A Methodology for Successful University Graduate CubeSat Programs

Aaron P. Aboaf, Elliott S. Harrod, Matthew Zola, Arunima Prakash, Scott E. Palo, Robert Marshall,
Marcin D. Pilinski, Nicholas Rainville, Andrew Dahir, Vikas Nataraja, Bennet Schwab,
Adam Gardell, Les Warshaw
University of Colorado Boulder
3775 Discovery Drive, Boulder, CO 80309
aaron.aboaf@colorado.edu

ABSTRACT

The University of Colorado Smead Department of Aerospace Engineering has over a decade of success in designing, building, and operating student led CubeSat missions. The experience and lessons learned from building and operating the CSSWE, MinXSS-1, MinXSS-2, and QB50-Challenger missions have helped grow a knowledge base on the most effective and efficient ways to manage some of the "tall poles" when it comes to student run CubeSat missions. Among these "tall poles" we have seen student turnover, software, and documentation become some of the hardest to knockdown and we present our strategies for doing so. We use the MAXWELL mission (expected to launch in 2021) as a road-map to detail the methodology we have built over the last decade to ensure the greatest chance of mission success.

INTRODUCTION

CubeSats at the University of Colorado

The University of Colorado Smead Aerospace graduate program has launched and operated four successful missions to date: CSSWE [1] [2], MinXSS-1 [3], MinXSS-2 [4], and QB50-Challenger [5]. Two missions are expected to fly in the next 18 months: MAXWELL & CU-E3 [6]. Four additional missions are expected to fly within the coming decade: COSMO, CANVAS, SWARM-EX, & AEPEX [7]. All of these projects are student led, managed, and executed with oversight from select faculty members and professionals and are staffed primarily by students at the master's level. Hundreds of students have contributed across these missions and joined the workforce with valuable skill sets derived directly from their experience working on these missions.

After a decade of implementing varying techniques to put together project infrastructure and programmatic infrastructure on various CubeSat missions we wanted to relay our experiences and recommendations about how we execute a student led CubeSat program. With four missions flown and at least six missions expected to fly in the coming decade, compiling this collective record of the best practices on how to run a CubeSat program is milestone of where we are ten years into the process. It is a collective effort from the faculty and student leadership involved on eight current and former graduate student led missions to continually revise existing program strategies and to incorporate major lessons learned over the years. We wanted to set forth a formal recommendation based on how we execute a successful student-run CubeSat mission. We hope that this paper not only acts as a record for ourselves to look back on in the future but that it provides a starting point for other educators and other Universities to gain some insight from our experience on how to successfully execute a student led CubeSat program of their own.

For much of the paper we use the MAXWELL mission as an example. The MAXWELL mission began in 2016 and is nearing the flight build phase in order to prepare for an expected launch in early 2021. The program execution on the MAXWELL mission is informed by lessons learned on past missions and has demonstrated great success in establishing a finely tuned baseline for programmatic practices. Figure 1 shows the final flight configuration of the MAXWELL CubeSat.

Project Sponsors

All of the current and former graduate student led CubeSat missions have had a source of funding outside the University that typically covers hardware, testing, and some student salaries. One of the ma-

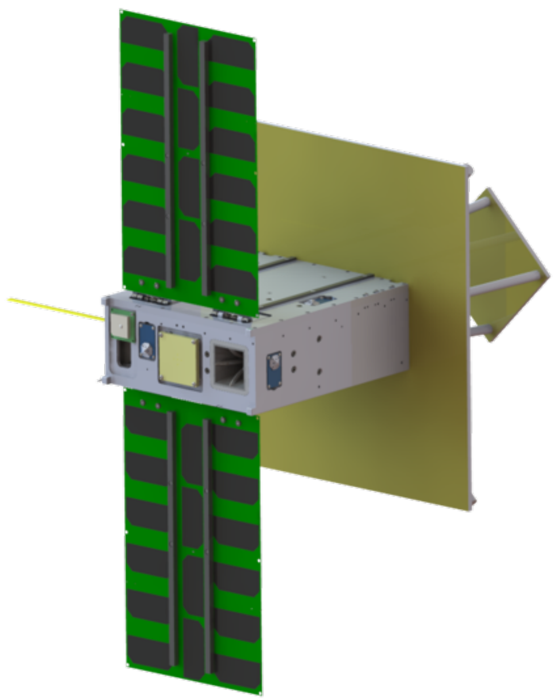

Figure 1: MAXWELL CubeSat

jor upsides of student run projects are that students participate on the project for credit towards their degrees which enables students to get credit while learning. All of the projects to date have been part of a National Science Foundation (NSF) grant, a University Nanosatellite Program (UNP) grant (in the case of MAXWELL) [8], or funded by advancing in a NASA Challenge competition. These are competitive opportunities that are widely available for academic faculty to apply for.

Students generally respond well to having some presence of external responsibility that lies outside day to day team activities. For missions such as MAXWELL there is, at minimum, an annual review with the staff at the UNP office that is typically conducted on site at the University. This gives all the students working on the project an opportunity to interact with the project sponsor and potentially additional expert technical reviewers. Students have the opportunity to get critical feedback about their design and solution approach in a context that feels almost like a true critical industry review rather than something more academically focused.

Having the project sponsor involved and designating an official program wide review at the end of each semester helps set milestones for progress and documentation across the program. Students understand the responsibility to gather all the loose ends they have been working on and bring them together in the context of the project. Often in preparation for these cumulative reviews, significant communication between the subsystem teams occurs and details related to system interfacing and testing are often worked out too. This extra spur of communications also helps identify deficiencies and risks which if had been left unchecked could become major headaches later on in the project life cycle.

Student Personnel Composition

Student personnel for our current and former CubeSat projects consists primarily of master's students. In particular, the University has opted to recruit students by offering credit through a two-semester long graduate projects course within the Aerospace Engineering department. Most semesters, the team consists of roughly 20 students, with 6-8 of them being enrolled in the graduate project course, 3-5 being enrolled in an independent study course, and the rest volunteer or hourly paid positions.

The MAXWELL CubeSat uses a particularly high percentage of custom-designed hardware as opposed to commercially available systems, which means that it is often necessary for students to be at the graduate level in order to have the knowledge necessary to design and test these systems. For MAXWELL we have designed our own power electronics circuit board to provide bus voltages, manage battery state of charge, and collect housekeeping data of the power system. This communicates with a custom command & data handling circuit board that manages all the other systems on-board the spacecraft. The embedded software proficiency and knowledge of electrical board design needed to design these is typically found at the graduate student level in students that have an electrical engineering background. If the CubeSat is more of an interfacing oriented design where EPS, C&DH, ADCS, etc. subsystems are purchased off the shelf or managed by some external contractor then the reliance on the skills typically held by graduate students is less so. The drawback of working with strictly commercially available subsystems can limit the ability to customize hardware for the mission.

Other aspects of the project, such as mechanical design, project management, and integration and testing can be worked on by motivated undergraduate students. We have found that the undergraduate students working on the project often stay involved for longer periods of time, which greatly aids in the tran-

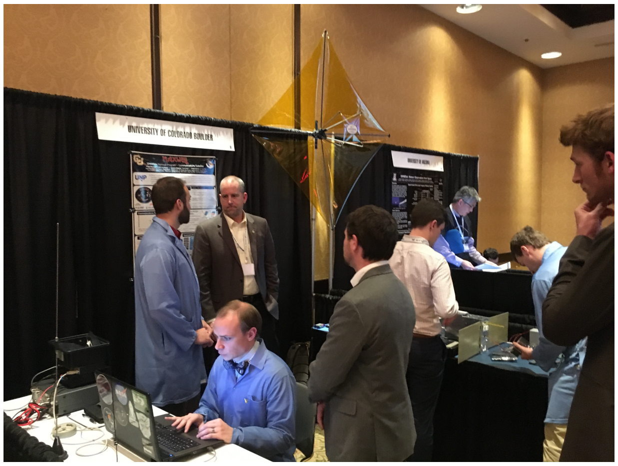

Figure 2: MAXWELL Leadership Preparing at the UNP-9 Flight Selection Review

sition between graduate students. For project management and systems engineering in particular, it is highly beneficial to find a student who is able to dedicate three or more semesters to the project since it takes significant time to fully on-board folks for these positions. An undergraduate who can spend two years with the project might take a bit longer to onboard and learn the necessary skills but since they stick around for much longer time frame than the typical two semesters of a graduate student, they can end up contributing a lot to the mission.

Balance

Managing a mission is a balancing act between programmatics and actually accomplishing work. While it would be ideal to cut as much "red tape" as possible and remove overhead from the process this can be disastrous in the context of the frequent personnel turnover brought on by the semester based educational environment. No matter how you slice it, the project life cycle will last years and the student workforce will turn over which leads to gaps in knowledge between the new workforce of students coming onto the project and those moving on. Thus it is critical that student led CubeSat teams achieve a balance between programmatics that help manage the retention of mission knowledge across generations of students and across subsystems and technical development.

There are two major facets of the MAXWELL CubeSat approach to creating and maintaining this balance. The first is the team wide task tracking infrastructure focused around waterfall charts and team wide weekly tag-up meetings to track task progress. Any slips in task deadlines are caught early on, allowing the team and the management to revisit the planning and re-evaluate and update the execution timeline. This trains students to estimate a time frame for a given task and complete it by the set deadline. A high-level task is broken down into highly specific tasks assigned to the relevant subsystems. Figure 3 shows a single testing task on a waterfall chart for the ADCS subsystem broken down into multiple sub-tasks each with an estimated hour time cost. High specificity in tasks helps students make a better selection as to what tasks they will be responsible for. With document tracking and waterfall charts, these tasks are well assigned and tracked. The whole process gives the students an impression of the industry expectations while giving them the flexibility to move around pick and choose their tasks. Students are better prepared to step into the industry, having had a taste of the process in a university-level setting.

The second part of the MAXWELL approach is centered around document tracking and version control. On MAXWELL this is a relatively recent addition to the programmatics but we have seen significant improvement from past semesters on the ability to hold onto knowledge within the project. An official document numbering, release, and tracking system was established that is applied to every important document on the project. Management decides what documents are tracked officially and these usually include test procedures, test results, definition documents, and training documents among others.

Task	Total Est.	Per Person	Total Acr.	Per Person
Sun Sensor System Integrated Test	66		0	
Write sun sensor integrated test procedure for photodiodes	15	15	0	0
Get test procedure approval for methodology	3	3	0	0
Run sun sensor integrated test and data collect	10	10	0	0
Code and results review for integrated data collect test	4	4	0	0
Address review action items	10	10	0	0
Procedure redlines for integrated test	4	4	0	0
Integrated test and data collect round 2	8	8	0	0
Delta code and results review	2	2	0	0
Complete verification documentation	10	10	0	0

Figure 3: Example of a Detailed Task Breakdown on a Waterfall Chart

Management handles most of the work to assign document numbers and track revisions of the documents which means the team members focusing on technical contributions don't have much extra to worry about. We also have an easy procedure for using the university approved digital signature platform to author official signed releases of documents which helps with tracking official versions of the documents. The MAXWELL team has also continued the tradition of using Google Docs to manage all internal team documents because the storage space is unlimited and all Google Docs products have automatically saved lifetime version history.

PROGRAM INFRASTRUCTURE

Leadership

Building an effective leadership structure is key for seeing early success and follow through on a CubeSat program. Leadership is responsible for instituting policies, procedures, and setting the example for the students working on the program. Finding the right students to manage the project can be a tricky task because it takes a balance of having the organizational know how, the technical proficiency, and also the availability to effectively be involved with the program.

On MAXWELL, as on past missions, we have two or three major student leadership positions on the project. These are the project manager, the systems engineer, and depending on the phase of the project also a chief engineer. It is critical that these two or three students have regular contact with the PI and the project advisors and that they communicate among themselves how best to relay the "vision" of the project to the students working on it. The PI should work the student leadership define what the "vision" of the project is and hold the expectation that the student leadership works to have each student working on the project meet the expectations of that "vision".

Leadership positions, especially after the PDR phase of the project are best supported by students who are willing to make a longer term commitment to the project. A student in these leadership positions that only spends a semester or two in them rarely has enough time to really grow into that role and own it before they move on to something else. Having a longer term commitment from a couple of students in these leadership positions leads to a more effective continuation of project procedures and documentation as well as keeping up the knowledge base on the project. Especially for the systems engineer and the chief engineer roles, having long term students fill these means that detailed technical knowledge about the system and the status of the project hardware and software is tracked much more accurately over time. The longer a student spends in these roles the more comfortable they become with the system and the more confident they become in determining what needs to be completed to see forward progress.

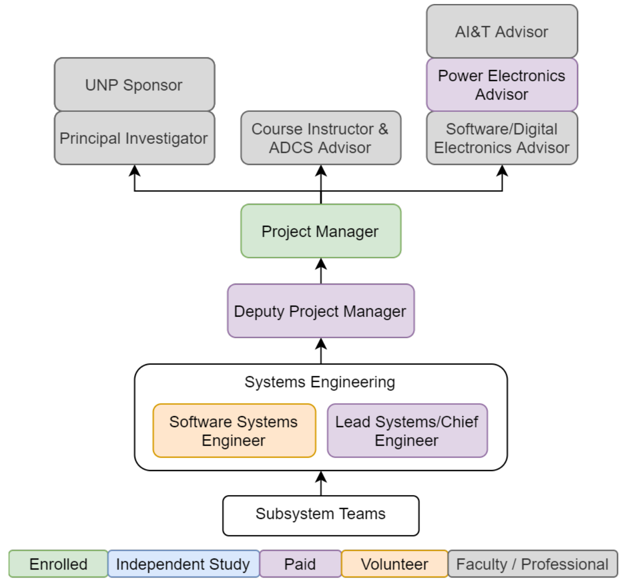

Figure 4: MAXWELL Leadership Org Chart 2019-2020

The student leadership should be vocal about recognizing particular accomplishments of the students working on the team in front of other students. This may be as simple as recognizing a student each week for completing a particularly difficult task or writing high quality documentation about a test. Even if the accomplishments are minor it provides some motivation to the whole team when they see that people can get recognized for putting forward good work. Going hand in hand with this is also the recognition of the important work done by past members of the team. As students transition off the project, some will leave their tasks with a very clear and detailed path forward for whoever takes them up. This should be recognized as the goal for all students working on the project as they too will eventually transition off. Highly motivated students have even created helpful "how-to" videos for incoming students which have been extremely helpful.

Recruitment & Staffing

From a project management perspective, project recruitment and staffing is one of the most critical parts of a university-based CubeSat project. For the CubeSat missions managed and staffed by students at the master's level, most of the outreach that is done is within the aerospace department. Due to this limited exposure to primarily aerospace engineering majors, there is a significant knowledge divide that develops in the project teams. We have found that aerospace engineers tend to be drawn to working on the attitude determination and control, mechanical design, and thermal control subsystems. This leaves large personnel gaps in other critical systems such as RF communications, electrical power design, and embedded firmware programming. Thus, it has been critical to the success of the CubeSat projects that teams get the word out to other departments at the university, such as Electrical and Computer Engineering and Computer Science.

Another issue unique to university-based CubeSat projects is the difficulty associated with student turnover. The full time commitment and focus typical of paid industry professionals is simply not possible for university projects because there is typically not a significant budget allocation for paying students. Instead, we have opted to format the project as a part of a two-semester long graduate projects course within the Department of Aerospace Engineering so students get credit towards graduation for their work. This means that a majority of the students working on the project are limited to only two semesters on the project because of the way the course and graduation requirements are structured. Additionally, almost all incoming students have little to no experience in CubeSat design, integration, test, or operations. As a result, there is an ongoing cycle that occurs each semester where the new students are going through an on-boarding process before they can start contributing. This cycle reduces the amount of time that team members can focus on progressing the project itself, because they are spending nearly a quarter of each semester on-boarding new students.

There has been an ongoing effort to find ways to keep students around for longer than two semesters, and this is often accomplished by getting paid position, taking an additional independent study enrollment, or by continuing in a volunteer capacity. Paid positions are directly related to budget restrictions and almost exclusively reserved for students that have already been involved in the project and are up to speed. The biggest challenge with volunteer students is that there is not an established method of holding students accountable for their work as they are not being directly compensated. It can be difficult to secure a long-term commitment out of the gate from a volunteer because it is hard for students to plan that far into the future. Some ways of combating this include offering independent study credits to students, so that they receive credit towards graduation, as well as assigning volunteers smaller tasks

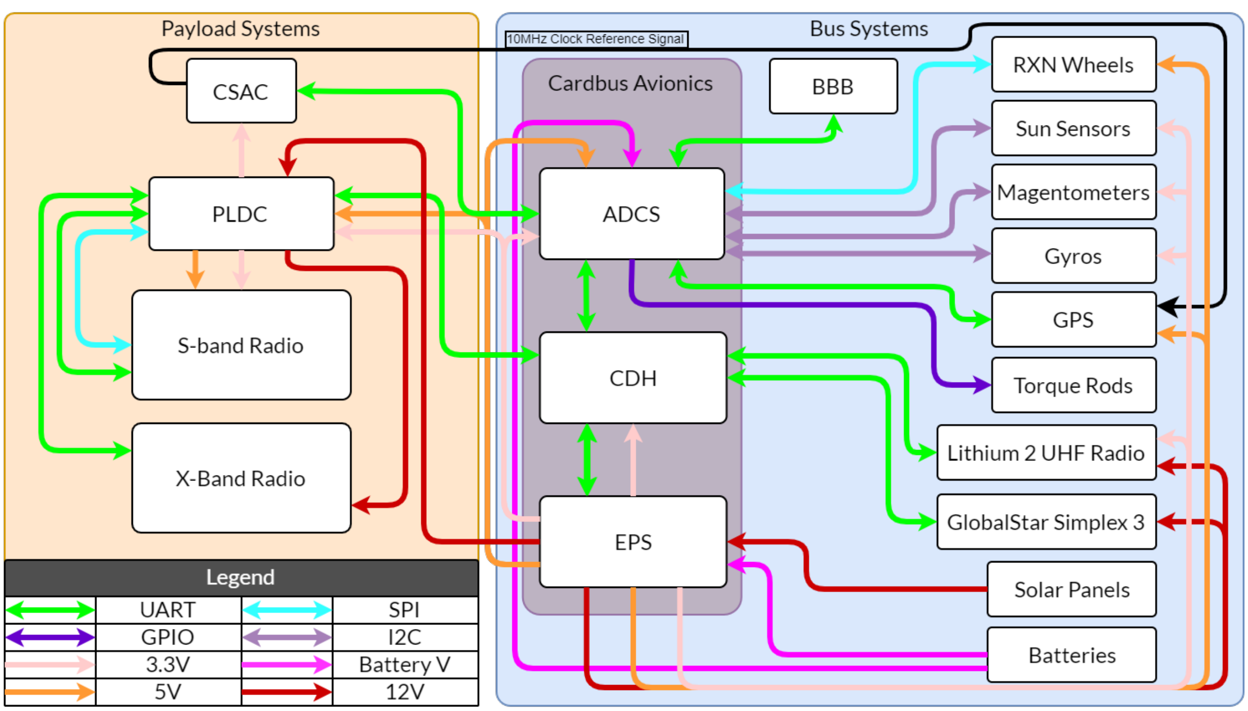

Figure 5: MAXWELL Highest Level System Power & Interface Block Diagram

and tasks not on the critical path that can be completed in a matter of weeks, so that their progress can be properly monitored.

Training Approach

Training student team members on any CubeSat program must be approached in an iterative manner to determine the methods that are the most effective. Evaluating the on-boarding process on a semesterly basis can quickly pinpoint the most effective methods and allow the team to build on them each iteration. Thinking of training as a constantly evolving process instead of a static method is critical to continuing to improve the the way it is done for new folks joining the mission. Teammates typically respond well to hands-on training processes where they can actively interact with the hardware under the supervision of an experienced student and see how subsystems work with each other. Shadowing experienced students working in the lab is also an added-value activity for new students.

We have also had the most success when a detailed on-boarding process is laid out that new students can follow on their own that takes them through documents describing the mission such as the Mission Handbook and the system state diagrams and hardware block diagrams. Figure 5 shows one of the first block diagrams that new students working on the MAXWELL program see because it does a good job of showing how every component in the satellite system is powered and interfaced in a way that is easily understandable. One of the most important things to remind all students, but especially new ones coming onto the project, is that they should be completely comfortable asking any questions even if they think they are trivial.

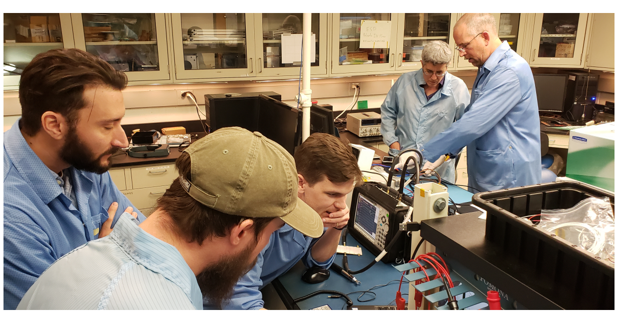

Figure 6: Hardware Specific Training with New Members in the Space Technology Integration Lab

Before any technical work can begin in the lab, safety and practical training also needs to take place. To reduce repetition on a person to person basis this training takes place as a group in the beginning of each semester. This consists of a general overview of specific lab policies, ESD practices, cleanroom practices, and a lab walk-through. It is encouraged that each team member goes through this training even if they have previously completed training.

Following general training, groups are established based on team members skillsets and subsystem interests for more specific training as shown in figure 6. Hardware focused engineers will practice lab specific soldering techniques and go through wire harnessing training. Software focused engineers will be trained on software specific coding practices and architectures as well as how to program the hardware in order to test new software features. This is an opportunity to practice skills without worry about damaging anything and to ask questions since experienced personnel are in the room. Following general training, the on-boarding program moves into program specifics as students settle on which subsystem the want to work with.

The method that has shown the most success on the MAXWELL CubeSat program is one that ties a general overview of each subsystem followed by an in-depth walk through of hardware and integration procedures. To accomplish this a Mission Handbook outlining the entire CubeSat is provided to new teammates. This handbook contains a high level overview of mission objectives followed by a low level breakdown of each subsystem. This provides new teammates the opportunity to gain a high level knowledge, while also providing a guide for where their skills might be most applicable.

Once a teammate has gained basic knowledge of the CubeSat's system level layout and the teammate's specific subsystem, a deep dive of the subsystem hardware is the next task. This involves providing the teammate with schematics of relevant printed circuit boards (PCBs) for their subsystem and a walk-through of how the PCB functions with the design engineer or the chief engineer (CE). The teammate is encouraged to ask any questions about reasoning for design or functionality. Following this in-depth walk-through of the subsystem, the teammate is taken into the lab where the design engineer will show how the subsystem is set up. This involves showing the proper location of the hardware, how the hardware is connected to the power supply, how the hardware interacts with other subsystems, and how to accomplish tasks like programming a chip or configuring the communication readouts to the computer.

At this time the teammate has gained the knowledge of how the subsystem functions and how to interact with the hardware. Now the subsystem lead engineer will begin assigning small one to two week tasks to the new teammate with a support engineer to assist with issues that may arise. This is the "trial by fire" part of the on-boarding process which will increase the teammates involvement with the project as well as provide a specific and clear way that they can make tangible contributions. The initial on-boarding process should only take about 10-15hrs before turning up the pace to get new engineers working on tasks.

As the teammate continues learning and contributing to the project, the subsystem lead engineer will begin providing reasoning of how the subsystem will directly correlate to the Cubesat's high level mission objectives. This provides the new teammate with a deeper understanding of their work and further understanding of how their work interacts with other spacecraft subsystems. This give students a bigger stake in the project which can improve the retention rate of students from year to year.

Transitions

In the University setting, personnel turnaround and knowledge transfer is a significant risk that needs to be addressed and mitigated. Without proper documentation and transfer of knowledge, new team members are left to their own devices to determine the state of the project and develop a plan to move forward. The primary way of ensuring that knowledge transfer takes place is through detailed documentation. At a surface level this includes keeping track of all completed work and a plan for future work. Ideally, students are staggered such that there is always a first-semester and second-semester member on each team such that the more senior member can spread the transition of knowledge over the course of an entire semester. For the the MAXWELL project, this process is aided in technical reviews being held twice a semester. In these reviews, all members of the team present their progress to the principal investigator and project advisors, as well as get input on how to move forward if anything is holding them back. During these reviews, new students are able to get a deeper understanding of the state of the project, which will help them effectively contribute and prepare to lead their subsystem in the future.

Two of the largest losses during personnel transitions are the knowledge of how to use unique hardware and software required for CubeSat development and the next tasks to be accomplished to move a particular subsystem forward. For the former, we've found it to be great practice to have students create instructional videos and guides for software and detailed procedures for hardware. This allows new

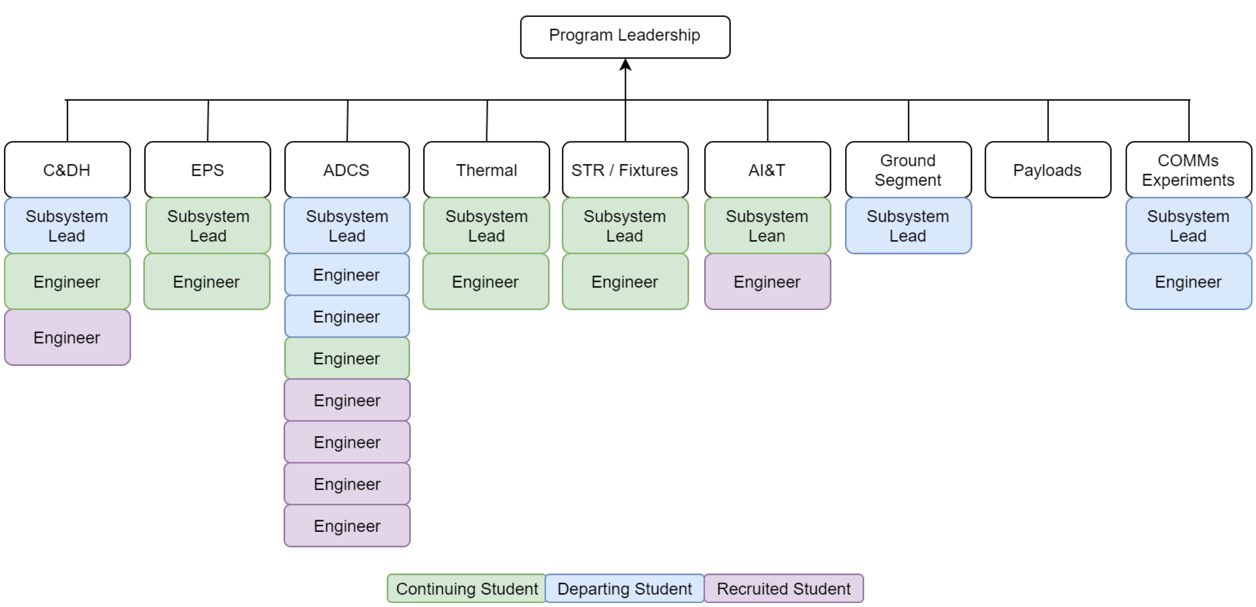

Figure 7: MAXWELL Spring 2020 to Summer 2020 Team Transition Status

students to work on the on-boarding process independently, which frees management up to facilitate the in-person on-boarding training and continuing subsystem engineers to continue to push tasks forward.

Additionally, it has proved to be largely beneficial to leverage the expertise of the faculty at the university to help aid in the knowledge transition if it is not possible for students to be on-boarded concurrently with existing students. Before students finish their time working on the project, it is expected that they complete a burn down list that outlines what tasks are left to be done for their particular subsystem. We have also seen that it is extremely important that students close out and complete the tasks they worked on during the semester because tasks that get left incomplete have a tendency to fade away and remain unfinished. With the proper foresight and preparation, the risk of knowledge loss amidst personnel transitions can be significantly mitigated, and it can even be advantageous to get a fresh set of eyes on the design.

Team Communication Tools

Ease of communication between team members is a key part of facilitating transfer of information quickly to the necessary parties so that work can move forward at the most efficient pace. Unlike industry which typically relies on email, the MAXWELL team and other student led CubeSat programs use an instant messaging platform like Slack or Discord that allows students to post in specific group channels and direct message one another. On the MAXWELL program the project advisors and the PI also participate in the Slack platform allowing them to respond directly to student queries so that everyone can benefit from the exchange.

Contingency Planning

Planning for unexpected events is something that is probably getting more thought these days. Destructive hardware failures can set back testing and or development of the system and are certainly not planned events. While these accidents are rare, we have never had a program go on without one. The standard mitigation strategies apply here by implementing procedures to safeguard hardware when in use and physical protection circuitry from voltage and current surges. When mistakes happen, because they will, it is also critical to spend some extra time analyzing what happened in order to figure out what can be implemented to prevent those mistakes from happening in the future. This presents an opportunity to develop short case studies that students can read coming on to the project to help them understand the importance of following procedures to protect the hardware.

There are other contingencies that should always be at least on the table for discussion as the project matures. Often budget, schedule, and sometimes personnel can present challenges for the completion of the project. It is important to remember that projects don't end with launch and there still are resource needs for operations and end of life tasks for the mission. Options to de-scope the mission are potential ways to alleviate concerns revolving around budget, schedule, and personnel.

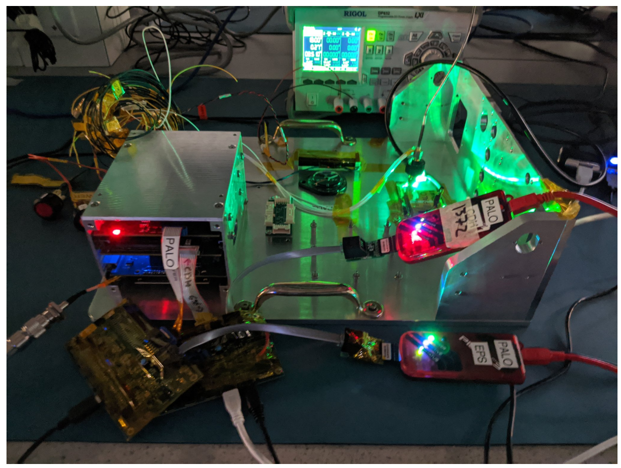

Figure 8: MAXWELL Remote Operated FlatSat Built in Response to 2020 University Lab Closures

PROJECT INFRASTRUCTURE

The Project Life Cycle

MAXWELL's mission life cycle is probably slightly on the long end of the typical build lifetime of a student let CubeSat project. Since MAXWELL uses mostly custom designed electronics and writes every bit of software the build time is longer just because there is so much more to do. MAXWELL is a UNP mission and thus follows the official system life cycle phases outlined by the UNP program. This includes some additional reviews near the beginning of the project life cycle (System Concept Review (SCR), Program Management Review (PMR), and Flight Selection Review (FSR)) that may not necessarily be representative of all missions. However, with the UNP review guidelines and progression the process ends up following the typical industry review order.

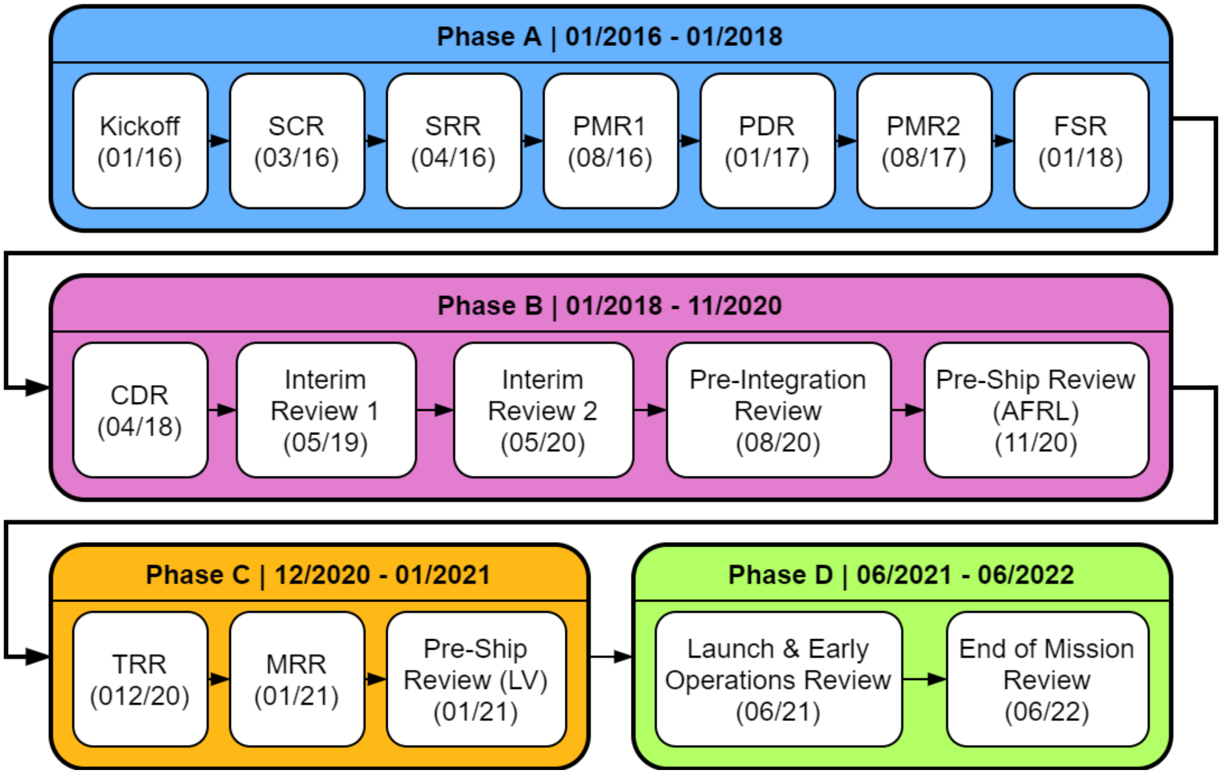

Figure 9: MAXWELL Project Life Cycle

Figure 9 shows the project life cycle for the MAXWELL project. Phase A lasted for almost 2 years and began with a kickoff introductory meeting and finished with a successful Flight Selection Review in January 2018. The first major review of Phase B was a Critical Design Review (CDR) which was followed by several interim reviews. Phase B will be complete when MAXWELL completes the Pre-Integration Review and Pre-Ship Review with flights hardware ready for testing. Phase C consists of Test Readiness Review, Mission Readiness Review, and Pre-Ship Review for launch after environmental testing is complete. The mission finished up with Phase D and the End of Mission Review happens when all end of life operations are completed and the mission objectives have been accomplished. As it currently stands, the MAXWELL mission is expected to span about six and a half years.

The project division into Phases A, B, C, & D is probably the most applicable to all University CubeSat missions. Phase A includes anything up through the mission PDR and Phase B includes the majority of the design and the engineering unit build. Phase C includes all the environmental testing on the flight unit and all final pre-flight checkouts. Finally Phase D goes through launch and finishes once end of life operations are complete. Each mission's total lifetime is really going to depend on what the mission is and the complexity of the build, but all will follow the Phase A, B, C, & D outline.

Hardware Approach

CubeSat hardware can really be divided into four categories: mechanical, electrical, commercial off the shelf (COTS), and ground support. In the early stages of the program, not a lot of thought is being put directly towards hardware as the program operates in a high level feasibility space, ie. nothing is considered at the fastener or resistor level. In Phase B the detail is fleshed out all the way down to the fasteners and resistors. Beginning in late Phase A there are probably a few commercially available development boards in house to prototype some of the spacecraft functionality and given the available budget there is also a low to medium fidelity model of the proposed mechanical structure. Phase B sees most of the hardware development and integration. Custom electrical boards are designed, reviewed, tested, and modified in cycle. COTS components are acquired for engineering unit level testing, and a high fidelity mechanical structure is settled on. Integration fit checks are also performed as components arrive and pass unit level and functional testing.

Printed circuit boards tend to be the most intensive hardware development effort on a CubeSat program. Depending on the complexity of the system and the availability of personnel with PCB design skills, there are usually multiple revs of each PCB required to get to a flight ready design. For the development of PCBs the MAXWELL team has established the following methodology.

Before beginning new development of a PCB, a strong version control system must be implemented. The design software might have version control already embedded within the software or even a basic GitHub will be sufficient. Having the version control process available will make changes down the road much easier. It will also allow the whole team to be able to report white wires and other design modifications with the board directly into the version control software so that everything related to that board design stays together in one place. We also highly recommend building a common parts library that can be used across all PCB designs to eliminate confusion and simplify parts usage across multiple designs. This will allow the team to buy reels of standard resistors and capacitors instead of having potentially several different part numbers for the same value part. It is also recommended that a BOM template is generated for a program to keep continuity between subsystems and ordering procedures.

During early stages of hardware development, there is great knowledge that can be acquired from internal and external technical experts. Throughout the development process the design engineer should hold in-depth technical design reviews of each PCB release. Openly discuss potential obstacles and remedies at these reviews to improve the robustness of the PCB design. When developing a new PCB the importance of having multiple, easy access test points cannot be stressed enough. This includes test pads as well as test points large enough to solder white wires. The ability to probe and inspect various issues with a PCB design will greatly reduce troubleshooting time and potential for shorting from hand soldering.

Once the PCB has been fabricated and populated, the PCB is given a serial number and added to a PCB tracking sheet. A best practice following PCB delivery is to perform a PCB acceptance test. This test involves five major categories: visual inspection, unpowered physical inspection, powered physical inspection, functional testing, and final assessment. For a visual inspection, the design engineer visually inspects each new PCB preferably with a microscope or magnifying glass and camera. This is an opportunity to look for any noticeable defects such as lifted traces, bubbles in the PCB substrate, unsoldered components, shorts across pads, diodes installed backwards, installed Do Not Install (DNI) components, crooked components, etc... An unpowered physical inspection is to ensure all of the power planes and communication lines have connectivity across the PCB. Using a handheld digital multimeter, check for continuity and discontinuity between all power planes, ground planes, and communication lines.

A powered physical inspection is when the PCB is powered on and the voltages levels of the power planes are verified. Functional testing also involves PCB specific items which may consist of programming a microcontroller, reading data from sensors, or actuating devices. The final assessment involves taking photos of the PCB and noting any issues found in the previous four steps. At this point, the design engineer makes a final assessment determining if the PCB is ready to go through further integrated testing or if the PCB needs to go back for rework. Any anomalies found are recorded in an acceptance test document and the PCB tracking document.

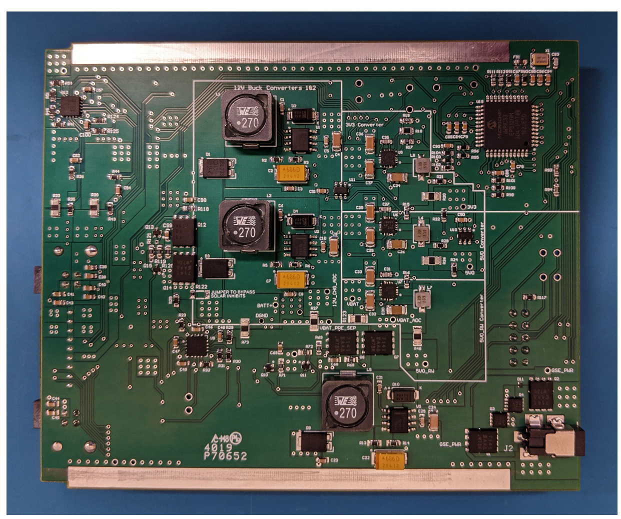

Figure 10: Completed EPS PCB After Functional and Acceptance Testing

Once the PCB completes acceptance testing successfully, further testing may be performed for the specific program. It is recommended that when development and testing is performed that engineers work in pairs. One engineer will vocally read the test procedure and another engineer will perform the operation while vocally confirming. By having teams work in pairs, this significantly reduces the potential for hardware failures due to negligence. During further development, any modifications made to the PCB or anomalies found during testing are recorded in the PCB tracking document.

Software Approach

The software on the CubeSat is what ties all the hardware systems together and managing proper software development can be tricky. The MAXWELL software platform is an example of taking existing software (from the QB50 mission) and trying to tweak it to run on a new platform with similar hardware. This is a challenge in itself because it is difficult to decide what code to recycle and what code to replace. Just because some code was appropriate for the QB50 mission does not mean it fits well in the context of the MAXWELL mission and it needs to be replaced. We have found that a lot of the communication driver level code is reused but much of the mission specific code needs to be ripped out and replaced. The overall software architecture stayed fairly consistent across the two projects but since each mission was different major sections of the code base had to be re-written.

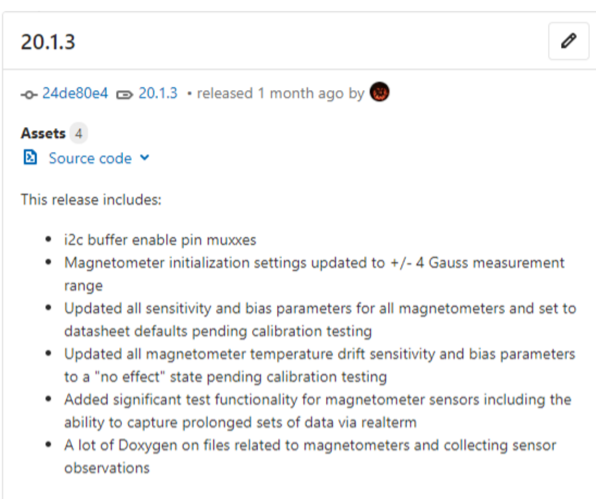

Figure 11: Example of an ADCS Software Release on GitLab

The first step when looking to do any software development is to establish a version control system. MAXWELL uses a GitLab interface to manage version control of all embedded and simulation software on the project. With the GitLab interface the MAXWELL team has created multiple software repositories that correspond with different parts of the spacecraft system. Each microcontroller that is programmed has its own repository along with major simulation repositories. There is even a repository for ground testing software and lab equipment support software.

The team has also defined a standard set of proper coding standards. This definition outlines how code should be formatted and how comments should be added to maintain code readability. All persons working on software should be constantly reminded that they will not be the last person to read through their code so they are expected to add all the relevant information in comments so that someone else who may not have a good understanding of what is happening in the code can still read through and understand what is going on. This is critical because software development on the project goes on for years and the personnel turnover is inevitable.

Comments should never be the only source of documentation for the software. MAXWELL uses Doxygen, an open source documentation engine, to generate linked HTML documentation for entire codebases. Doxygen compliance ensures that comments make it into the code and creates a LaTeX and HTML documentation package that can easily be read, searched, and distributed. Another very useful diagram to develop is a software state diagram. This is probably something that is developed prior to most of the code base as well since it will inform the overall architecture structure of the code. Making sure to keep the state diagram updated is an ongoing task as aspects of the spacecraft system are adjusted while the design progresses forward.

Early in the project life cycle software starts out fairly conceptual and decisions about the overall software architecture are made so that the planned software implementation is going to be compatible with the hardware. Hardware and software integration is absolutely something that can be considered when choosing hardware to interface to as well; sometimes the software development time cost of interfacing with certain hardware can be too difficult to overcome and it is easier to just avoid choosing that hardware component.

As the project life cycle moves into Phase B the software work on the project will start to diverge into two camps. One camp will be for direct software to hardware communication which is primarily the low level embedded programming for communication protocol drivers and for directly communicating between various subsystems and sensors. The other camp will start to work on higher level software implementation related to the flight algorithms and mission decision making. It is important to not let these two camps diverge too much and that the software interface between the camps is consistently updated so that information can easily flow from the hardware into the decision making parts of the software.

Another piece of documentation that is critical for the success of mission is a software interface control document. This will define all the software interfaces between subsystems making sure there are both enough communication busses for subsystems to be physically connected and define all the data fields, protocols, and information designators that get transferred both internally and externally to other software interfaces.

Testing often with software goes hand in hand with the version control tools. Having the version control means that running many tests on the software side can be easily tracked and repeated. Frequent testing with the hardware is important for ensuring that interfaces are robust and the more time spent in the software prior to launch the greater the chance that mission ending bugs might be caught and resolved.

AI&T Approach

A detailed assembly, integration, and test plan must be laid out prior to starting the integration process. The assembly plan is a detailed layout of how the spacecraft is put together. It is important to dry run these steps with the engineering unit with the actual tooling to find points in the process that need to be adjusted for a smooth flight integration process. Using the engineering unit as a practice run will help flesh out the integration procedure and identify issues with tooling, integration order, and harnessing before attempting a build with the flight unit.

A detailed integration plan involves determining how the subsystems will interact with each other. Some spacecraft rely on a card stack (backplane or motherboard) integration of PCBs, while others might rely on wire harnessing. Detailed drawings of serial communications schematics are useful for determining how the spacecraft works from a systems perspective. With this knowledge on hand, a detailed drawing of a wiring harness is created and the best harness mounting points on the spacecraft are marked. Using the engineering unit to workshop wire harnessing and determine the exact lengths required for flight is also extremely useful for a seamless flight integration.

Testing is the most important stage of delivering a successful spacecraft. The rule of thumb here is to test as you fly, and fly as you test. At this stage in the spacecraft's life cycle, the configuration must not change. There are three main tests to complete before delivery, each involves writing a detailed test plan and having equipment ready beforehand.

In our experience, thermal cycling is where most failures occur. Using an ambient pressure thermal chamber or ambient temperature vacuum chamber before attempting more expensive thermal vacuum chamber testing helps uncover issues before they happen in thermal vacuum. This also a great way to build up ground support equipment from basic tests into more involved tests.

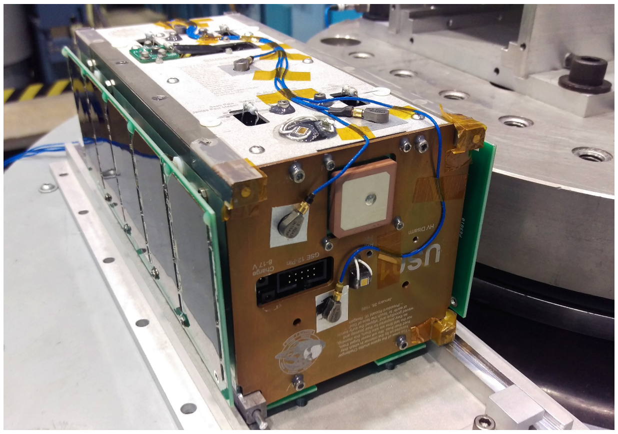

Figure 12: QB50 Satellite Being Prepared for Vibration Testing

After successful completion of the thermal cycling, the spacecraft may move into the thermal vacuum chamber. This chamber will more realistically simulate a space environment. Prior to placing the spacecraft in the TVAC, there must be harnessing made to interact between the spacecraft and the TVAC feedthroughs.For our TVAC testing we have the spacecraft undergo 8 full thermal cycles from the lowest expected temperature to the highest expected temperature. Full functional testing of the spacecraft should be completed before and after to provide a baseline and check for anomalies. Consider and plan for less common things in TVAC like RF hats if you want to test radio communications as well. TVAC is a time and labor intensive test and the more preparation taken beforehand can dramatically improve the smoothness of a TVAC test.

Vibration testing is the final test to complete before delivering the spacecraft. This uses a vibration table that will simulate the rocket profile of the intended launch vehicle. Prior to the test, an interface plate must be designed to interact the the table. Full functional testing of the spacecraft should be completed before and after to provide a baseline and check for anomalies. CubeSats that have come out of the graduate projects program tend to mechanically over designed because there is typically extra mass available and mechanical structures do not need to be optimized for weight thus the mechanical safety factor tends to be quite high. Consider vibration testing as mostly a workmanship verification of the system signaling that it is ready for the trip to space.

RECOMMENDATIONS

The MAXWELL CubeSat program had benefited greatly from the decade of experience gained from previous student led missions through the graduate program at the University of Colorado and has continued to improve the techniques and methodology used to train students to contribute meaningfully to the mission. We have seen the importance of having the project sponsor and external reviewers involved at least once each semester to facilitate critical and constructive feedback and to point out deficiencies that need to be addressed. Doing this makes students go through those few weeks of trying to tie up loose ends which increases communication among the subsystems and often uncovers risks that need to be addressed and mitigated.

We have seen the benefits of establishing a long term plan for leadership approach. Programs want to keep talented students in leadership positions for longer terms than the typical student commits as this assists continuity, maintaining the "vision" of the project, and the retention of the knowledge base. We have successfully incorporated motivated undergraduate students to help supplement a nominal graduate student workforce because they can make longer term commitments to the project.

Iterating our training process each semester has helped refine the way new information is presented to students and new students have gotten up to speed and began to contribute in shorter amounts of time after each iteration. Personnel and semester transitions are challenging to manage and we have continued to hit recruitment hard while also trying to raise awareness about the project across engineering departments which has led to greater interest and student involvement overall.

Finally, we have seen the importance of having a task burn down list for each subsystem to keep track of hardware and software work even after students transition off the project. Knowing what still needs to get done is an easy thing to track as progress is made but a very difficult thing to think of from scratch every time there is student turnover. Having good document, hardware, and software version control processes and tracked documentation also helps programs run as efficiently as possible in an academically structures student led environment.

ACKNOWLEDGEMENTS

This work has been supported in part by students involved with the MAXWELL CubeSat program which is funded by contract partnership between the Utah State University Research Foundation and the Regents of the University of Colorado Boulder, Contract No. FA9453-16-D-0004 (TO 0001) & Subcontract No. CP0049860.

References

[1] Q. Schiller, X. Li, S. E. Palo, L. W. Blum, and D. Gerhardt. The Colorado Student Space Weather Experiment: A successful student-run scientific spacecraft mission. In *AGU Fall Meeting Abstracts*, volume 2015, pages ED14A–04, December 2015.

[2] Rick Kohnert, Scott Palo, and Xinlin Li. Small space weather research mission designed fully by students. *Space Weather*, 9(4), 2011.

[3] James Paul Mason, Thomas N. Woods, Amir Caspi, Phillip C. Chamberlin, Christopher Moore, Andrew Jones, Rick Kohnert, Xinlin Li, Scott Palo, and Stanley C. Solomon. Miniature X-Ray Solar Spectrometer: A Science-Oriented, University 3U CubeSat. *Journal of Spacecraft and Rockets*, 53(2):328–339, mar 2016.

[4] James Paul Mason, Thomas N. Woods, Phillip C. Chamberlin, Andrew Jones, Rick Kohnert, Bennet Schwab, Robert Sewell, Amir Caspi, Christopher S. Moore, Scott Palo, Stanley C. Solomon,

and Harry Warren. MinXSS-2 CubeSat Mission Overview: Improvements from the Successful MinXSS-1 Mission. *Advances in Space Research*, in press(-):–, feb 2020.

[5] Andrew Dahir, Ariel Sandberg, and James Paul Mason. Advancement, testing and validation of an innovative smallsat solar panel fabrication process. In *31st Annual AIAA/USU Conference on Small Satellites*, 08 2017.

[6] Sarah Withee, Gabriel Altman, Wesley Caruso, Charles Gillard, John Sobtzak, Brodie Wallace, and Kyle Wislinsky. Closing the deep space communications link with commercial assets. In *33rd Annual AIAA/USU Conference on Small Satellites*, 08 2019.

[7] Robert A. Marshall, Wei Xu, Thomas Woods, Christopher Cully, Allison Jaynes, Cora Randall, Daniel Baker, Michael McCarthy, Harlan E. Spence, Grant Berland, Alexandra Wold, and Elliott Davis. The aepex mission: Imaging energetic particle precipitation in the atmosphere through its bremsstrahlung x-ray signatures. *Advances in Space Research*, 66(1):66 – 82, 2020. Advances in Small Satellites for Space Science.

[8] Jesse Olson. University nanosatellite program - 20 years of education. In *33rd Annual AIAA/USU Conference on Small Satellites*, 08 2019.