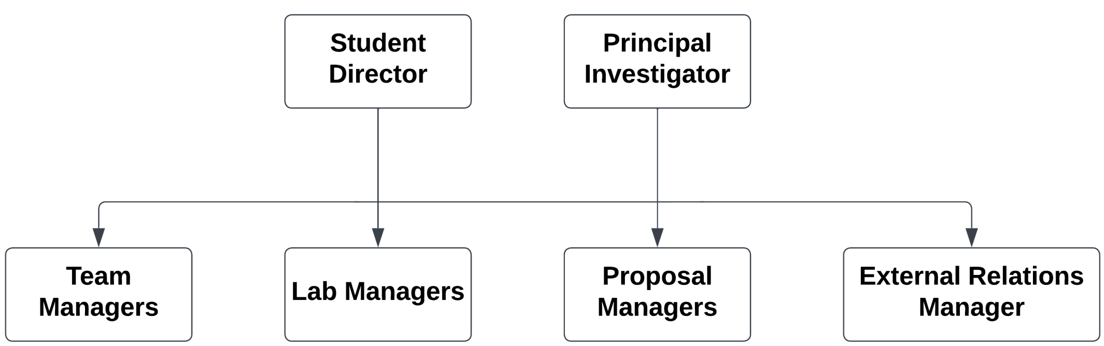

Figure 4.2. Resource Management Organization Chart

4.1.2. Individual Missions. Each mission should have the same structure within a university team; changing structures could confuse members. A set mission organization helps when setting up a mission by providing a predefined model. Figure 4.3 shows the mission organization commonly used in university teams. The chief engineer handles technical oversight of each subsystem including implementing proper systems engineering practices. The program manager handles program management and assists the chief engineer with outstanding tasks with support from the HR Team. The chief engineer and program manager along with the principal investigator(s) manage the subsystem leads who direct their respective subsystems. A common practice in M-SAT is to name co-leads and deputy leads for subsystems and this method has proven to aid in subsystem success.

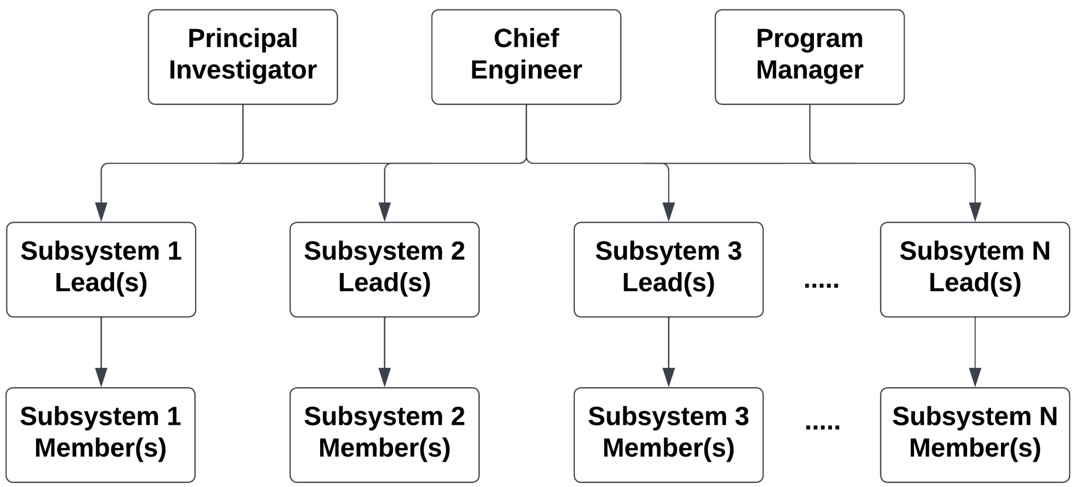

Figure 4.3. Mission Organization Chart

4.2. UNIVERSITY SUPPORT

Pursuing and securing support from the university is a promising resource for student teams to be successful. The university can support a team in multiple ways and the team should work with various departments within the university to determine support levels such as providing dedicated lab space. Refer to Section 4.5 for detailed descriptions of facilities necessary for a university satellite team. Departments can also provide financial support in the form of research or funding for the integration of the curriculum with the team. Refer to Sections 4.3 and 6.1.5 for more on funding sources and integrating coursework with the university team's missions, respectively.

Faculty can provide guidance, mentoring, and technical expertise to the students. Ideally, a team should recruit faculty members to support the team's efforts in every discipline on the team. University teams whose spacecraft reach orbit nearly always had one or two highly motivated faculty members and other staff members supporting the mission [26]. Typically, students on the team join knowing little about satellite design and development. Therefore, mentorship is important; the UNP program has resources and many other resources are online. However, having faculty who directly mentor and supervise the students will enable the students to learn more and faster (in this author's experience). A team should work with the faculty at their university and compile a list of faculty, contact information, and level of commitment for each. Understanding the level of commitment and support will help the team know how much they should depend on that faculty member. This will also ensure the students do not overwhelm a faculty member and potentially lessen the motivation for them to continue their support. Faculty can also be enticed to assist by enabling their support to be used to present research papers at conferences and to publish them in archival journals.

4.3. FUNDING SOURCES

The team should determine the funding sources and document them in the SEMP. Commonly, UNP provides enough funding to fully integrate a satellite, but this is a competitive program and only selected schools receive funding. NASA's CSLI is another program that only provides launch funding, not satellite development funding. There are other miscellaneous research grants from organizations like the National Science Foundation, Department of Energy, NASA, etc. These are not always consistent sources though, so the team always be seeking additional funding. Consistent funding can come from the university, so the team should consider the research and student organization funding the university provides, though this is typically lower amounts. Additionally, senior design students typically receive a modest budget for their classes. This is typically a small amount but can be substantial enough to provide secondary support to the team's budget. Another method of consistent funding is to integrate the satellite development with coursework and labs.

4.4. RECRUITING PLAN

Recruiting a multidisciplinary team is critical for success; the following is a list of common majors that are recruited for M-SAT. Teams should not limit their recruiting to the following majors and should make their own list based on the relevant majors available at their university.

M-SAT's recruitment list:

- Aerospace Engineering
- Computer Engineering
- Electrical Engineering
- Engineering Management (Industrial Engineering)
- Mechanical Engineering
- Computer Science
- Physics
- Business
- Information Science Technology

Once a list is compiled, then when, where, and how to recruit should be planned. At Missouri S&T there is a student organization fair every semester that has proven to be a strong recruiting tool for M-SAT. Alternatively, many professors at Missouri S&T allow teams to give a short recruitment presentation in their classes. M-SAT has even seen success with flyers with a QR code for a survey to get student information. There are many other ways to recruit and every university has unique opportunities to effectively recruit.

4.5. FACILITIES

There are multiple facilities required to build and develop a satellite. At the minimum, a team should have a dedicated lab for satellite development. This lab should include high processing computers, electronic stations, storage, and other tools needed for satellite development. Without these tools or facilities, students will not have the ability to develop satellites. Most teams who successfully launched a satellite also had their own ground station [26]. Other tools and facilities that are helpful but not required include shaker tables for vibration testing, vacuum chambers with or without thermal capabilities, machine shops, and secondary labs. These facilities are not required because they can be outsourced but having access to these facilities can reduce costs and development time.

5. FORMATTING THE SYSTEMS ENGINEERING AND MANAGEMENT PLAN

Each of the proceeding sections explains the format and organization of an SEMP tailored to university satellite teams. Each section explores recommended topics to consider. Of course, it is acceptable to add, modify, or remove sections from this recommended SEMP to tailor it for a specific university team. Appendix A showcases the M-SAT SEMP as an example for other teams. Note that a team should not simply ”copy and paste” the M-SAT SEMP into their own rather it should be used as a reference when creating their own. The M-SAT SEMP is specific to the M-SAT team and its team expertise and dynamics. Other university SEMPs may look similar but will be different because each team's environment is different, and as a result, each team should be managed differently.

5.1. INTRODUCTION

The introduction should, at minimum, include sections explaining the purpose and scope of the SEMP. Additionally, it may be useful to have a change log to track the changes over the course of the document and the authors who have worked on it. Some teams may create the SEMP as a “rule book” while others may use it as a reference. Regardless, it is important that the purpose and scope of the SEMP are explicitly stated so team members know what to expect from the SEMP and how to use it.

5.2. MISSION STATEMENT

A mission statement is a concise statement that communicates the purpose and objectives of an organization, team, or individual. It is a brief but powerful description of the core values and goals that guide the decision-making and actions of the entity in question. The mission statement for the university team should be purposeful and align with sponsors' and stakeholders' goals. Typically, in the university environment, funding is given to student teams to educate students and provide research to improve the current state-of-the-art. The team mission statement should correlate to these purposes. A well-crafted mission statement should provide a clear direction for the organization or team, inspire its members, and help stakeholders understand its purpose and vision. Once a mission statement is created, then the SEMP should be built around achieving that mission.

5.3. HUMAN RESOURCES TEAM

A group commonly missing from a university team is the human resources team. This team allows the mission members to solely focus on the technical aspects of the mission. The human resources team should handle responsibilities such recruiting, new member onboarding, team events, and university politics. Regarding university politics, there should be guidance in the SEMP to assist the human resource team. This section also organizes the human resource team, responsibilities for each position within the team, and it provides sections to assist with the management. Some examples of this include a recruiting plan, new member management, current member management, student funding sources, meeting schedule and itineraries, and website management.

5.4. MISSION MANAGEMENT

Mission management for university satellite teams includes the planning, design, development, testing, and operation of a satellite mission. This section provides a template for mission management. Each mission's structure and management may be different due to the circumstances surrounding each mission. However, consistency between missions should be maximized to identify what does and does not work well. The mission management section should discuss the organization, mission planning and scheduling, internal and external reviews, personnel budget management, design and development procedures, mission operations, and other topics that will provide the mission with the tools to be successful.

5.5. ENTERPRISE SOFTWARE

Enterprise software for university satellite teams can provide a powerful tool for managing the complex processes involved in designing, developing, and operating a satellite mission. Enterprise software is used to manage and organize teams. Examples include Google Drive, Microsoft Teams, Gitlab, and Slack. Enterprise software is used to store files, enable communication, and manage information, among other functions. This section should explain what enterprise software the team uses and how it is used. For example, Gitlab has useful features to streamline the management of software. However, if the team does not have the expertise to use Gitlab, then all features may not be used. This section should explain features or provide links to external references, that allow members to learn how to use necessary Gitlab features. Regardless of the software, this section should identify what software is used, what features are used, and how they are used. This streamlines the management process and allows the team to be efficient when using enterprise software.

5.6. DESIGN PROCESSES

Design processes for university satellite teams involve the systematic planning, conceptualization, and development of the satellite, and they involve a number of steps. Consistency in systems engineering processes and practices such as a design process allow for the team to reflect upon the current processes and improve them over time. Thus, having a defined design process will streamline the design process and provide a guide for more effective documentation. The general process the M-SAT team uses is as follows, although not every design will need to utilize each step. The detailed general process outlined in the M-SAT SEMP guides members through what to consider during the design process.
General Design Process:

1. Determine Purpose and Scope

2. Define System Constraints and Requirements

3. Conduct and Document Trade Study
4. Define System Hardware Identification
5. Define Version Control
6. Create a Technical Report on Design
7. Establish a Changelog Procedure

5.7. TESTING PROCESSES

University satellite teams must follow a rigorous testing process to ensure that their satellite performs as intended when launched into space. Similar to the previous section, having a consistent testing process helps the team identify repeated issues and update the general testing process to eliminate the issue from arising again. A general process the M-SAT team uses is as follows and is expanded on in the M-SAT SEMP.

General Testing Process:

1. Define Purpose and Scope
2. Research and Document the Testing Theory
3. Create and Document a Test Set-Up and Procedure
4. Conduct the Test and Show Repeatability
5. Analyze Data
6. Discussion on Results

Testing Methodology. There are multiple ways to approach and conduct tests regarding satellites. An effective testing methodology M-SAT has adopted is to start simple and add complexity. Starting as simply as possible provides the team with verified baseline results for future comparison. Even if these results are already known or the test seems too simple, it should still be done. Slowly adding complexity allows a more controlled testing environment and yields streamlined problem-solving. This methodology or others can be implemented into this section of the SEMP for the benefit of the team if one wishes to do so.

5.8. REQUIREMENTS MANAGEMENT

Requirements management is a crucial process in systems engineering that involves identifying, documenting, tracing, and tracking the requirements of a project. The first step in requirements management is to clearly define the scope of the system, including its intended use, stakeholders, and constraints. Once the requirements are identified, they must be documented in a management tool such as a Requirements Verification Matrix (RVM). Once the requirements have been documented they should be communicated to the stakeholders to confirm all parties are in agreement. During the project life cycle, the Chief Engineer, or technical lead, is responsible for the management of the requirements. This includes traceability, modifications, validation, and communication of the requirements.

5.9. SOFTWARE MANAGEMENT

Proper software management is crucial for university teams. Effective software management ensures that software systems are developed, tested, and maintained efficiently and effectively. A common issue university teams encounter is proper software management. There are multiple methods to effectively manage software but, it is critical that the adopted method is applied consistently and documented. The first step in software management is selecting a repository-based software, such as Gitlab or Github, intended for software management. This section should also describe what features the team uses and what rules the team follows for maintenance. The SEMP does not necessarily need to explain the features, though it is acceptable to explain them in this section or in Section 5.5, Enterprise Software.

Regarding coding standards, it is recommended to follow an industry standard. such as the Linux Kernel Coding Style. Developing code that is written with the same rules and techniques makes the code readable and helps engineers to be more efficient and able to detect errors faster.

Another topic to be covered in this section is software reviews; peer-reviewed software results in efficient code and lowers the risk of poor code being merged into master code. There are multiple methods for conducting a software review, and some repositories like Gitlab have built-in features that can streamline software reviews. Whatever method or feature is used, this section should discuss the procedure for software reviews to ensure they are conducted with care and consistency.

Testing of the software should be discussed in this section. Researching industry testing methods, such as auto-tests, is recommended to determine how the team should conduct software testing. The complexity of the testing procedures should fit the expertise of the team and may change over time as the team grows technically. See Section 5.7 for a short overview of a testing methodology that can be adopted when testing code.

5.10. RISK MANAGEMENT PLAN

Risk management is the process of identifying risk, determining the acceptance of the risk, and then taking action to mitigate or control those risks. The goal of implementing a Risk Management Plan in the SEMP is to provide an initial start for the university team in risk management. It is recommended that the team have a separate and more complete Risk Management Plan in addition to the SEMP. It will take time, dedication, and technical knowledge to write a complete Risk Management Plan, so it is recommended to dedicate a section of the SEMP to a Risk Management Plan in the interim because risk management is a complex and difficult topic. There are different programs and principles for effective risk management practice. The M-SAT SEMP discusses the application of NASA's Risk Management practices as discussed in Section 2.3.

5.11. ALUMNI ENGAGEMENT

Alumni engagement can be a significant factor in the success of university satellite teams. Alumni can provide knowledge, expertise, networking, collaboration, funding, resources, reputation, and more. Arguably, the most common challenge university teams face is student turnover. One purpose of alumni engagement is to mitigate risks associated with student turnover. The first step in alumni engagement is recording contact information as team members graduate, including personal phone numbers and emails. This provides a link for direct communication with alumni that can be used in numerous ways. Engaging with the team alumni takes little effort but can be rewarding. Alumni who stay up to date on M-SAT benefit from opportunities such as having a convenient method to recruit members for job opportunities. Conversely, side-engaging with the alumni provides the team with support through networking, technical support, and even financial opportunities. This section should discuss methods to consistently engage with alumni; for example, M-SAT has an alumni email listserv that is used to send out newsletters at the end of each semester.

5.12. MISSION LIFE CYCLES

A mission life cycle for a satellite mission is the series of phases that a satellite mission goes through, from the planning and design phase to operations of the mission. Proper planning, management, and execution of each phase is critical for the success of the mission. One purpose of an SEMP is to educate, thus this section provides the team with a description of satellite mission life cycles. This could be a small primer, as seen in the M-SAT SEMP, or a detailed overview of each mission life cycle phase. The level of detail depends on the purpose and scope of the SEMP and other supporting resources the team has available to it on satellite mission life cycles.

5.13. MANAGING A SUCCESSFUL TEAM

Managing a successful satellite team involves various factors that require careful planning and execution. This section serves the purpose of providing advice and suggestions for the leadership team's management strategies. These topics can come from professionals, literature, and current experienced members. The M-SAT SEMP includes topics including motivating students, good practices of effective teams, and patterns in efficient groups. This section is meant to include topics that do not fit in a specific previous section. Therefore, this section was created to fill gaps not addressed by the previous sections of the SEMP.

A recommended subsection is one discussing how to motivate students. The biggest difference between university satellite teams and industry teams is that students typically volunteer part-time while industry workers get paid full-time. Thus, motivating techniques for students need to be established to help guide the leaders of the team. Some students can obtain financial opportunities when working on the team while other students can receive course credit. It is easier for those members to be motivated compared to students who are volunteering their time without immediate reward. Typically most of the team is composed of volunteers, therefore, the leaders should attempt to create an engaging atmosphere. This can be done by making team members feel welcome and inspired by the significance of their contribution both in the context of contribution to the team and to the mission; the atmosphere should encourage their enthusiastic contribution and strengthen a sense of community among team members [25].

6. IMPLEMENTING A SYSTEMS ENGINEERING AND MANAGEMENT PLAN

The best university teams in the space industry employ strict systems engineering practices for their satellite development [6]. Hence, it is important for the entire university team to be aware of and familiar with the team's SEMP. The purpose of the proposed SEMP, was to act as a guide and reference for all students that engage with the team. An SEMP is not effective if the team members are not following the directions and recommendations outlined in it. Appendix A showcases an example of the SEMP M-SAT created and currently used.

6.1. TAILORING THE SYSTEMS ENGINEERING AND MANAGEMENT PLAN

Every university satellite team is different in size, funding, support, and dynamic. As a result, it is expected that every team will develop their own version of an SEMP. The SEMP is a living document, and as the team or external factors change, so should the SEMP. The SEMP should flow with the team and be an updated source for systems engineering and program management practices.

6.1.1. Personnel Management. One key to forming a successful team is recruiting and managing its members. In a UNP lessons-learned paper, they identified ”forces” in successful teams that multiple members need to exhibit [11]. Three of those are motivating forces, momentum, and verification [11]. The motivating force is a member who understands the overall vision and mission scope and can communicate that to the members, thereby motivating them [11]. A group of members will need to be the momentum of the team and accomplish the majority of the work. M-SAT, along with many other university teams [26], has a history of the minority completing the majority of the satellite development. Verification of work by internal reviews is another key factor. Multiple sections, such as Software Management, discuss reviews.

6.1.2. External Support. The team should identify and document methods to secure financial, technical, and programmatic support. A crucial party is the university; without support from the university, the team must focus on sustaining resources rather than satellite development [11]. If the team has full support in developing their satellite(s), they must also consider launch services and who will provide and assist them with this process. Not only are launch services expensive, but it also requires extensive administration support to satisfy all the launch provider requirements. The team should also attempt to engage and collaborate with aerospace businesses to network and receive mentorship. It is not uncommon for universities to partner with government entities or other businesses in the industry to receive support. Commonly, the team will be given a payload to fly on behalf on the external party and the team is in charge of developing a satellite for this payload.

6.1.3. Emphasizing Documentation. A common challenge a university team faces is a lack of well-written and extensive documentation. A commonly forgotten piece of information is documenting the rationale behind decisions [21]. A method to mitigate risks that come from student turnover is providing high-quality documentation. Documentation should be detailed enough that freshmen university students, with little technical knowledge, can understand the information.

6.1.4. Role of a Principal Investigator. The principal investigator (PI) is a faculty member who assumes responsibility for the university team, thus they are one of the most important positions on the team. The PI's responsibilities include conceptualization and development of the research project, funding and budgeting management, supervision of the research team, publication and dissemination of research findings, data analysis and interpretation, and managing external collaborations. Students typically focus on the short-term of the current state of the team and missions but the PI cannot follow this mindset. The PI must adopt the mindset of supporting the entire satellite program rather than individual satellite projects [22].

6.1.5. Integration of Coursework with the Satellite Missions. Since 2021, the M-SAT team had a tremendous team growth of non-aerospace engineers joining the team. This was a result of integrating senior capstone courses with the team. The projects proposed were a mix of solving current issues or redesigning needs and developing systems to use for future missions. Successful teams have a common theme of including a mix of curricular and extracurricular work [26]. Due to seniors joining the team, word of mouth reached younger students and generated interest in joining the team knowing their senior design project could involve their extracurricular work on the team. This was a selling point because many engineering students want to get involved in an extracurricular team. The knowledge that the extracurricular work could eventually turn into curricular work made the M-SAT team more attractive.

6.2. USAGE OF A SEMP

The leadership team should be familiar with the SEMP regarding team and mission management and system engineering practices. The oversight required by the leaders is important for a mission's success and the ability to use the team's SEMP will assist them in the management of the team. Ideally, every member should reference the SEMP to practice proper systems engineering as outlined by the team leaders in the SEMP. Each new member should be tasked with reviewing the team's SEMP. The SEMP should explain when and where it is used so that new members understand how and when to use it.

6.3. MANAGEMENT OF A SEMP

Systems engineering and management plans are essential in ensuring a university team will be successful. They are only effective if they are properly managed, communicated, and updated. It is the HR Team's responsibility to manage and implement the team's SEMP. Whoever is leading the HR Team, for example, the Student Director for M-SAT, should be the sole owner of the SEMP. The dynamic of the team and the space industry could change, thus the SEMP might require updates to provide the team with guidance. The HR Team should monitor new risks, team status, mission progress, and update the SEMP accordingly. An outdated SEMP will lead to poor and improper management of the missions and team.

7. M-SAT LESSONS LEARNED

7.1. SYSTEMS ENGINEERING

M-SAT encountered technical and program management setbacks during the developments of MR and MRS SAT, M^3, and APEX. Nearly all of these drawbacks resulted from immature systems engineering practices. Under proper systems engineering practices, technical issues can either be avoided or streamlined to save time, money, and other factors critical for mission success. Not only will the proper practice of systems engineering improve the chances of mission success, but it also facilitates more learning for the students on the team.

7.1.1. Documentation Practices. The MR and MRS SAT subsystems were documented very well but there were various gaps in subsystems that proved detrimental to the progress of those subsystems. For example, the MR and MRS SAT mission has a very complex guidance, navigation, and control algorithm used to autonomously track a resident space object. This code was originally written in Matlab and duplicated in Simulink so the Simulink model could be deployed as an executable on the flight computer, a Raspberry Pi. The Simulink model was more complex than the Matlab code primarily do to the fact that a socket had to be set up to facilitate communication between the primary flight code executable and the GNC executable. There was minimal documentation on the Simulink model and its implementation.

As a result, there were two consequences. The first was due to there being no documentation, over time the leadership team was aware of the Simulink model, but failed to understand its importance in the big picture. Once the leadership team realized its importance, it took nearly a year for the team to find the tractability between the model and it use for the mission. This required contacting multiple alumni to track down the specific alumni who understood the model and how to implement it, which was difficult. The second consequence was that once the leadership finally understood why the Simulink model was a critical path item, there was no documentation on implementation. Thus, the team had to take time to learn the model and investigate how to implement the model. There were informal testing notes on the model properly working after being deployed on a Raspberry Pi yet because there was no formal documentation the team had to redo the entire test.

Information not commonly documented on the team were trade studies and decision rationale. If a formal trade study took place, it was often not documented, thus the alternatives and reasoning for the decision were lost. Therefore, the team was sometimes in a state of knowing the current design but not why it was designed the way it was. Whenever a design did not function as intended the team had to start from scratch to research alternatives. This resulted in wasted time, where having old trade studies to refer to could have streamlined the new trade study process.

These are two examples of hurdles the team had to encounter due to poor documentation. This is not to say the entire team's documentation was all poor, because it was good overall, it just needed improvement in select areas. The sections where the documentation was poor or non-existent were the limiting factors in the schedule and inhibited critical-path progress. Thus, having successful documentation is similar to successfully flying in space. A 90% grade is not good enough, and sometimes 99% is still not good enough. The 1% missing in documentation can be devastating to the level of causing the entire mission to slip and potentially end a mission for a student team. Thus, a student team must have good and comprehensive documentation and something as small as one test procedure not being documentation can be what determines if the student team can successfully fly a satellite or not.

7.1.2. Coding Practices. M-SAT has had high-quality computer science and engineering students that enabled the software development of the MR and MRS SAT mission. In 2023, the software development for the MR and MRS SAT mission was nearly complete and included over 15,000 lines of code. Gitlab features such as pipelines were used and required technical knowledge to properly maintain. As discussed in Section 7.2.1, the team did not consistently have a variety of majors on the team. As a result, a sufficient number of quality software students was not maintained. This and COVID-19 left an empty software team moving into the fall of 2021. Eventually, the team was able to recruit more software engineering students but that team was left well-behind schedule and with a severe knowledge deficit: a Gitlab repo with submodules, piplines, required development, multiple repos, and minimal documentation on the code and issues.

The leadership team was tasked with relearning the entire MR and MRS SAT software and keeping the members interested and learning. This became an issue as members wanted to join the team and write code, instead of reading someone else's code and attempting to understand and document it. Fortunately, the code was well written so the team could quickly learn, but given the complexity and length of the software, it took a year for one member to understand most of the code.

The MR SAT software is an eight-threaded algorithm with sockets, forks, over twenty classes with only four pages of documentation written in 2021. The MRS SAT code had no documentation. The only way to learn the software was to read code. Having documentation, something as simple as Doxygen-generated documents, would have been helpful to the team. The use of Gitlab was not documented or consistently created confusion for members attempting to understand how M-SAT used Gitlab. Pipelines were used but became mismanaged as pipelines failed and without the knowledge of how pipelines work, they were not fixed immediately. Few repos used the "issues" feature, thus determining future code development was difficult. A properly executed SEMP would have guided the team's software students through effective usage, management, and documentation which could have resulted in efficient software development.

7.1.3. Lack of Frequent Technical Reviews. The lack of technical reviews can have significant negative impacts on the satellite mission. Conducting scheduled technical reviews is an important process that involves a thorough evaluation of the technical aspects of the satellite, and it helps ensure that desired quality and performance standards are met. Technical reviews can identify and resolve technical flaws, low-quality or inefficient designs, and incompatibility between subsystems. The M-SAT team only conducted technical reviews that were required and facilitated by UNP. Student university teams should have more technical reviews than the industry typically requires to mitigate the inexperience of students. These do not necessarily need to be formal, but some fashion of a technical review should be done as a team once a semester and multiple times a semester for each subsystem.

7.2. PROGRAM MANAGEMENT

7.2.1. Lack of Multidisciplinary Recruiting. One of the pitfalls the team encountered was the lack of consistent student membership from a variety of majors. This was a result of not having a human resources team and a lack of focus on recruiting. There was never a lack of aerospace engineering students on the team but, other majors such as computer science, computer engineering, and electrical engineering students were not consistent. Aerospace engineering students were forced to learn new disciplines where the learning curve was high, resulting in those students learning the bare minimum, having low motivation, or having insufficient time to dedicate themselves to learning an entirely different discipline than what their coursework is in. As a result, knowledge was lost and had to be reestablished by the new students who were not able to learn from their predecessors.

7.2.2. Schedule Management. Schedule management is a difficult task for student teams due to student unpredictability and student inexperience [6]. Most students have little experience when it comes to schedules and management of them [6]. Those who do still struggle due to varied student commitment levels. The size of M-SAT constantly fluctuated as members showed interest and then left, and the hours a student would dedicate to the team varied greatly during the semester. As a result, poor schedules were created that were never met. In response to this, a methodology from the book Essentalism is given. Once one has made a schedule, add 50% to the time estimate [27], known as the 50% rule. People nearly always underestimate schedules [27]. A study on college students showed that if one asked them to estimate how long it would take them to write their thesis if everything went wrong and if nothing went wrong they still underestimated their schedule by 10%, and 100% respectively [27].

No matter the phase of the mission, this methodology, or one similar, should be followed. High expectations should be maintained for the team but they should appear in the day-to-day activities and not on a schedule. The 50% rule will account for unpredictable delays which can come from students, delivery delays, or campus delays

7.2.3. Lack of Human Resources Personnel. Prior to the fall of 2021, the team had one resource manager that handled the onboarding of new members. This was the only position or entity separate from missions. After reading the responsibilities of the HR team (Section 5.3) it seems short-sighted that only one member was in charge of all those responsibilities. In reality, the resource manager only sent a new member quick start guide to the prospective members with information on how to join the team and its enterprise software, and all the other responsibilities were not addressed.

Since the M-SAT's implementation of their HR team, the Resource Management Team, the team received more faculty support, collaborated with the local amateur radio club, received more funding for students, increased capstone support from one to five different majors, doubled the team size, and enjoyed consistent membership from other majors, among other positives. The most recent accomplishment of M-SAT's Resource Management Team is the first version of the M-SAT SEMP. This tool empowered the M-SAT team to be successful and fulfill its mission statement.

8. CONCLUSIONS

This thesis proposes a modified SEMP used to manage a university satellite team with the hopes of better enabling other university satellite teams in implementing effective systems engineering and program management practices. The conventional SEMP is discussed and compared to the modified SEMP with the goal to be easier to tailor to a university satellite team. Recommendations on the team structure and other program management aspects are provided to streamline the formation of the team's SEMP. The SEMP can serve as a guidebook for the team, and includes aspects of program management, risk management, project management, systems engineering procedures, and configuration management, among others. Guidance on the implementation of the SEMP is provided to help incorporate the SEMP into the day-to-day activities of the team.

Lessons learned from the M-SAT team with respect to systems engineering and program management are provided to support the need for a SEMP. Additionally, the M-SAT SEMP and other M-SAT resources are provided in the appendices as references to help guide teams in the process of formulating a SEMP.

In summary, a SEMP is an important document for a university satellite team that could solve many challenges, provide streamlined solutions, and increase the productivity of a team. There have been many successful university satellites and each team has approached program management and systems engineering in a variety of ways. Thus, every team should consider what has and has not worked for their team and incorporate those principles into their own SEMP.

APPENDIX A.

M-SAT SYSTEMS ENGINEERING AND MANAGEMENT PLAN

1. INTRODUCTION

1.1. PURPOSE

This System Engineering and Management Plan (SEMP) is specific to the Missouri S&T Satellite Research Team (M-SAT). This document provides extensive guidance on how to run the team and its missions. This SEMP is intended to be a living document that is revised and changed as the team dynamic changes. This document is offered for use by other satellite teams with appropriate tailoring to their program.

1.2. SCOPE

This document provides the M-SAT Leadership team with the knowledge and potential tools to successfully support the team and its missions. This document is not an all-encompassing document for the team nor does it describe how to run a specific mission. It is up to the leadership team to use this SEMP as a guide for applying principles to specific missions. There are many examples for applying said principles but this guide does not provide examples for every scenario the team may encounter.

In addition, every member of the team can benefit from reading this document and understanding how the team is managed. There are systems engineering principles that any member can take from this document and apply themselves without the guidance of the leadership team. All members are encouraged to skim through this document to learn basic principles of systems engineering and program management to support the team and its satellite missions.

1.3. CHANGE LOG

Version	Date	Author	Reason
1.0	1/20/23	Jake Anderson	Initial Commit

2. M-SAT MISSION STATEMENT

To educate engineers of the future through small satellite design, development, integration, testing, and flight.

3. RESOURCE MANAGEMENT TEAM

3.1. PURPOSE

The Resource Management (RM) Team is a separate entity from the missions in M-SAT. The Resource Management team is structured to release burdens otherwise imposed on the mission. Examples of this include recruiting, external relations, new member management, and current member tracking/mentoring, etc. In the following sections, each position within the RM Team is defined, though the RM team is meant to be fluid with multiple managers helping each other. The responsibilities listed below should be treated as recommendations so that managers collaborate to meet the listed responsibilities.

3.2. ORGANIZATION

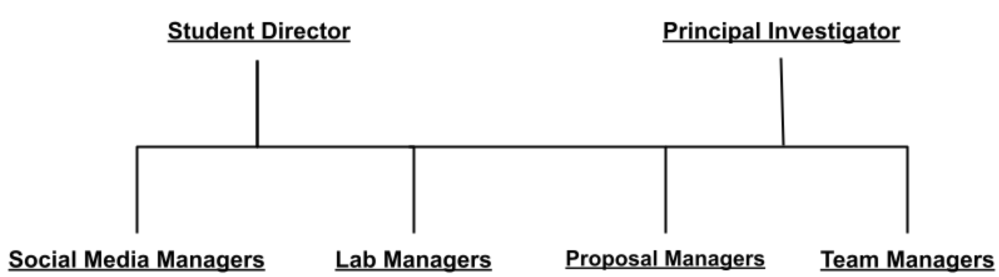

Figure 1. RM Team Organizational (Org) Chart

3.2.1. Principal Investigator. Overview:

The Principal Investigator is a term derived from technical proposals. The Principal Investigator for the M-SAT team and its missions is currently Dr. Pernicka. Dr. Pernicka is the acting advisor for the Team and provides advice and recommendations to the RM team.

3.2.2. Student Director. Overview:

The Student Director oversees the RM team with assistance from the Principal Investigator. However, the Student Director primarily manages the RM Team and should be a highly motivated member who can lead the whole team.

Responsibilities

- Send out and review weekly reports
- Manage Resource Management team
- Help all managers with their responsibilities
- Actively fill vacant management positions
- Support continuous team improvement
- Helps CE/PMs manage members
- Attend and/or present at recruiting events
- Schedule biweekly training
- Sell and manage team gear inventory
- Recruit teams from senior design professors
- Forward to the team student opportunities for campus funding (URS, Work Study, OURE, etc.)

- Actively manage the team's SEMP

Preferred Qualifications

- Leadership/Management experience
- Professional presentation skills

Beginning of Semester Checklist

- ☐ Plan out biweekly training schedule
- ☐ Fill any vacant RM positions

End of Semester Checklist

- ☐ Investigate ways to improve RM team and Mission Teams
- ☐ Send out Senior Design Assessment Form
- ☐ Send out M-SAT Assessment Form

On-Boarding

- Transition M-SAT Resource Account
- Read the M-SAT SEMP

3.2.3. Team Manager. Overview:

The Team Manager primarily handles recruiting and new member management. Additionally, they should be the M-SAT Liaison, if necessary, for the senior design groups

Responsibilities

- Run, manage, and improve the New Member Program (NMP)
- Work with professors to propose senior design projects to senior students
- Plan recruiting events
- Plan team bonding events

Preferred Qualifications

- Outgoing and personable personality
- Recruiting experience
- Proficient with Microsoft Office
- Proficient with LaTeX
- Professional presentation skills

Beginning of Semester Checklist

- [ ] Create a new ”New Member Tracking Sheet” and ESD quiz
- [ ] Create a ”New Member Form” and update the website
- [ ] Send a mass recruiting email to members who previously showed interest but did not continue with the team
- [ ] Work with the mission leads to sponsor/propose projects to senior design groups (EE/-CompE, Comp Sci, and Mech E)

End of Semester Checklist

- ☐ Recruit members for winter/summer break (Econection, Flyers, etc.)

- ☐ Review tracking sheets and discuss the effectiveness of the current NMP

On-Boarding

- Familiarization with New Member Program
- Learn how to set up tracking sheet at the beginning of the semester

3.2.4. Lab Manager. Overview:

The Lab Managers care for the lab and ensure the M-SAT Team is properly educated to safely conduct assembly, integration, and testing in the lab. They specifically handle the ESD training and should provide lab training to the team every semester.

Responsibilities

- Conduct lab training sessions and actively build the current team's training resources
- Keep lab stocked with lab supplies
- Ensure everyone is following lab rules and ESD procedures
- Improve safety and organization of the lab
- Work with subsystem Leads to keep lab organized and clean
- Work with the MAE department to provide lab tours

Preferred Qualifications

- Proficient lab safety and tooling skills (ESD, Soldering, Crimping, Electrical Testing, etc.)

- Proficient cleanliness and organization skills

Beginning of Semester Checklist

- ☐ Work with Student Director to schedule biweekly training sessions
- ☐ Schedule ESD training within first two weeks of each semester

End of Semester Checklist

- ☐ Schedule lab clean-up days and work with subsystem leads for subsystem attendance to help clean the lab

On-Boarding

- Lab Safety and Tooling Training sessions
  1. ESD
  2. Electrical Testing
  3. Crimping
  4. Clean Room
  5. Hand and Paste Soldering
- Lab Tour Training

3.2.5. External Relations Manager. Overview:

The External Relations Manager handles the M-SAT social media, work with alumni, and other miscellaneous responsibilities as assigned. They are the most diverse managers in the RM team and should support the other managers.

Responsibilities

- Keep website updated
- Make bi-weekly posts on social media platforms
- Make monthly Ecconnection posts
- Handle alumni engagement
- Assist in outreach efforts
- Assist Team Managers with recruiting efforts
- Send semesterly newsletters to alumni

Preferred Qualifications

- Public relations experience

Beginning of Semester Checklist

- ☐ Reserve a booth for Minerama/Springorama
- ☐ Update new member form on the website every fall

End of Semester Checklist

- ☐ Send newsletter to alumni

On-Boarding

- Learn how the website is managed
- Get social media log-in information

3.2.6. Proposal Manager.

Overview:
The proposal manager is typically a seasonal role with M-SAT. Proposal managers are only needed while the team is looking for funding opportunities to support new or current missions. Dr. Pernicka will inform the RM team when the team should start preparing for developing proposals and at that time one or more proposal managers may be needed.

Responsibilities

- Research new proposal and funding opportunities
- Work with Dr. Pernicka to write proposals
- Prepare for future proposals

Preferred Qualifications

- Proficient technical writing skills

3.3. RECRUITING PLAN

Below is a list of events or connections from which M-SAT has had a good history of successfully recruiting members!

- Minerama Events (Begining of Fall/Spring)
- Major Specific RSOs
  - Ex. Society of Hispanic Professional Engineers

- Major Specific Societies and Honor Societies
  - Ex. AIAA, ASME, etc.
- Sophomore Design Classes

3.3.1. Professionalism at Recruiting Events. First impressions are important, especially at events where multiple teams are attending. For example, Minerama events have 50+ student organizations in attendance. It is not a competition, but the goal is to stand above all of the other RSOs to recruit quality members. Below is a list of guidelines for recruiting events

Open Forum Events

- Dress in business casual or M-SAT Gear (Match if multiple presenters)
- Have official M-SAT posters
- Bring satellite equipment if possible (Ex. Old Sat Structures, Prop Systems, Solar Panels, etc.)

Presentations

- Dress in Business Casual or M-SAT Gear (Match if multiple presenters)
- Use one of the M-SAT-themed presentations (M-SAT has many recruiting presentations from which to choose)
- If allowed bring props (Ex. Old Sat Structures, Prop Systems, Solar Panels, etc.)

3.4. NEW MEMBER MANAGEMENT

3.4.1. New Member Program. The New Member Program (NMP) is structured to be a completely self-driven program. The purpose of this is to only retain those who are accountable, self-reliant, and motivated. Members are sought who join the team to be committed and able to be trusted to complete an assigned task. The NMP is a short and simple course that will set the new members up for success.

The New Member Program

1. Introduction to M-SAT Video
2. Mission Videos (One per mission)
3. SSE Lab Tour
4. ESD Quiz w/Supplemental Resources
5. Current Leadership Team and Mission Status
6. Introduction to Teams
7. End of NMP Survey

Note: All videos should be less than 15 minutes to make on-boarding as efficient as possible.

3.4.1.1. Introduction to M-SAT Video. This video serves as a sales pitch for the new member to provide them with a high-level overview of team responsibilities, opportunities, and some team history. This should be general enough to be used for recruiting as well.

3.4.1.2. Mission Videos. Each mission has its own mission overview video that explains the ConOps and its subsystems. This should be sufficiently generic so that it can be used throughout the whole mission cycle.

3.4.1.3. SSE Lab Tour. This video covers lab organization, do's and don'ts, storage, cleanliness, and a primer to ESD.

3.4.1.4. ESD Quiz. There will be a recorded presentation and a supplemental document to teach new members ESD practice. A new member will not get lab access until an ESD quiz is passed with 100%. (Note: The supplemental document has every quiz question verbatim)

3.4.1.5. Current Leadership Team and Mission Status. This video will inform new members about who the PM, CE, and Leads are for each mission with details about the current project status of each mission and subsystem. This video will need to be updated on a semester basis.

3.4.1.6. Introduction to Microsoft Teams. This video provides the new members access to Teams where the team communication and documentation storage is provided. It also covers the use of Microsoft Teams including but not limited to Tags, Sharepoint, and the M-SAT Calendar, etc.

3.4.1.7. End of NMP Survey. This survey is for new members to provide feedback on the program with the goal of always striving to be better. The survey also asks what missions and subsystems the new members are interested in, along with their year and major. This information is tracked by the team managers and communicated to the mission leads to help retain new members.

3.5. CURRENT MEMBER MANAGEMENT

3.5.1. Expectation of Members. It is important for the leadership team to have and maintain expectations for the members on the team regarding status. (Volunteer, senior design, paid students, etc.). Expectation awareness makes scheduling more predictable for the leadership team. Both paid members and senior design students must have set expectations prior to joining the team or at the latest one week after joining the team. Changing expectations after already getting paid or doing work for a class generates negativity. To mitigate any risks that may occur to changing expectations without changes to status, it is best to communicate expectations and responsibilities prior to changing members status. For volunteers, the leadership team should expect the minimum from those members because the leadership team has no control over what they may or may not do on a weekly basis. Therefore, the best way to mitigate the risk from lack of control is to lower expectations. However, actions took on the lower expectations should be done carefully. For example, while creating a schedule, the lower expectations should have a substantial effect on scheduling but when assigning members to projects the volunteers should have equal consideration. There is no perfect way to act upon this so the leadership team should work together and manage the members with their best discretion.

3.5.2. Retaining and Motivating Members. There are multiple techniques for retaining members and motivating members to stay on the team and do quality work. It is the Resource Management Team's responsibility to get new members but it is up to the Mission Leadership Team to retain them and give them good reasons to stay.

One thing the leadership team should watch out for are lone wolves. The lone wolf is a team member who ends up working alone. Even if they have the capability and desire to do it alone, do not allow this to happen. Working in a team results in members keeping each other accountable, promotes a team atmosphere, and good relations. If a member is capable of completing a project by themselves, then pairing them with one or more younger members will encourage knowledge transfer. The same principle applies novice members, if a project is assigned to someone who does not have the knowledge or skills to complete it, they will not be motivated to begin the work. As humans, we psychologically desire to work with others, though there are outliers who prefer to work by themselves; however working in groups increase productivity, enjoyment, and morale.

Tips and Tricks

1. Schedule work days and keep members accountable for attendance and invite new members to come

2. Be proactive during subsystem meetings and ask if anyone would like to join a specific project

3. Assign groups for projects. It is fine if people form their own groups, but if members do not take the initiative to form a group, subsystem leads should form teams.

4. If a new or current member is not involved much throw them into a project with others. Typically new members are unsure about where and how to start so it is up to the subsystem leads to guide them through the process of finding something they enjoy!

3.5.3. Member Education. Continuing education is the concept that those out of the educational system are still learning and educating themselves to maintain the state of the art knowledge and technology. It would not be appropriate to use that term here because all team members are students and are continuously learning via course work. However, the team encourages members to create, give, and attend training events, such as the bi-weekly training hosted by the RM team, especially outside of their expertise on the team. This serves to benefit the members and the team. Given that, subsystem leads should proactively give, or delegate, a training session at least one a semester to give the opportunity for all team members to attend and gain knowledge on that particular topic. These training sessions should additionally be recorded and upload to the team's file storage and YouTube for future members view. All team members should learn and gain industry applicable knowledge as much as they can.

3.5.4. Bi-Weekly Training. The Student Director is in charge of working with team members and outside personnel to provide technical training. These meetings shall be at the same time and place of the general meetings on the off weeks (general meetings are biweekly).

3.5.5. Subsystem Training. At the time of writing the initial version of this SEMP, a subsystem training program was not started. It is outside the scope of this document to cover subsystem training but once the subsystem training program has been created this section should serve as a guide for managing and updating the subsystem training.

3.6. WEBSITE MANAGEMENT

The external relations managers is in charge of the website and will ensure it is up to date multiple times per semester. The website should cover the team's current missions, have links to other M-SAT media (YouTube, LinkedIn, etc.), and have a New Member form for new members to fill out, at the minimum.

3.7. SENIOR DESIGN GROUPS

There are many senior design groups the team works with. See below for a list of current majors and professors with whom we work. As always, the team is continuing its efforts to recruit diverse team members and plans to add in more groups in the future.

Current Senior Design Groups:

- Aerospace Engineering - Dr. Pernicka
- Computer Science - Dr. Gosnell
- Computer Engineering - Dr. Woodley
- Electrical Engineering - Dr. Woodley
- Mechnical Engineering - Dr. Midha

The RM Team is in charge of working with the professors and proposing projects to the senior design groups. The RM Team shall also be the primary point of contact to lesson the burden on the missions leadership team. At the end of every semester the RM team shall send our a survey to the senior design team to get feedback on the process and on what M-SAT can improve upon to facilitate the senior design teams success.

3.8. STUDENT FUNDING OPPORTUNITIES

3.8.0.1. Internal Funding Opportunities. Missouri S&T currently provides multiple programs to encourage undergraduate research and because M-SAT is a research team, NOT a design team, our members reap those benefits. Below are common opportunities our members enjoy but this is not an exhaustive list.
Missouri S&T Opportunities

- OURE
- OURE Scholars
- URS
- Work Study

3.8.0.2. External Funding Opportunities. External Opportunities generally fund the projects and specific students research. There are countless funding opportunities that are outside the scope of this document. Though most of the funding comes from the Air Force Research Labratory's (AFRL) University Satellite Program (UNP) and from NASA's Undergraduate Student Initiative Program (USIP) and CubeSat Launch Initiative (CSLI).

3.9. TEAM AND MISSION MEETINGS

3.9.1. General Meetings. There are bi-weekly general meetings throughout the entire semester. In the fall, semester the general meetings shall start the second week of school, so that recruiting can occur the first week. In the spring semester the general meetings shall start the first week of school, for the reason that the team is organized and continuing on projects from the previous semester. This encourages the team to get the ”ball rolling” the first week of the spring semester.

General Meeting Itinerary:

1. RM updates
2. Mission updates
3. Dr. Pernicka updates
4. New Member QR code
5. Team Chant!

3.9.1.1. General Meeting Guidance. M-SAT general meetings have two goals. One is to provide a venue for new members to learn about the team as well as engage with the team. The second is to educate the team on the current status of the mission, not just to purely provide updates.

In the past, general meetings felt uninformative and unnecessary resulting in poor team attendance. After reflection, the leadership team decided to revamp the approach to the general meeting. The leadership team provides updates at the general meeting including traceability and current hurdles the project is facing. Examples of providing traceability are below. Note that full traceability should not be established in one general meeting, rather throughout the semester traceability is added to the projects to provide the team with primers on each project. This is done because if full traceability was added at the first general meeting, subsequent meetings there would not be as useful. Additionally, if full traceability was discussed for each project, it is likely the general meeting would take a couple hours. It will be up to the leadership team's discretion on how to implement traceability.

A non-comprehensive list of talking points to provide traceability includes:

- Explaining dependencies of the project
- Exploring the project's critical path items

- Explaining why the project is important
- Describing what the project entails
- Outlining what disciplines fit the project well
- Scheduling of the project
- Disclosing challenges facing the project
- Explaining what the project accomplishes/proves
- Exploring what the project could potentially alter
- Assessing if the projects involve subsystem collaboration
- Informing if the projects need more manpower

3.9.2. Subsystem Meetings. Each semester the weekly subsystem meetings should start after the first general meeting; this can be complicated at the beginning of both semesters, but more so in the fall. Starting these meetings as soon as possible gets the ”ball gets rolling” more quickly. Using www.when2meet.com and Microsoft Teams Scheduling Assistant(if team members have updated calendars) are handy when scheduling meetings.

Subsystem Meeting Itinerary:

1. If new members are present, introductions should be made (at minimum Name, major, and current project working on)
2. Subsystem Lead gives individual update
3. Each group/project gives their update
    - This is the time for discussion about each project. Subsystem members will ask questions concerning their project and the leads will ask questions ensuring accountability of subsystem members.

4. Subsystem Lead, PM, or CE will discuss any new and unassigned projects as well as emphasize additional help needed on any current projects.

5. Subsystem Lead, PM, or CE shall ensure each group working on projects understands their deadlines and emphasize the overall mission consequences if the deliverables are not met within that time frame.

6. Team Chant!

3.9.3. Mission Meetings. Mission meetings will be held at the discretion of the current leadership team. There are many cases where mission meetings are useful, but that will depend on the situation and time of year.

Mission Meetings are useful and recommended in the following situations:

- Beginning of the semester before subsystem meetings start
- Before a mission review
- End of the semester to update the team on progress made and where resources need to be directed moving forward
- Start or conclusion of a mission
- Start or conclusion of a multi-subsystem project (e.g. DiTL Testing)
- During the proposal phases
- Early design phases

4. MISSION MANAGEMENT

4.1. ORGANIZATION

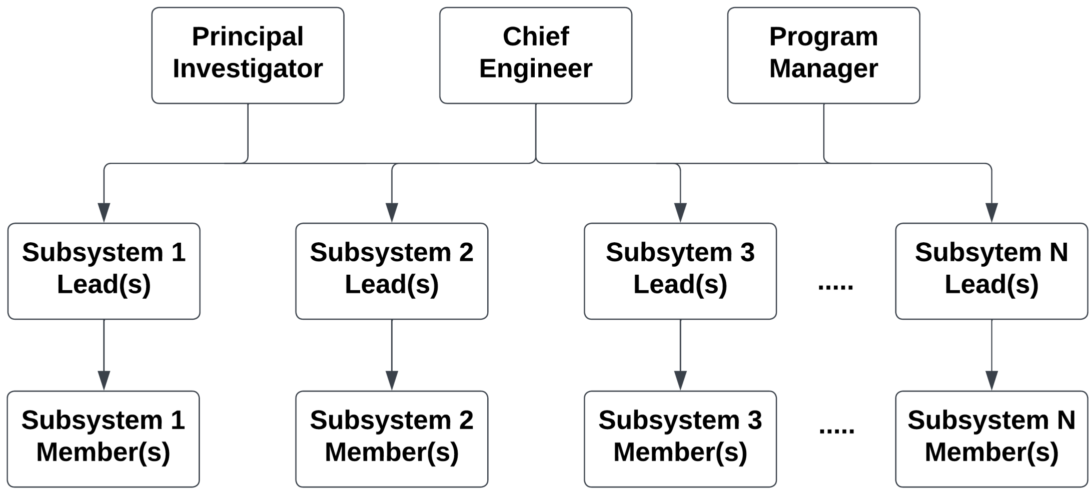

Figure 2. Mission Org Chart

4.1.1. Chief Engineer. Responsibilities

- Be the team's technical expert on the mission
- Maintenance of the RVM including updates, documentation, and assigned projects that will satisfy RVM requirements
- Continuous assessment and management of the mission schedule
- Manage the subsystem lead's goals, progress, and assist when challenges are presented
- Identify risks and respond to them appropriately; refer to Risk Management for more information
- Support the team in preparation for mission reviews
- Should be fully engaged (technically) in at least one subsystem

Preferred Qualifications

- Prior M-SAT Lead Experience (1+ Years)
- Able to commit 14+ hours a week
- Determination and Self-Initiative
- Multidisciplinary/Systems Engineering Knowledge

Beginning of Semester Checklist

- ☐ Work with System Leads to make project documents
- ☐ Work with PM to fill vacant lead positions
- ☐ Work with PM to review and update the schedule
- ☐ Conduct/Update hardware inventory

End of Semester Checklist

- ☐ Documentation. *"The Only Difference Between Screwing Around and Science Is Writing It Down"* - Someone with a satellite in space.
- ☐ If you are one semester from graduating then start working to find a deputy CE.
- ☐ Review the mission schedule and reflect on the past semester's work and how to keep progressing

On-boarding

- Get a notebook for M-SAT notes; it is impossible for you to remember everything you are going to learn in the following months and makes your life much easier to write things down and reference them
- Read the 100s of every subsystem
- Be able to recite the CONOPS and mission statement/objectives

- Know the phase of the mission and critical path items to progress to the next phase
- Review previous mission reviews and know when the next one is
- Learn the mission schedule
- Get owner access to Microsoft Teams and Gitlab

4.1.2. Program Manager. Responsibilities

- Maintenance of the mission schedule, ensure each subsystem stays on track, and update the schedule if necessary
- Create general meeting slides and present at general meetings
- Create slides for meetings with sponsors (e.g. AFRL Biweekly Tagup slides)
- Work with each subsystem to schedule weekly meetings and ensure leads are scheduling one weekly work day (at minimum) per week
- Oversee paperwork the mission must submit
- Manage the mission personnel budget
- Work with CE to order components
- Identify risks and respond appropriately, see Risk Management for more information
- Work with Team Managers to onboard and retain new members
- Assist RM team with presentations
- Most important: be proactive, not reactive. Anticipate team needs and preemptively address them!

Preferred Qualifications

- Previous experience in the respective mission

- Able to commit 7+ hours a week
- Any Resource Management leadership experience
- Determination and Self-Initiative
- Project Management Experience
- Microsoft Office Apps Experience (e.g. Excel)
- Proficient with Latex

Beginning of Semester Checklist

- ☐ Organize subsystem meeting schedule
- ☐ Work with CE to fill vacant lead positions
- ☐ Work with CE to review and update the schedule
- ☐ Start updating personnel budget

End of Semester Checklist

- ☐ Directly ask every subsystem lead if they plan on continuing as lead or want help the proceeding semester
- ☐ Organize winter/summer groups
- ☐ Reconcile personnel budget and get final positions and contact information

On-boarding

- Get a notebook for M-SAT notes; it is impossible for you to remember everything you are going to learn in the following months and makes your life much easier to write things down and reference them.
- Read the 100s of every subsystem
- Be able to recite the CONOPS and mission statement/objectives

- Know the phase of the mission and critical path items to progress to the next phase
- Review previous mission reviews and know when the next one is
- Learn the mission schedule
- Get owner access to Microsoft Teams and Gitlab

4.1.3. Subsystem Leads. Responsibilities

- Plan work days for your subsystem (feel free to have multiple work days for each project)
- Work with Team Managers to onboard and retain new members
- Run subsystem meetings
- Delegate projects/work to subsystem members
- Assist subsystem members with their projects and ensure they have the needed material, personnel support, and resources to safely complete the project
- Ensure subsystem members are documenting their work during the project and at completion of a project

Preferred Qualifications

- Previous experience in the respective subsystem
- Able to commit 5+ hours a week
- Any leadership experience
- Determination and Self-Initiative

Beginning of Semester Checklist

- ☐ Review subsystem schedule and ensure timelines are realistic, and address risks in the schedule
- ☐ Ensure Project Board is updated
- ☐ Create project documents for complex projects to support the project board
- ☐ Determine if a deputy lead needs to be chosen for this semester. A deputy lead is appointed if the lead position will be vacant the semester following the current one. It is good practice to always have a deputy lead in place, as they can help lower your workload.

End of Semester Checklist

- ☐ Make sure all projects are documented, even if not completed. The team can not assume any member will come back and continue the project.
- ☐ Review schedule and work with the PM/CE to update if necessary

On-boarding

- Get a notebook for M-SAT notes; it is impossible for you to remember everything you are going to learn in the following months and makes your life much easier to write things down and reference them.

- Review every document in your subsystem folder over the course of the next month. 100s first then the previous lead can guide you from there

- Know the phase of the mission and critical path items to reach the next phase

- Review previous mission reviews and know when the next one is

- Current and incoming leads must meet on a biweekly basis during the on-boarding period

- Add incoming lead to mission leads channel in Microsoft Teams
- Give incoming lead Gitlab owner access (if the subsystem uses Gitlab)
- Familiarize yourself with the Project Schedule and RVM regarding your subsystem
- Get to know all the current members in that subsystem and all the other leads
- Read previous reviews to understand the development of the subsystem throughout the project and its current status
- Get to know relevant faculty/alumni contacts for your subsystem (e.g. Dr. Kimball for Power or Dr. Anklesaria for Structures)

4.2. APPLICATION PROCESS

1. Send out a Microsoft Form for applicants to fill out
2. Once the form is closed, there will be a meeting including Dr. P, CE, PM, and any leads interested in attending to determine who to choose. Ideally, a consensus decision should be made, otherwise majority rules.
3. Host interviews if a tie occurs or if the deciding group decides this is the best course of action

4.3. MEETING SCHEDULE

See Section 3.9 on occurrence and itinerary of mission meetings.

4.4. PERSONNEL BUDGETS

The personnel budgets will be a vital resource and become more useful as the mission progresses. The personnel budget should be organized by semester and then by subsystem. Within each subsystem, the members should be tracked with the following information:

- Full Name
- Major
- Year
- School Email
- Personal Email
- Phone Number
- Project Keywords

The goal is for each mission to track the members that worked on the mission. If a member is on multiple subsystems, track them through each subsystem. First, this gives the team a head count to record the size of subsystems and the overall mission team to determine if either may be too small or too big. Additionally, this will provide references for members in the future to have the ability to talk to alumni and know the correct alumni to contact.

4.5. MISSION PLANNING

Mission planning is difficult in the industry, it is even more challenging for this team. It is difficult to accurately predict the number of team members we will have and how many hours per week each member will devote. This makes it challenging to create schedules but this is not to say it is impossible. Thus, critical thinking is important when creating a schedule for the mission.

In response to this, a methodology from the book *Essentialism: The Disciplined Pursuit of Less* is given. Once one has made a schedule, "add 50% to the time estimate," known as the 50% rule. People nearly always underestimate schedules. A simple and time-efficient way to combat this is the 50% rule. No matter the phase of the mission this methodology should be followed. It is much better to overperform than underperform. It is important to have high expectations for the team but, it should appear in the day-to-day activities and not on a schedule. The 50% rule will account for unpredictable delays whether the cause is from students, deliveries, campus, etc.

4.6. MISSION DESIGN REVIEWS PLANNING

Mission Design Reviews can be daunting because professionals are criticizing students' work. It is important to understand that without these design reviews, many more university missions would fail. Professionals have more experience and in turn have had more mistakes to learn from - so use that to the team's advantage. The best way to do so is to ensure the team knows the intricate details of the mission. With this, the team can answer questions to the fullest extent possible, allowing better constructive criticism from the professionals. If the team can not answer questions about the satellite and upcoming plans, then the professional will not be able to provide the best feedback.

For instance, a professional could pose a question about why there are two Raspberry Pi's on the satellite and why they communicate over UART. If you do not know, the professional could recommend getting rid of one and putting all the software on one Raspberry Pi. Where if a student was able to say one Raspberry Pi does not have sufficient RAM to execute the software, and UART is used to communicate pictures between the two, then the professional could provide such feedback such as "UART is a simple protocol but slow so if time is a constraint then communicating over a Ethernet cord and protocol may be a good alternative." This is a very specific case, but hopefully this illustrates the need to be knowledgeable of the subsystem.

Design Reviews are also fantastic experiences for students to present and gain experience in a more stressful environment than they are accustomed to. Experiencing stress often helps team members manage stressful situations and hone skills in those environments. However, it is important, especially for students who struggle with presenting, to practice presentations the week before a review. It is highly recommended that the leadership team, and whoever else may be presenting, practice presenting the mission review slides to a group prior to the actual review.

4.7. INTERNAL REVIEWS

It is critical to a mission's success that there are internal reviews of design choices, testing results, trade studies, budgets, etc. These reviews can prevent design flaws and enable better designs even if there are no technical flaws. Attendance and feedback for these reviews are critical factors. It is important for professors, alumni, or anyone else who can provide constructive criticism and valuable feedback to attend. It is also important for subsystem members and general team members to attend these reviews as part of the learning process. These reviews can be very informative and helpful for those who are presenting and attending. It will be up to the mission leadership to schedule these and ensure attendance is high.

5. ENTERPRISE SOFTWARE

5.1. WHAT IS ENTERPRISE SOFTWARE

Enterprise Software is software used to assist with the management and organization of the team. For example, M-SAT currently uses Microsoft Teams and Gitlab as its Enterprise Software.

5.2. MICROSOFT TEAMS

Microsoft Teams is used for daily communication, team announcements, team calendar, and storage.

How to use channels:

- General Channel
  - The General Channel's "posts" are used for team-wide communication
  - The General Channel's "calendar" is used for general meetings, subsystem meetings and work days, and other team or subsystem events
  - The General Channel's "files" are used to store team resources such as training videos and documents.
- Mission Channels
  - The Mission Channel's "posts" are used for mission announcements and planning.
  - The Mission Channel's "files" are used to store all mission documents.
- Subsystem Channels
  - The Subsystem Channel's "posts" are used for subsystem announcements and planning.
  - The Subsystem Channel's "files" are not to be used for any official documentation. Temporary working files are acceptable, however.
- Resource Management Channel
  - The RM Team "files" are used to store RM Team Materials as well as general team materials and references.
- Mission Leads Channel

  - The Mission Leads Channel is a private channel for CEs, PMs, Student Directors, and Subsystem Leads.

- Leadership Channel
  - The Leadership Channel is a private channel for CEs, PMs, Student Directors, and any vital management position for the team.

- Senior Design Channels
  - Each senior design group (Ex. AE, MechE, etc.) should have its own channel to use at its own discretion. This will provide an environment for the senior design groups to directly interact with the RM Team and other team members who support them.
  - The ”files” tab is utilized so the M-SAT Team has complete access to their work which eliminates the risk of lost work because M-SAT never received it.

5.3. GITLAB

Gitlab is used to store the teams software. See 9.2 for more on Gitlab usage.

6. DESIGN PROCESSES

6.1. REQUIREMENTS VERIFICATION MATRIX

The Requirements Verification Matrix (RVM) should be finished, reviewed, and approved before moving into any design process. The RVM drives the requirements and constraints that the design processes shall follow and enables traceability of the designs. If the RVM is properly managed and the design processes are properly documented, traceability is easy to follow and allows streamlined redesigns if the need should arise.

6.2. GENERAL DESIGN PROCESS

It is not necessary to provide explicit direction on every type of design choice and process because it will vary mission-to-mission. It is outside the scope of this document to cover mission-specific design processes. In place of this a general procedure is defined. Not all the sections discussed here will apply to the specific design process, so the team should use their best judgment when creating design documentation. The design documentation should fully follow and describe the design process implemented. The current leadership team is the authority for verifying that the design process and documentation are acceptable.

6.2.1. Purpose and Scope. One must first understand the design purpose. There are cases when a design process is conducted because there is something fundamentally wrong, but other times it may be used to bridge a gap between subsystems. Though the purpose is not limited to this, it is important to completely understand the purpose.

The scope of this process determines how detailed or complex the design process needs to be. This is best explained with an example. The scope of designing a circuit board can vary. The scope of the design process can include a power analysis of the board to fit the current power budget or a power analysis could fall outside the scope of the design and another team (perhaps the Power subsystem) needs to fit this power budget to your design. This is a simple example that illustrates the importance of knowing the scope of the design process.

6.2.2. System Constraints. Before any design research is conducted it is critical to know and understand the constraints imposed on the design. Some missions or subsystems may be more strict than others but there will always be constraints.

Examples of Constraints:

- Mass
- Size
- Cost

- Accuracy
- Dependability
- Current Team Experience/Expertise
- Team capability of designing circuit boards (COTS vs Custom Made)
- Subsystem Budgets
  - Propellant
  - Power
  - Data
  - Volume
  - Mass
- Complexity needed
- Other subsystem dependencies
  - Thermal Constraints
  - Power Limitations
  - Data Storage Limitations
  - AIT Considerations

Note: This is not an exhaustive list!

6.2.3. Trade Study. A trade study should be conducted after the purpose, scope, and constraints are identified. This documentation is typically separate from the final design documentation though both should exist. See Section 10.2.1 for details regarding trades studies.

6.2.4. Define System Hardware Identification. Some complex hardware systems need identification schemes to identify hardware. For example, part numbers for a structure or a specific naming scheme for thermal sensors.

6.2.5. Define Version Control. Some processes may need version control which can come in many forms. Release versions for software and revision numbers for hardware are examples of version control.

6.2.6. Technical Report on Design. Once the design is fully realized and all constraints met, a technical report will be written to describe the design. Format may vary, but it should provide everything from the theory to the application of the design. For a student team, it is important to include the theory for multiple reasons. It forces the designer to fully understand the theory and gives confidence in the design, it allows streamlined learning process for new members, and it provides a reference for new missions looking back on older documentation.

6.2.7. Change-log. It is common for design processes to be updated throughout the mission, and in some cases, it is advantageous to have a change-log so members can understand the history of a design. This enables them to avoid mistakes and learn from others' mistakes.

7. TESTING PROCESSES

7.1. GENERAL TESTING PROCESS

Like the design process, it is not necessary to provide explicit direction for every testing process. A general procedure is defined here, so not all the sections discussed below may apply to the specific testing process. Use your best judgement when creating design documentation. The testing documentation should fully follow and describe the testing process undertaken. The current leadership team is in charge of verifying the design process and that documentation is acceptable.

7.1.1. Purpose and Scope. One must first understand the testing purpose and scope. Typically, testing is to confirm that a design works in order to satisfy RVM requirements. Regardless of the reason, the first step of the testing process is to determine the need for testing and what requirements are being tested for. One should not test things just to test; prior to writing a procedure, what is going to be tested and to what extent it should be tested should be determined. The scope of the test determines the level of detail the test results need to confirm.

7.1.2. Testing Theory. Once the purpose and scope are understood, the intent and theory of the test will be carefully considered and documented. This part includes how and why the test will work and any analysis that will be done with post-test data. How to form conclusions from the analysis will be determined also.

7.1.3. Testing Set-Up and Procedure. A testing set-up and procedure need to be documented to the best of the team's ability prior to the actual test. The setup and procedure should be detailed and include notes on uncertainties or aspects to be considered before and during the test. The team members will consider the equipment, software, and facilities needed during this process. Additionally, safety of personnel and testing equipment/facilities are to be considered. If necessary, diagrams, flow charts, and other picture-aided materials will be added to make the testing setup and procedure as effective as possible. After the test, the team members who conducted the test may need to update and improve upon the test setup and procedure.

7.2. TESTING METHODOLOGY

An effective methodology used in testing is to start simple and add complexity. Starting as simple as possible allows the team to get verified baseline results. Even if these results are already known or the test seems too easy, it should still be done. Slowly adding complexity allows a more controlled testing environment and yields easier problem-solving.

For instance, leak testing can be difficult because propulsion systems and assemblies can be complex. One could assemble the whole propulsion system and then leak test it. Hopefully, it is leakproof, but if not, isolating the leak location can be time-consuming and require complicated procedures. A proactive approach is to assemble the propulsion system in small segments and conduct short-term leak tests in between to determine leaks at each segment or subassembly. This approach results in locating leaks faster because the team knows exactly which segments were added since the previous test. Although initially it takes more time, it can ultimately save time.

7.2.1. Analysis and Testing Results. The analysis and testing results are critical parts of the testing process. Analyzing the test and data to form conclusions will be a thoughtful process. Not only should the team members analyze the data, but they will carefully consider if the results suggested from the test and analyses make sense. All analyses are to be checked by more than one member to ensure quality and accurate analyses are being conducted.

7.2.2. Documentation Organization. The list below should be used as a template to structure the testing documentation. This is not an exhaustive list nor will every testing document need everything in this list, so the leadership team should provide guidance when making or updating testing documentation.

The introduction serves as a summary of the test. The purpose gives high-level goals for the test. The scope covers why the test is being conducted. Testing requirements reference any official or unofficial requirements that are being tested, for example, RVM requirements. The criteria for pass or fail explicitly states what constitutes a pass or fail based on the testing analysis and results. Applicable documents state internal and external references used for this test. Safety compliance and hazardous operations should cover personnel, equipment, or material risks associated with the test and mitigation procedures for the risk. Quality assurance should cover procedures taken to ensure the test is being conducted correctly. Refer to 10.2.3 for more on quality assurance/engineering.

The setup covers everything needed to conduct the test and anything that is optional for the testing environment. The testing procedure should be detailed and divided into multiple procedures if deemed necessary. It is common to have multiple procedures for multiple tests in one testing document. This is due to the testing methodology discussed in 5.7, so it is acceptable to have a series of tests leading to the main test that satisfies the testing criteria.

The Testing Theory is an optional section. Some tests are simple enough such that a theory section is not needed. For example, a propulsion system leak testing procedure typically does not need a theory section because of its simplistic nature. However, characterizing thrusters for a propulsion system using a pendulum setup would clearly require a theory section.

Testing results and analysis should explicitly state, at minimum, the expected results, post-processing of the data, and consequences of the test results. The conclusion should summarize the results and state the success (or lack thereof) of the test. It may be necessary to have subsections on future work needed or requirements that were satisfied in the RVM.

- Introduction
  - Purpose
  - Scope
  - Testing Requirements
  - Pass/Fail Criteria
  - Applicable Documents
  - Safety Compliance
  - Hazardous Operations
  - Quality Assurance
- Set-up

  - Required Facilities
  - Equipment Needed
  - Materials Needed
  - Personnel Requirements
- Testing Procedure
- Testing Theory
- Testing Analysis and Results
- Conclusion
  - Future Work
  - RVM Satisfaction

8. REQUIREMENTS MANAGEMENT

This section outlines how M-SAT will create a Requirements Verification Matrix (RVM) and manage it. First, the team should use the M-SAT template for the RVM found in TRS103. Directions in the template lay out how the team should formulate and decompose requirements. Once the RVM has been established, it should be provided to the sponsors and other stakeholders to confirm agreement among all parties. The template also provides direction for traceability and validation. The Chief Engineer for the mission holds responsibility for the management of the requirements. This includes traceability, modifications, validation, and communication of the requirements.

9. SOFTWARE MANAGEMENT

9.1. CODING STANDARDS

9.1.1. Indentation. A common conflict between developers in the programming field is tabs vs spaces. A middle ground many find is to set the tab key to insert four spaces, and is what M-SAT has adopted. For simplicity's sake, only one statement is placed on each line.

9.1.2. Line Length. Each line ideally should be under 80 characters. Sometimes comments can go over this limit (the exception being function documentation), and sometimes macros affect readability. If a statement goes over the limit, then it will continue onto the next line. If it has parameters being passed to a function, it will have the same indention as the opening parenthesis of the function to which it is passed. An example:

```
// --------------------------- This is 80 characters
    ----------------------- |
    printf("Hello world!\n");
    reallyLongFunctionName(theLongFirstParameter,
        aSecondLongVariable,
                           theWrappedThirdParameter);
```

9.1.3. Brackets and Spaces. Given industry standards, braces are to be placed last on the line, and the closing brace should be by itself, similar to K&R format except that this includes function definitions. The only exception is for do-while statements, where the closing bracket is followed by the `while`. Obviously, do not leave whitespace at the end of lines. It is optional to include the braces if whatever would be in the block is just one statement, just as long as it is within the same conditional if-else chain (either it is all with braces or without).

When dealing with pointers there is no space after the asterisk ('*') character:

```
int *ptr;
```

There should be no spaces between the keyword and parentheses:

```
if for switch case do while
```

Binary and ternary should have a space before and after them:

```
= + - < > * / % | & ^ <= >= == != ? :
```

There is no space after unary operators (`sizeof` is treated like a function):

```
& * + - ~ ! sizeof typeof alignof __attribute__ defined
```

There is no space before or after increment & decrement operators:

```
++ --
```

```
#include <stdio.h>
#include <stdbool.h>

int main() {
    bool isEnabled = true;
    int counter = 0;

    printf("The the variable enabled? ");
    if(isEnabled) {
        printf("yes");
        counter = 5;
    } else {
        printf("no");
        coutner = -5;
```

```
        }
        printf("\n");

        if(counter > 0)
            printf("The counter is greater than zero.\n");
        else if(counter < 0)
            printf("The counter is less than zero.\n");
        else
            printf("The counter is zero.\n");

        do {
            counter++;
        } while(counter < 10);

    }
```

9.1.4. Naming. It is better to be verbose in order to have highly descriptive variable names. The team would rather have code that is self-explanatory and have to type a few more characters than have code that looks like a math proof with just variables called `x` and `m`. The only exception is for variables in `for` loops, which then it is acceptable to use letters (usually starting with `i` and proceeding from there). Global constants must be `ALL_CAPS` and words are spaced using underscores, otherwise `camelCase` is used with the first letter lowercase. Function names follow the same conversions. In `C++`, custom object types follow `CamelCase` rules, except the first letter is always capitalized, and variables of that type follow conventional variable rules.

File names for C/C++ source files should follow the same convention as variables (first letter is not capitalized). Objects in C++ should be alone in files (each object has its own .cpp and .hpp file) with the first letter capitalized. Source files for C should have a .c extension and header files with a .h extension. For C++, source files have a .cpp extension and header files have a .hpp extension. This is so that it is easy to differentiate C++ files from C files.

File Structure:

```
pi@raspberryPi:~ $ cd project
pi@raspberryPi:~/project $ ls
start.cpp   Dog.cpp   Dog.hpp   MyCustomObject.cpp
    MyCustomObject.hpp
```

```
// In start.cpp
#include <iostream>

using namespace std;

static int MAX_NUM_OF_DOGS = 5;
static int DOGS_PER_DAY = 3;

int main() {
    int numOfDogs = 5;
    MyCustomObject myObj; // A user-defined object.

    printf("We have %d dogs.\n", numOfDogs);

    for(int i = 0; i < DOGS_PER_DAY; i++) {
        if(numOfDogs <= MAX_NUM_OF_DOGS) {
            Dog newDog = new Dog(); // Another user-defined
                object.
```

```
            myObj.addDog(newDog);
        } else {
            cout << "The dog shelter is at maximum capacity"
                << endl;
            break;
        }
    }
}
```

9.1.5. Functions. The coding standard for the Linux Kernel explains the usage of functions nicely:

”Functions should be short and sweet, and do just one thing. They should fit on one or two screenfuls of text, and do one thing and do that well. The maximum length of a function is inversely proportional to the complexity and indentation level of that function. So, if you have a conceptually simple function that is just one long (but simple) case-statement, where you have to do lots of small things for a lot of different cases, it's OK to have a longer function.

However, if you have a complex function, and you suspect that a less-than-gifted first-year high-school student might not even understand what the function is all about, you should adhere to the maximum limits all the more closely. Use helper functions with descriptive names (you can ask the compiler to in-line them if you think it's performance-critical, and it will probably do a better job of it than you would have done).

Another measure of the function is the number of local variables. They shouldn't exceed 5-10, or you're doing something wrong. Re-think the function, and split it into smaller pieces. A human brain can generally easily keep track of about 7 different things, anything more and it gets confused. You know you're brilliant, but maybe you'd like to understand what you did 2 weeks from now.

In function prototypes, include parameter names with their data types. Although this is not required by the C language, it is preferred because it is a simple way to add valuable information for the reader."

Never use `goto`'s (without an extremely compelling argument).

9.1.6. Commenting and Documentation. While there is nothing inherently bad about commenting, it is possible to over-comment. Comments should not be about HOW the code works; it is much better to write your code so that its logic is obvious and instead focus on WHAT your code does. Try to avoid the need to put comments inside functions; if you have to partition and explain what a certain part of a function does, you should review Section 9.1.5 again. You should put comments above the function prototype (usually in the .h file) to document what the function does or why it is used, the parameters passed to it, and what it returns. It is also helpful to include any warnings about the function (if it throws any error or turns a `NULL` pointer). The syntax below used for commenting enables features used in Visual Studio to show the information when the cursor is over the function. Figure 3 shows an example of what this looks like.

```
/**
 * @brief     This function divides two numbers.
 * @param     dividend    The two numbers to divide.
 * @param     divisor     The two numbers to divide.
 * @return    The quotient of two numbers.
 * @note      This function returns MAX_INT if divisor == 0.
 */
int divide(int dividend, int divisor) {
    if(!divisor)
        return dividend / divisor;
    else
        return INT_MAX;
}
```

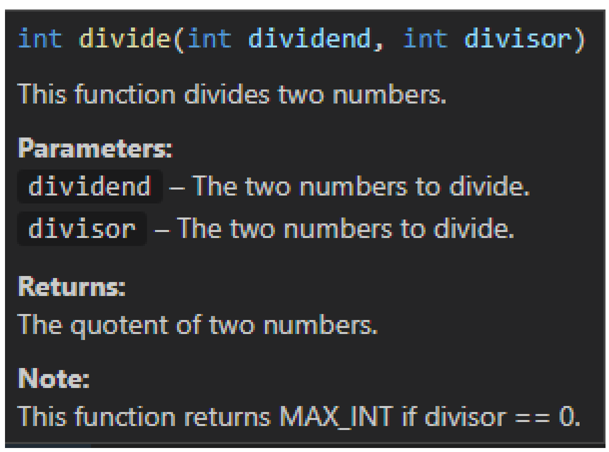

Figure 3. Example of Visual Studio's Feature

9.1.7. Macros. Use macros for constants, and macro functions for little tedious tasks (remember the tedious part). Surround macro functions in parenthesis and surround everything in parenthesis (remember, the whole function is inserted wherever it is used, so it could have unexpected bugs). Generally, macro functions follow the same naming rules as global constants, but it is not as enforced. Just make sure the work is obvious.

```
#define CHECK_BIT(byte, num)            ((byte) & (1 << (num)))
#define APPEND_2_BYTES(byte1, byte2)   ((byte1 << 8) + byte2)
#define GET_BYTE(data, num)             ((data >> (8 * num)) &
    0xFF)
#define bswap_16(num)                   (((num >> 8) & 0xFF) |
    ((num << 8) & 0xFF00))
```

9.2. GITLAB

Gitlab is used to store and manage the team's software. Each mission should have a single directory that can store multiple repositories for its subsystems. Repositories should be properly divided by subsystem and potentially divided within the subsystem. For example, in C&DH, there should be a separate repository for each device (MCUs for example).

9.2.1. Gitlab Workflow. Gitlab has useful features of which to take advantage. Features like branches, issues, and merge requests should be used at the minimum. Other features, like pipelines and gitmodules, should only be used by those who understand them and can properly document how to maintain them.

9.3. ORGANIZATION

It is important to organize the Gitlab so that it is intuitive and easy for new members to learn how to use it. There is no right or wrong way to organize repositories (repos), but there are better ways than others. The Software Lead, with the guidance of the CE and PM, should decide how to best organize the repo. It is recommended to take motivation from old repos as models for what intuitively makes sense and what does not.

The first, and possibly the most important file, is the README.md. Gitlab auto-generates this file for a new repo. This should fully explain what this code is meant for, how to navigate within it, how to compile it, and any other important aspects of the code. See below for a list of sections that should be covered, but the README should not be limited to this list. Note that this should not explain how the code works but what it does (the intent of the authors). How the code works should be explained with supporting documentation, see Section 9.4

- Organization overview
- How to compile and build the flight code
- How to flash the code (if using microcontrollers)
- Contributing rules
- List of contributors
- To-Do list (in the early stages of a repo this can be helpful, but Gitlab's issues tracker can be much more versatile than a to-do list)

9.4. SOFTWARE DOCUMENTATION

This section does not apply to software documentation using software comments. This section discusses the documentation that is separate from the software but is provided as supporting material.

9.4.1. Doxygen. All code written shall be documented using doxygen, a c++ code documentation generator.

9.4.2. Flow Charts. For both simple and complex systems, flow charts help those to understand the code without reading it. This is advantageous and shall be required. The level of depth will be up to the discretion of the software lead, but the flow charts shall be documented in the CDH100s.

9.5. AUTO-TESTS

Auto-tests are a useful tool in managing and testing a software base. The idea is that every data structure, function, macro, etc. that is created must have an auto-test that completely tests the software and its output. For example, if there was a function that takes two data structures you created and added them to a new one, then that function should be tested to use dummy, but real, inputs and compare the outputs against the known to assess structure changes. If the definition of the output data structure changed then this function would possibly not compile. If it does compile but somehow the output is now different then now the team knows this function needs to be addressed, whether that is deprecating or updating the function. If the output is the same then nothing needs to change! When building these auto-tests it is useful to log errors so that the team knows what went wrong and where. This comes in handy when managing a large software base. Additionally, these auto-tests should be set up to be run all at once, because it makes it simple to check if something broke while making changes to a repo. Doing this can save time and is a standard in the industry. This team flies real satellites in space, so this is a standard the team must follow and take seriously.

9.6. HARDWARE UNIT TESTS

Hardware Unit tests are similar, but slightly different, to the auto tests. The key difference is auto tests should test software independent of hardware integration. Hardware unit tests software dependent on hardware integration. It is best practice to divide the unit tests by hardware unit, if possible, and fully test the device, whether that is simply turning a solenoid on and off or testing communication to get data from a sensor.

10. RISK MANAGEMENT PLAN

Risk Management is the implementation of specific procedures or principles to understand risks and eliminate or minimize consequences as a result of the risk that all engineering projects encounter. Risk management plans vary in depth, effectiveness, and meaningfulness. The risk management plan discussed here is a short, meaningful and effective plan. This is not meant to be a fully encompassing plan. It highlights the most common and extreme risks along with brief strategies for mitigating those risks.

10.1. KNOWN RISKS

10.1.1. Student Unpredictability. Student unpredictability generally comes in the form of lack of consistent availability, motivation, and communication. These can lead to failure to meet mission deadlines, insufficient work progress, poor documentation, and ever-changing team size, etc.

10.1.2. Lack of Knowledge. The team members are students learning to be engineers thus the limited experience the team members have results in mistakes resulting from poor decisions. Not a single member on the team is yet a professional engineer; rather at most they are novice engineers. An effect in having novice engineers/students building satellites is they are much more likely to design systems with flaws.

10.2. METHODS TO MITIGATING RISK

10.2.1. Trade Studies. Trade studies are an effective way to minimize risks when making proper design choices. They can be tedious due to the time needed to document the design, or redesign process, but it slows the pace of finalizing a choice which can offer time to uncover mistakes or find better solutions. The trade study also ensures that the design is completely analyzed by requiring the CE, PM, PI, and subsystem Lead approval. Each trade study should be formatted into sections seen below and in Appendix C.

1. The Problem
2. System Requirements
3. Solutions
4. Evaluation Criteria (See examples below)
5. Final Design Choice
6. Approvals

Examples of Evaluation Criteria:

- Software Integration Considerations
- Hardware Integration Considerations
- Cost
- Power Consumption
- Mass
- Size

10.2.2. Principal Investigator Role. The principal investigator (PI) of the team is crucial to the team's success. The PI is a leader of the team, and as an experienced professional, they guide the team through proper systems engineering, technical and program choices, budget management, and other key risk areas. It is important for the team to properly keep the PI informed to enable the PI to provide feedback to the team.

10.2.3. Quality Engineering. Quality Engineering is an engineering discipline that manages the control, practicality, and quality of engineering products and services. Thus, Quality Engineering is a broad topic to cover, but some examples of quality engineering practices are discussed so members are able to understand and define their own quality engineering practices for designs, tests, assemblies, etc.

The easiest way to ensure quality engineering is to have extra team member(s) working on the project. This allows for peer review, and most members are able to identify simple mistakes. Often, tasks done in the lab are done by experienced members with less experienced members shadowing. This enables learning because the experienced member(s) are mentoring and fully explaining what is happening. This process typically results in fewer errors due to the older member(s) having their work tripled checked by explaining what they are doing and having other members observe.

Gitlab's features of merge requests are a form of quality engineering. The merge requests are a form of peer review, which is a practice of quality engineering. Thus, all software written for mission analysis, testing, or flight will be peer-reviewed in a formal fashion. This does not need to be in the form of a Gitlab merge request, but the review should cover any changes, additions, or removal of software.

10.3. RISK MANAGEMENT PROCEDURES

NASA's approach to risk management is broken up into two methods, Risk-Informed Decision Making (RIDM) and Continuous Risk Management (CRM). These are complementary processes, with typically CRM following RIDM. Overall, NASA's risk management plan at its simplest is the identification of risks then applying either the RIDM or CRM method. There are five critical roles in the risk management process; stakeholders, risk analysts, subject matter experts, technical authorities, and the decision maker. Stakeholders are ones who have interests invested in the program, risk analysts are those who handle the risk management, subject matter experts are those with the technical experience and wisdom to provide support via technical knowledge, technical authorities are those whose opinions or declarations hold decision influencing power, and the decision maker(s) who had the authority and responsibility with making decision. All five of these roles can affect the outcome of a decision and therefore all five roles should be involved in the risk management processes discussed in the following sections.

10.3.1. Risk Informed Decision Making. RIDM is typically done in the first phase of the mission as a way to identify risks and find solutions to eliminate or minimize risk, or from here on out mitigate risk. RIDM is split up into six steps, as seen in the list below.

RIDM Process:

1. Understand the stakeholder expectations and drive performance measures
2. Compile feasible alternative
3. Set the framework and choose the analysis methodologies
4. Conduct the risk analysis and document the results
5. Develop risk-normalized performance commitments
6. Deliberate, decided on an alternative, document the decision rationale

Starting with step one, the stakeholder expectations form the top-level requirements in the RVM. The top-level requirements are typically vague and complex. There are multiple solutions to getting more tangible requirements for a risk management plan. The first is a deeper dive into the lower-level requirements in the RVM that are applicable to the risk and scenario. Another is to look at budgets, schedules, or other documentation to track down numerical constraints. Some examples include financial, time, and power constraints. Another source that may not be as quantifiable is safety constraints. Aside from constraints, performance measures should be defined. A performance measure is a way to identify if a constraint is being met. For example, a typical constraint for a satellite is cost. If the budget is $100K that defines a constraint, but the performance measure is the project cost. Overall, performance measures should be compiled to accurately track the risk based on constraints that enables one to move forward in the risk management process.

The second step, compiling feasible alternatives, is simply searching for possible alternatives while minimizing decision traps (see Section 2.3.4). During this step, one should briefly identify feasibility. However, most of the analysis and examination are done in the following steps. It is pertinent that multiple alternatives are identified to ensure the best solution is selected and to have a list of alternatives for future consideration if the selected solution is not adequate.

In step three, set the framework and choose the analysis methodologies, the types of analyses are determined. The analyses should be selected so the results support the decision-making process. Therefore it is critical that all analyses are considered during this step to ensure nothing is missed. Working with satellites, this can become a multidisciplinary process. For example, a guidance, navigation, and control analysis can be dependent on the propulsion analysis. So redoing the propulsion analysis would require a revised GNC analysis as required by proper systems engineering practices. This requires traceability within the team which is why risk management processes should be properly communicated and documented. The previous example was a technical analysis but there are other analyses that should be considered such as safety, schedule, personnel power, reliability, operations, etc. Structuring the analyses is also a part of this step. Thus, configuration control should be set up to ensure all parties are working with the same information prior to moving into step four. Additionally, analysis outputs should be defined to understand what information is needed from the analyses. This provides traceability and a sense of the bigger picture to everyone.

Once the risk analysis framework is established then step four, conducting the risk analysis and documenting the results, can be completed. The goal of step four is to conduct any analyses and quantify performance measures. Once quantified data is in hand, one should also model the uncertainty in the data to support the decision-making process. Though qualitative information is important to consider in the decision-making process if there is a way to model qualitative information into quantitative data confidently and accurately then one should do so. If this is not the case then one should be prepared to assess and weigh the qualitative information during the decision-making process. Additionally, an assessment of the credibility of the risk analysis results should be considered during this step.

RIDM Functional Roles*

Stakeholders - A stakeholder is an individual or organization that is materially affected by the outcome of a decision or deliverable; e.g., Center Directors (CDs), Mission Support Offices (MSOs).

Risk Analysts – A risk analyst is an individual or organization that applies probabilistic methods to the quantification of performance with respect to the mission execution domains of safety, technical, cost, and schedule.

Subject Matter Experts – A subject matter expert is an individual or organization with expertise in one or more topics within the mission execution domains of safety, technical, cost, or schedule.

Technical Authorities – The individuals within the Technical Authority process who are funded independently of a program or project and who have formally delegated Technical Authority traceable to the Administrator. The three organizations who have Technical Authorities are Engineering, Safety and Mission Assurance, and Health and Medical. [11]

Decision-Maker – A decision-maker is an individual with responsibility for decision making within a particular organizational scope.

*Not to be interpreted as official job positions but as functional roles.

Figure 4. Functional Roles

Step five is developing risk-normalized performance commitments. A performance commitment is a performance measure that is compared to the ideal performance. The risk-normalized approach means that the performance measure ordering of priority and ideal performance is the same for all alternatives. This allows for a uniform comparison between alternatives. One should also determine risk tolerances to create acceptable ranges the performance measure should fall within. These risk tolerances should still satisfy the requirements. The relationship to the composed constraints and the priority level of the performance measure should be considered when establishing risk tolerances.

The last step is to deliberate, select an alternative, and document the decision rationale. Prior to deliberation, two objectives need to be met. First, the members of the deliberation should be determined. Members should only be those with functional roles, see Figure 4 for more on function roles. Next, a Technical Basis for Deliberation (TBfD) should be created. This will document steps one through five so that the deliberation has access to the same information and other future purposes. The TBfD also serves as a guidebook on discussion during the deliberation. Deliberation may be an iterative process because new risks, analysis flaws, and more can be brought to light during the deliberation which may result in re-baselines and required updates to the TBfD prior to a final decision being made. Once a final decision is made, it will be documented in a Risk-Informed Selection Report that serves to document the deliberation process and the reasoning for an alternative selection.

10.3.2. Continuous Risk Management. CRM is done throughout the mission cycle to manage known risks. CRM consists of knowing the scenario, the likelihood of the scenario, and the consequence or outcome of the scenario. An overview of the CRM process can be seen in Figure 5.

The CRM starts with identification of the scenario in the form of a risk statement. The risk statement should contain the ideal condition, the departure from the ideal condition, the asset under risk, and the consequence of departure, at the minimum. The risk statement's purpose is to provide a high-level understanding of the scenario and risk. The analysis step consists of an analysis of departure from the ideal condition and the magnitude of consequences it would cause. Next, the plan step involves either accepting the risk or taking actions to mitigate the risk. Alternately, one could decide to elevate the decision-making process to someone higher in the program or organization, this can be moving the

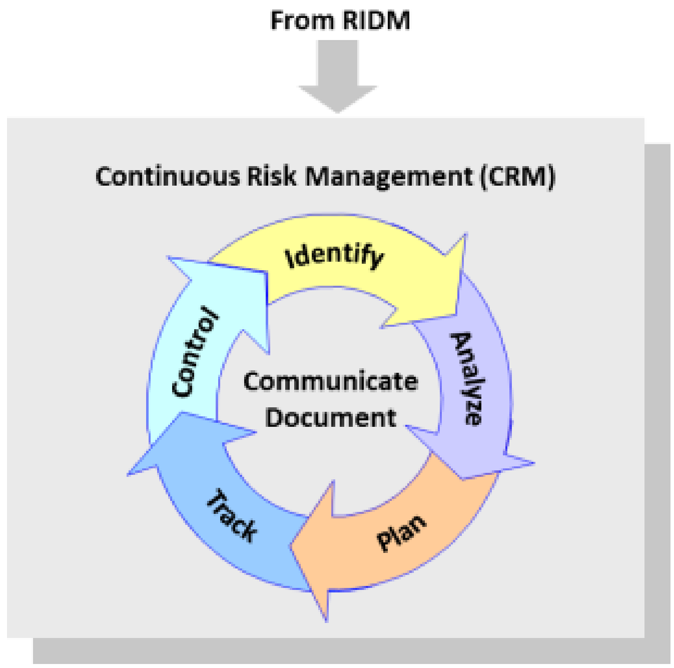

Figure 5. CRM Process

responsibility to a CE, PI, or even the sponsor. Taking actions to mitigate the risk can come in the form of creating contingency plans and researching risk drivers or preventative measures. Once the plan of action is defined for the risk and scenario, then tracking can start. This entails the collection of any data or monitoring of the situation and the risk drivers in order to take the proper course of action that was defined in the plan phase. The last step is control, where the outputs of the track step will determine the course of action in control. If the data or observations show that the risk is properly mitigated then a control document should be made. If the data or observations show the inability of the plan of action then the plan step needs to be revisited and revised. Regardless of the outcome, the process should be formally documented so that the CRM process can be documented and communicated to the necessary figures.

10.3.3. RIDM and CRM integrating. RIDM decides how to meet the requirements or constraints while CRM handles the management of the risks associated with implementation. As shown in Figure 6, the results of the RIDM process are used to create a scenario and risk statement for the identify step of the CRM process.

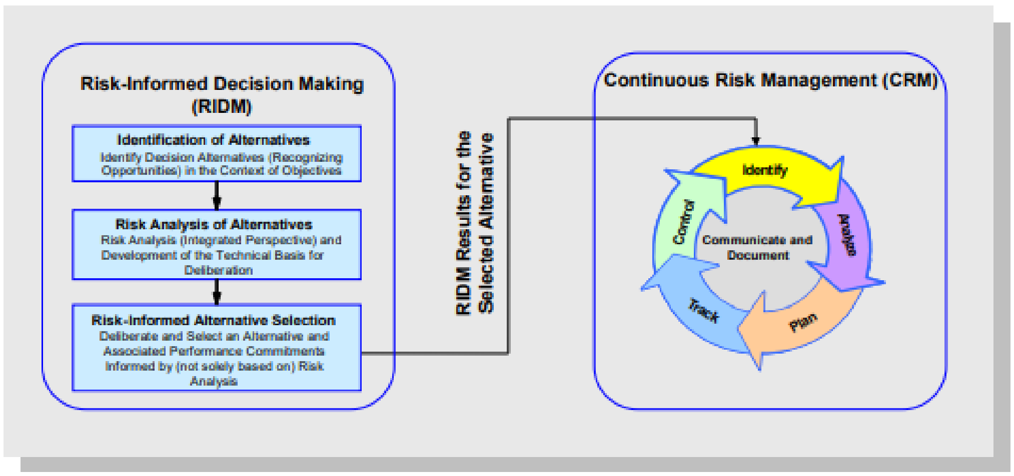

Figure 6. RIDM to CRM Integration

During the planning phase of the CRM, newly identified risks or scenarios can be discovered which may result in a ”re-baseline” of the RIDM process. Prior to the re-baseline, the severity of the new information should be considered to determine how to adjust the original RIDM documentation. It is rare for the RIDM to be completely redone so the goal of the baseline is to minimize the changes while still conducting a quality update. Once the re-baseline is completed, then one should move forward again in the CRM process starting with step one.

10.3.4. Risk Analysis Report. The M-SAT team has adopted a Risk Analysis Report that combines NASA’s documentation of the technical basis for deliberation and risk-informed selection report. This report documents the entire RIDM process enabling a more streamlined CRM process. Anytime a potential risk is identified, this risk analysis report should be created. During the RIDM process, if the risk has been deemed acceptable

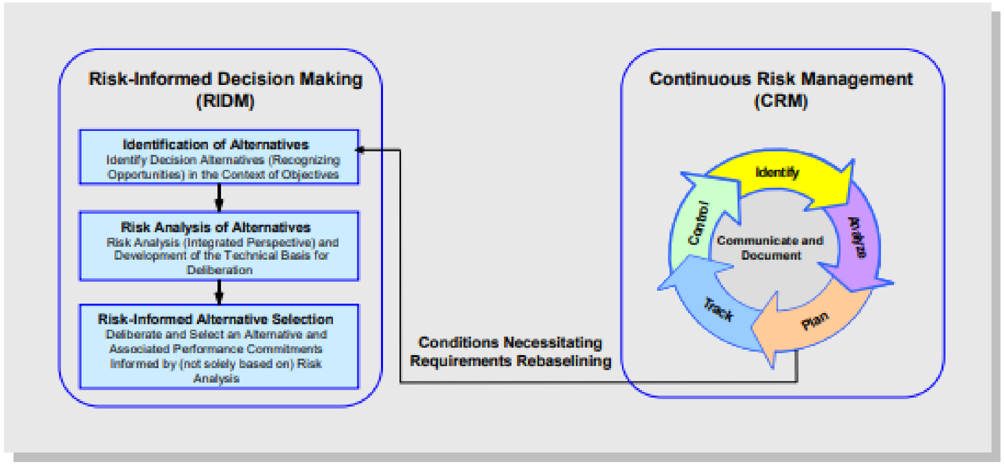

Figure 7. CRM to RIDM Rebase

or no-existent the risk analysis report should be produced to notify future members of this analysis. See RT101 in the technical resources folder for a template and guide for the risk analysis report or Appendix B.

11. ALMUNI ENGAGEMENT

The external relations managers are by default in charge of the alumni engagement activities and responsibilities. The Student Director should support this effort along with other members from the RM Team. Engaging with the alumni provides the team with support through networking, technical support, and even financial opportunities. As a result, the team should make an effort to stay connected with alumni.

11.1. ALUMNI EMAIL LISTSERV

At the end of every semester, the RM Team, and specifically the external relations managers, should work with graduating seniors to get their personal emails added to a Microsoft Outlook Group. This is not required, but is highly encouraged. The listserv will not be used frequently. The purpose of the listserv is to keep alumni engaged with M-SAT as they enter the workforce. There are many benefits for both parties as listed:

M-SAT Benefits

- Alumni can provide presentations to the team (educational or recruitment)
- Alumni can be solicited for many purposes (partnerships, sponsorships, specific training, etc.)

Alumni Benefits

- Alumni will receive a semesterly report on M-SAT's progress that includes substantial team progress and roadblocks
- Alumni will receive solicitations for partnerships when new missions are being proposed
- Provides a connection to contact the M-SAT Team for employee recruitment purposes

The RM Team should send out a newsletter to the listserv updating alumni on the team at least once per semester. Other items that can be included are a solicitation for alumni seminars, fundraising, partner requests, etc.

11.2. ALUMNI CONTACT LIST

At the end of every semester, the RM Team shall work with the mission leads to update the alumni listserv and contact list.

The contact list shall record the following:

- Name
- Email

- Phone
- Major
- Subsystems
- Missions
- Expertise

11.3. ALUMNI MEETINGS AND SEMINARS

Typically the team's alumni continue to work in the space industry and gain valuable knowledge that could benefit the team. Of the alumni who are willing to give back to the team, there are two different scenarios that can be leveraged. The first and easiest way to get alumni to engage with the team is to have them do technical seminars on a topic of their choosing and/or expertise. These seminars should be scheduled to assist the team with current challenges or knowledge gaps. These seminars can be specific to M-SAT or not. The second scenario is having informal meetings with the alumni to ask technical questions or get advice. This can be valuable to learn the history of a mission or subsystem and gain valuable insight into what they did before, so the team does not make the same mistakes. Additionally, the alumni can teach and educate the members about what they have learned in their professional career.

12. MISSION LIFE CYCLES

It is outside the scope of this document to fully explore satellite mission life cycles but because this guide serves as a reference to manage the team, it includes a brief introduction to mission life cycles with resources and initial direction.

12.1. CONCEPT STUDIES AND PROPOSAL

The first part of the mission life cycle includes producing a broad spectrum of ideas and interests for potential missions. The ideas must be presented with intent and the ideas should reflect what is possible for the team as well as the sponsors' guidance. From those ideas, concept studies will be done to determine what is best for the team and its mission statement. Concept studies can include developing ConOps, getting feedback from sponsors, initial mission overviews, etc.

Things to consider during concept studies:

- Team expertise and those supporting the team
- Cost feasibility
- Schedule feasibility
- Risks
- Sponsor requirements and recommendations
- Identify potential partners and stakeholders

This is not an exhaustive list

Once a final mission is chosen, writing the proposal is next. The proposal process varies for different sponsoring entities, and special attention should be paid to the request for proposal (RFP) to determine the requirements and any guidance on preparing the proposal.

12.2. DESIGN AND TESTING

Design and testing will overlap. It would be unwise to fully design a satellite without a single test. It is likely for design errors to occur, and testing is a good way to find any design errors. Design should start with creating an RVM and finalized ConOps. From there, subsystem CCDs and ICDs should be made with trade studies supporting design choices. The design of the satellite is reflected in the mission documentation; if it is not documented, then it has not been designed. Testing should start as simple as possible, with unit tests for sensors for example. Then the team will progress in test complexity to the Day in The Life Test. Following this procedure easily establishes traceability for the team members.

12.3. FLIGHT INTEGRATION

Prior to flight integration the majority of the mission's documentation should be finalized. This includes assembly, integration, and testing plans. Under the assumption these are quality plans, the highest risk comes from human error. Typically, students are experiencing flight integration for the first time and that in itself increases risk. An effective manner to mitigate this risk is by quality engineering. See Section 10.2.3 for more on quality engineering practices.

12.4. MISSION OPERATIONS

Missions Operations is a phase not many universities nor students get to experience. The Missions Operations planning should occur well before launch to ensure streamlined procedures are followed during Missions Operations. Having well-defined procedures during the Missions Operations eliminates delays caused by on-the-spot decisions, and it minimizes emotional impacts from mission operators and stakeholders, primarily the university participants. However, unexpected scenarios may happen, and a procedure for unexpected scenarios should also be addressed during the Missions Operations planning.

13. MANAGING A SUCCESSFUL TEAM

M-SAT's goal is to fulfill its mission statement of educating engineers of the future. Given that statement, as long as students are provided opportunities to learn and grow as engineers, then M-SAT is achieving its mission statement. Launching satellites in space is important to the team because it is a great learning opportunity but it is not critical to achieving the mission statement. Education can happen leading up to the launch and operations of a satellite in space. Sending a satellite to space only enhances the team's learning.

13.1. HIGHLY RESPECTED TEAMS

M-SAT strives to be a highly respected team among the other student teams at the university. Having members that respect each other and the team as a whole, is important for a positive and motivated atmosphere that will enable the success of the team's mission statement.

M-SAT's accolades and history help build respect for the team and for new members. M-SAT has a history of getting into Phase A in the UNP Program along with one mission that reached Phase B. Many of the team's alumni have gone to work for the top aerospace companies in the world. Including but not limited to NASA, SpaceX, Lockheed Martin, Northrup Grumman, General Dynamics, Boeing, The Aerospace Corporation, and more. A history of success tells potential members that joining the team is a great opportunity and it is important to convey this information as new members join.

History does not define the future; the team will only continue to be successful if its members keep each other accountable and continue to progress on the current mission(s). It is important that every member is kept accountable for assigned tasks. Without accountability subsystems and missions can collapse and students tend to leave the team when they see it coming.

Organization and formalities set the tone for how members perceive the team. It takes work to properly structure and maintain the team's organization and other formalities such as documentation. It is a crucial aspect of the team because without it, the team will be chaotic and members will not know how they fit in to the team. This could result in members feeling left out and potentially leaving the team. Likewise, if a member can understand the structure and organization of a team, they can identify how they fit in the team and understand their impacts on the team thereby, enhancing their experiences and their motivation to stay.

13.2. INCENTIVES FOR STUDENTS

It is important to understand that students are not on the team for selfless reasons; they choose to join the team for self-benefit. The team members will stay on the team and be a productive member if they understand the benefit. The majority of students do not get paid, but those who do tend to be easily motivated by the financial aid received for doing work. For those that do not get paid, motivation needs to be found another way. The primary motivator for non-paid students is the hopes of obtaining an internship, co-op, or a full-time job. It is the leadership team's responsibility to convey the fact that working on the M-SAT will improve the chances of obtaining jobs and higher pay.

It is important the students know the self-benefit of being on the team, enjoy being on the team, and are working on something in which they are interested. If one of these three statements is not upheld, then it is likely students will not remain on the team for more than a semester.

13.3. TEAM BONDING

Whenever members of the team are together, whether it is in the SSE lab or in the field conducting tests, there should be a positive atmosphere that makes the members enjoy being part of the team. This seems obvious but in the midst of everything that goes on inside and outside of the team, it can commonly be forgotten.

At least once a semester, the RM Team shall plan a team bonding event to build and encourage a positive team atmosphere. Many of these events have food and games for team members to relax and enjoy the event.

Examples of Team Bonding Events:

- Senior Send-Off Picnic
- Team Bowling Night
- BBQ and Field Games
- Project Presentations

13.4. THE SELECT FEW

The select few is a theme that appears in M-SAT along with many other university teams. This is an idea that there are a select few individuals, typically ranging from one to ten, that do the majority of the program management and technical work for the missions. Given that, it is important to identify and retain those members who excel. The easiest way is to get them paid through M-SAT funds, scholarships, or even work-study.

Due to the circumstances of the team and human nature, it is difficult to avoid this. Therefore, it is important to recognize this fact and embrace it to the fullest extent. The personality of the ”select few” tends to include being self-motivated and hard-working. The RM Team will look for these traits when recruiting to maximize member quality.

14. CONCLUSION

The M-SAT SEMP provides guidance for the leadership team and its members. The team will use their best discretion to apply principles from this SEMP. The SEMP will minimize mistakes, but it is inevitable that students will make mistakes, especially when it comes to systems engineering and program management. It is important to recall the mission statement, ”To educate engineers of the future through satellite design, development, integration, testing, and flight.” The primary mission of the team is to educate the members of the team. It is crucial that members do not focus on only the successes and failures of the team's satellite missions. A successful mission does not fulfill the mission statement nor does a failed mission mean the mission statement has failed. As long as the students who are team members are learning and challenging themselves to be better engineers, then the M-SAT team can be called a successful student team.

15. REFERENCES

15.1. INTERNAL REFERENCE

- TRS101 - Risk Analysis Report Template
- TRS102 - Trade Study Template
- TRS103 - Requirement Verification Matrix Template

15.2. EXTERNAL REFERENCES

- Constructing a System Engineering and Management Plan for a University Satellite Team
- NASA's Systems Engineering Handbook
- NASA's Risk Management Handbook

- The Lunar Engineering Handbook
- Essentialism: The Disciplined Pursuit of Less

16. PCB DESIGN EXAMPLE

Appendix B serves as an example of how to apply the general design process to a specific design process.

16.1. DESIGN PROCESS

16.1.1. Determine Purpose and Scope. As discussed in Section 6.2.1, the purpose and scope need to be determined. The majority of the purposed can be derived from the RVM but the scope will need to be determined by collaborating with other subsystems.

16.1.2. Trade Studies. Multiple Trade studies may be applicable based on the scope of the design. For example, trade studies may need to be made on MCUs, avionics designs, and communication protocols, board size, etc.

16.1.3. Determine Revision Control. For simplicity, the revision control for the PCBs is as follows:
PCB Name Rev:Rev Number

The PCB Name should be thoughtful and related to its purpose. For example, the board controlling the power system is commonly named the Electrical Power System (EPS) board.

The revision number shall follow the format of X.YY, where X and YY are separately incremented. The inital revision number of a board will be 1.0. For minor changes, the YY revision shall be incremented. For major revisions, YY shall be reset and X shall be incremented. Also, a new PCB Design Document shall be made for each major revision. This ”resets” the revision change log section and keeps the document cleaner. The difference between major and minor changes is up to the discretion of the team. For examples see the following.

16.2. PCB DOCUMENTATION FORMAT

16.2.1. Board Purpose. The first section should cover the PCB's purpose. This includes the constraints imposed upon the board and the objectives it is attempting to meet. For example, it is primarily a board to handle power transfer, communication, etc. This section should explicitly state the requirements it will satisfy from the RVM.

16.2.2. Interface Control. This section will cover the PCBs software or hardware interfaces to other devices. Any external connections (headers, dsubs, mounts, etc.) will be documented here and clearly described. Communication protocols to other boards should additionally be documented here along with a set list of documented commands or exchanges of information between the PCB and another device.

16.2.3. Schematics. The schematics section shows the PCB design along with an explanation of how elements are designed and how they work. The schematics should be split up into smaller subsections. The theory and intent are explained for each schematic for reasons explained in Section 6.2.6.

16.2.4. Change-log. The change-log is a section in the PCB design documentation for the revision. It is important for a change log to be documented properly, as discussed in the previous section, so that members understand what was changed for what reason. This section is reset every time a major revision happens.

16.3. REVISION PROCESS

The following is the procedure M-SAT members are required to follow for PCB Design.

PCB Revision Procedure:

1. The member(s) shall implement the design rev (1.0→1.1) and conduct all working changes in a new and separate directory.

2. The member(s) shall make unit changes, additions, or removals to the board. (Unit change refers to all three from here)

3. The member(s) shall have the unit change reviewed by another individual with PCB design experience.

4. The member(s) shall document the unit changes in the design document for the PCB. The documentation should, at minimum, show before and after figures of the schematic and a brief overview explaining the unit change.

    - Note: In some cases, a single figure is not enough to show the change. If this is the case then it may be recommended to break up the unit change into smaller unit changes. In other cases, it is acceptable to skip the figures if the overview is detailed enough for an inexperienced member to understand.

5. Steps 2-4 shall be repeated for all unit changes.

6. A final review shall take place to ensure changes are correct and relevant subsystems are aware of the changes.

    - The final review shall consist of the revisor(s), reviewer, and other noteworthy personnel such as professors, subsystem leads, subsystem members, CE, PM, etc.

7. Once the revision is approved, the revisor's names and revision date shall be recorded in the documentation.

Examples of Major Changes:

- Board size change
- Multiple additions/removals from the board
- New MCUs that affect the layout of the board
- Substantial reorganization of the board

Examples of Revision Increments

- Minor change: EPS Rev 1.2 to EPS Rev 1.3
- Major change: Motherboard Rev 1.5 to Motherboard Rev 2.0

APPENDIX B.

M-SAT TRADE STUDY TEMPLATE

1. INTRODUCTION

1.1. SCOPE

This subsection will outlines the scope of this trades study.

1.2. APPLICABLE DOCUMENTS

This subsection will list any internal or external applicable documents

2. DEFINING THE PROBLEM

This section will clearly state the design needs. There are two subsections that fall into the scope of this section, system requirements and goals.

2.1. SYSTEM REQUIREMENTS

The system requirements state what requirements the trade study must solve, these typically come from the RVM but other requirements that come from launch providers are shown here too. Requirements are statements that the solutions must meet, with no exceptions.

2.2. SYSTEM GOALS

System goals for the trade study as objectives the trade study should meet. Such as minimizing cost but, they should be as explicit as possible. Therefore, instead of stating to minimize cost, is it more appropriate to state the solution should be under five thousand dollars. The difference between requirements and goals should be explicitly stated to give the designers clear expectations moving into the trade study process.

3. SOLUTIONS

The more solutions documented in this trade study the better. One purpose of this trade study is to provide a reference for the future of the mission. If the chosen solution from this trade study is deemed insufficient, the document can serve to be a great tool. Rather than starting from scratch, future members can use this trade study as a starting point for a new trade study during the redesign process. Additionally, future missions will benefit from having old trade studies to review. The solution should be detailed and have references to informational items such as data sheets and wikis.

4. EVALUATION

The evaluation section will take the proposed solutions and compare them. A scoring based on requirements and goals met is typical in a section like this. There are multiple methods to weigh and score solutions so the team members should work together to create a scoring systems that works best for the specific trade study.

5. FINAL DESIGN CHOICE

This section discusses the explanation of why the decision was made. This should cover what made the chosen solution best and why the alternative solutions were not chosen.

6. DIGITAL APPROVALS

Chief Engineer: <Type Name>

Date: <Insert Date>

Program Manager: <Type Name>

Date: <Insert Date>

Subsystem Lead: <Type Name>

Date: <Insert Date>

Principal Investigator: <Type Name>

Date: <Insert Date>

APPENDIX C.

M-SAT RISK ANALYSIS REPORT TEMPLATE

1. INTRODUCTION

This template provides initial documentation guidance on risk analysis and management. Please see the M-SAT Systems Engineering and Management Guide (SEMP) for more guidance on risk management.

1.1. TECHNICAL SUMMARY

The technical summary should provide a risk statement. A risk statement contains a condition, departure from ideal condition, and asset under risk, and a consequence. This subsection should also provide and information that is important to be aware of to accurately understand the risk.

1.2. APPLICABLE DOCUMENTS

This subsection will reference any internal or external documents used to support the risk analysis.

2. REQUIREMENTS AND EXPECTATIONS

2.1. RVM REQUIREMENTS

List RVM requirements this risk has effect on.

2.2. OTHER REQUIREMENTS

List or describe other requirements this risk has effect on. Examples include requirements from launch providers, stakeholder expectations, technical burdens, etc.

3. DERIVATION OF PERFORMANCE MEASURES

A performance measure is a way to identify if a constraint is being met. Performance measures should be compiled to accurately track the risk based on constraints which enables one to move forward in the risk management process. This section should include a list of performance measures and why they have been chosen.

4. DECISION ALTERNATIVES

This section will identify feasible alternatives by simply searching for possible alternatives while minimizing decision traps. It is pertinent that multiple alternatives are identified to ensure the best solution is picked and to have a list of alternatives in the future if the decided solution did not work. This section will provide detailed information on the possible alternative and point the readers to more information via data-sheets, websites, industry contacts, etc.

5. RISK ANALYSIS FRAMEWORK AND METHODS

This section will identify any analyses or methods to compare and weigh the alternatives. The analyses and methods should be picked so the results support the ability to make a decision. Therefore it is critical to ensure all analyses are considered during this step to ensure nothing is missed. Analyses can include technical, safety, schedule, personnel power, reliability, operations, and other kinds of information. Structuring the analyses is also apart of this step. Thus, configuration control should be setup to ensure all parties are working with the same information prior to moving forward. Additionally, analysis outputs should be defined to understand what information is needed from the analyses.

6. RISK ANALYSIS RESULTS

This purpose of this section is to document the results of the analyses or other methods conducted. Assessment of any information is conducted also to determine accuracy and importance so the decision maker fully understands the results and can make a risk informed decision.

6.1. RISK ANALYSIS CREDIBILITY ASSESSMENT

7. PERFORMANCE COMMITMENTS

This section should will document performance commitments, priority levels of the performance commitments, risk tolerances, and acceptable ranges of the performance measure. This information is tracked to ensure the decision maker fully understands the environment surrounding the risk.

8. DELIBERATION

This section will document the deliberation process and the rationale for the decision to demonstrate the decision was risk informed and shows the decision maker worked with other to make an informed decision. This includes discussion on organization of the deliberation, analysis of the alternatives (pros/cons for example), etc.

8.1. DELIBERATION NOTES

Meeting notes from the deliberation go here.

8.2. DECISION AND RATIONALE

Details on the decision making process and rationale behind the decision is discussed in this section.

REFERENCES

[1] U. S. Database, "Satellites in Orbit by year of Launch," 2021.

[2] Euroconsult, "Government space budgets," 2022.

[3] R. Millan, R. von Steiger, M. Ariel, S. Bartalev, M. Borgeaud, S. Campagnola, J. Castillo-Rogez, R. Fléron, V. Gass, A. Gregorio, D. Klumpar, B. Lal, M. Macdonald, J. Park, V. Rao, K. Schilling, G. Stephens, A. Title, and J. Wu, "Small satellites for space science," *Advances in Space Research*, vol. 10, p. 64, 2019.

[4] D. Joyce, L. Winterton, D. Pierce, J. Letchworth, and B. Dingwall, "NASA Undergraduate Student Instrument (USIP) Project–Lessons Learned," Annual Conference on Small Satellites, 2019.

[5] "University Nanosatellite Program Website," 2023.

[6] S. Franke, M. Pilinski, M. Diaz-Aguado, S. Forbes, and G. Hunyadi, "The University Nanosat Program from Concept to Flight: A Dual Student Program Perspective on What Works and What Does Not," 20th Annual Conference on Small Satellites, 2006.

[7] "CubeSat Launch Initiative Website," 2023.

[8] NASA, *NASA Systems Engineering Handbook*. 2016.

[9] O. of the Deputy Director for Engineering, O. of the Under Secretary of Defense for Research, and Engineering, *Systems Engineering Guidebook*. 2022.

[10] K. Craig and M. L. Nagurka, "Multidisciplinary engineering systems 2nd and 3rd year college-wide courses," 2010.

[11] J. Olson, S. Patel, K. Yoshino, L. Jasper, J. Ganley, and L. Hunt, "Helping students reach space: University nanosatellite program challenges and lessons learned," 32nd Annual AIAA/USU Conference on Small Satellites, 2018.

[12] "Incose systems engineering," 2023.

[13] G. M. Horine, *Absolute Beginner's Guide to Project Management*. Que Publishing, 2009.

[14] F. M. Al-Rashed, "Risk management of bridges and satellites: A comparative study of bridges and satellites," *Journal of Bridges*, 2019.

[15] NASA, *NASA Risk Management Handbook*. 2011.

[16] "Iso/iec/ieee international standard for systems and software engineering – life cycle management – part 4: Systems engineering planning," tech. rep., ISO/IEC/IEEE, 2016.

[17] A. M. Steward, "A guide to the establishment of a university satellite program," Master's thesis, University of Missouri-Rolla, 2007.

[18] B. Cox and M. Aktik, "The intelligent application of quality management to smallsat program," 19th Annual Conference on Small Satellites, 2005.

[19] L. Berthoud and M. Schenk, "How to set up a cubesat project - preliminary survey results," University of Bristol, 30th Annual AIAA/USU Conference on Small Satellites, 2016.

[20] B. Wallace, S. Palo, and J. Sobtzak, "A university deep space cubesat mission: Lessons learned from the university of colorado boulder earth escape explorer (cu-e3)," 36th Annual Small Satellite Conference, 2022.

[21] *Interplanetary CubeSats: Summary of the Current Thinking, Approaches and Lessons Learned*, ASU Deep Space Summit, 2022.

[22] M. Cho, M. Teramoto, T. Yamauchi, G. Maeda, S. Kim, and H. Masui, "Program management for sustainable university cubesat programs based on the experience of five generations of cubesat projects, birds program," Laboratory of Lean Satellite Enterprises and In-Orbit Experiments, Kyushu Institute of Technology, 36th Annual Small Satellite Conference, 2022.

[23] E. Kroeker, A. Ghosh, and V. Coverstone, "Buidling engineers: A 15-year case study in cubesat education," University of Illinois at Urbana Champaign, 30th Annual AIAA/USU Conference on Small Satellites, 2016.

[24] L. A. Ziegler, "Confirguation, manufacture, assembly and integration of a university microsatellte," Master's thesis, University of Missouri-Rolla, 2007.

[25] S. N. Withrow, "Program management for concurrent university satellite, including propellant feed system design elements," Master's thesis, Missouri University of Science and Technology, 2019.

[26] L. Berthoud, M. Swartwout, J. Cutler, D. Klumpar, J. A. Larsen, and J. D. Nielsen, "University cubesat project management for success," 33rd Annual AIAA/USU Conference on Small Satellites, 2019.

[27] G. McKoewn, *Essentialism: The Disciplined Pursuit of Less*. Crown Publishing Group, 2014.

VITA

Jacob Daniel Anderson was born in St. Louis, Missouri. He grew up in the St. Louis area and started attending Missouri University of Science and Technology (Missouri S&T) in 2018. Jacob graduated with his Bachelor of Science in Aerospace Engineering, becoming a first-generation college graduate in his family, in May of 2022. He attended graduate school at Missouri S&T where he graduated with a 4.0 in his Master of Science Degree in Aerospace Engineering in May 2023. During his undergraduate education, Jacob was an undergraduate research assistant with multiple professors conducting research in aerodynamics, fuel and flow cells, and satellite systems. Jacob also had internships in aerospace manufacturing and space simulations. Prior to his senior year, he joined the Missouri S&T Satellite Research Team where he quickly became Chief Engineer for the MR and MRS SAT Mission, and by the end of his time at Missouri S&T he had also held the roles of Command and Data Handling Lead, Communications Lead, and Student Director supporting both the M^3 and MR and MRS SAT Missions. As a graduate student, Jacob was a graduate teaching assistant for Introduction to Spacecraft Design Lab. Jacob also was on the varsity track and field team at Missouri S&T where he attended Division Two Outdoor Track and Field Nationals in 2022 where he placed 14th with ambitions to attend the Division Two Outdoor Track and Field Nationals in 2023. Jacob will start his career off in the industry by joining The Aerospace Corporation in Colorado Springs, Colorado.