VISORS Assembly, Integration, and Test Plan

Grace P. Krahn[1]

Georgia Institute of Technology, Atlanta, GA, 30332

The VISORS mission utilizes two 6U CubeSats to form a distributed telescope with a 40-meter focal length to study energy release regions in the solar corona. VISORS is an NSF-funded collaborative effort between a large number of universities, commercial suppliers, and institutions that is often challenging to manage, especially in a dynamic environment like AI&T. An AI&T plan can help give mission-level clarity on what must happen to assure spacecraft readiness and hardware robustness before delivery. This process is best visualized with an AI&T flow diagram, which can show sequential progression through the project objectives and event dependencies in an easily trackable manner. This paper describes a high-level nominal sequence of events to provide a big picture understanding of the AI&T flow. Breaking down each block further, each subsystem's progression from individual component development to integrated systems testing is visualized through colors and arrows. Different categories of testing are described and included on a subsystem and system level in the AI&T plan with objectives for each test outlined. Finally, lessons learned from AI&T are given to help the VISORS project planners to better estimate the time and personnel needed to plan for, write, and conduct procedures.

ADCS	=	Attitude Determination and Control System
AI&T	=	Assembly, Integration, and Test
AVI	=	Avionics
BCT	=	Blue Canyon Technology
CAD	=	Computer-Aided Design
COSMOS	=	Command and Control of Embedded Systems
CSIE	=	Compact Spectral Imager Electronics
DIA	=	Detector Instrument Assembly
DSC	=	Detector Spacecraft
EDU	=	Engineering Development Unit
FHL	=	Flight Hardware Lab
FHW	=	Flight Hardware
GNC	=	Guidance, Navigation, and Control
GT	=	Georgia Institute of Technology
GTRI	=	Geogia Tech Research Institute
HSA	=	Hosted Software Application
HW	=	Hardware
LASP	=	Laboratory for Atmospheric and Space Physics
LRF	=	Laser Range Finder
NASA	=	National Aeronautics and Space Administration
NSF	=	National Science Foundation
OSC	=	Optical Spacecraft
PAIB	=	Payload Avionics Interface Board
PCB	=	Printed Circuit Board
PROP	=	Propulsion
RF	=	Radio Frequency
SIR	=	Systems Integration Review

[1] Engineering Graduate Student, Space Systems Design Laboratory, 620 Cherry St. NW, Atlanta, GA 30332.

SW	=	Software
VISORS	=	Virtual Super-resolution Optics using Reconfigurable Swarms
XB1	=	BCT commercial CubeSat spacecraft bus
XLINK	=	Crosslink (intersatellite link)

I. Introduction

Virtual Super-resolution Optics using Reconfigurable Swarms (VISORS) is a National Science Foundation (NSF) funded university-driven mission designed to push the bounds of CubeSat precision formation flying technology. Its scientific goal is to take high resolution images of the solar corona by utilizing two independent 6U CubeSats to form a distributed telescope with a focal length of 40 meters, which is the distance from the optical instrument (like a lens) to the detector electronics. One spacecraft contains the optics needed to filter and focus the light while the other contains the detector instrument to capture the science images; these two vehicles are known as the Optics Spacecraft (OSC) and Detector Spacecraft (DSC), respectively. To properly photograph the sun, the detector in the DSC must look through the optical sieve mounted in the OSC while maintaining its position to no more than a few millimeters off the focal point. The two instruments must be aligned along their boresights to properly capture an in-focus image, as shown in Figure 1. The primary requirement for the VISORS mission is highly ambitious: to take a single high-resolution picture of the sun's corona to find energy release regions and successfully send it back to Earth. To accomplish this feat, VISORS must meet a set of four secondary mission requirements, which are also technology demonstration objectives. Directly taken from the VISORS Requirements Verification Matrix documentation, these secondary mission requirements are as follows:

- Control the formation to mm-level position accuracy when attempting an observation to collect imagery
- Demonstrate usage of an inter-satellite communication link to achieve autonomous maneuver-planning and execution
- Employ e/i-vector separation in all mission phases to enable passive collision avoidance and maintain near-proximity relative orbits
- Utilize on-board propulsion systems to execute maneuvers for relative orbit stationkeeping and formation reconfigurations

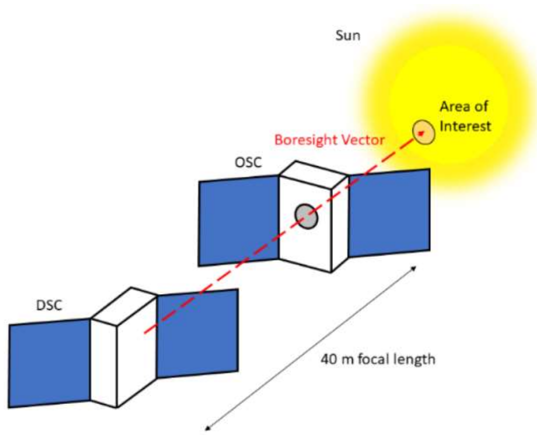

Figure 1. VISORS science mode configuration for capturing in-focus images of the solar corona [1].

The VISORS mission is unique in the satellite world for a variety of reasons. For one, its pair of satellites have been tasked with meeting these impressive technical requirements on a shoestring budget of around $5 million total (including launch costs). Compare this budget to the cost of what a large space borne telescope could cost; launched in December 2021, the James Webb Space Telescope has costed around $10 billion over the course of its lifetime so far [4]. Normally, if one were to engineer a single satellite with the same 40-meter focal length as the VISORS mission, it would take a budget on the same level as major NASA missions like James Webb to accomplish.

The cost savings of using two small satellites instead of one large one, however, has its price in technical difficulty and required technological development. This challenge is addressed by involving universities and institutions with expertise in various aspects of the mission to provide hardware and software for the mission. The number of collaborators (external to Georgia Tech) involved in the development is one of the other aspects that makes the VISORS mission unique but also incredibly challenging. Due to the complexity of achieving precise alignment of two distinct spacecraft on the millimeter level, the development of advanced GNC algorithms to manage their relative positions and alignment is required to perform this mission. Engineers at Stanford have been working on this mission-enabling technology for years and are developing it further for VISORS to demonstrate it in flight for the first time. An important connection needed to make the GNC algorithms work is for both spacecraft to be aware of the other's knowledge of its own state. On VISORS, this will be achieved through an intersatellite radio link – or crosslink (XLINK) - that conveys information about each satellite's GPS location, among other information. The GNC algorithms determine relative position through differential GPS measurements that are relayed to the other spacecraft through the XLINK radios, which are transmitted through patch antennas located on each side of the briefcase-shaped spacecraft. These XLINK radios operate with a combination of commercially supplied electronics that serve as the main radio and custom boards developed by Washington State University (WSU) that amplify and transmit the signal through six externally mounted patch antennas (per spacecraft). The software for the XLINK system is developed and tested, however, by yet another collaborator: the Ohio State University (OSU). Meanwhile, NASA Goddard is responsible for the design of the primary scientific instruments on the VISORS mission: specifically, the optical sieve and the detector (camera) instrument itself. The board assembly that controls the detector and stores the captured image data is called the Compact Spectral Imager Electronics (CSIE) and is designed by the team at the Laboratory for Atmospheric and Space Physics (LASP) at University of Colorado – Boulder. Together, the CSIE and detector form the Detector Instrument Assembly (DIA), which is entirely housed on the DSC.

VISORS not only employs universities and NASA, but also has contracts with commercial companies to supply extensive portions of the satellite. To command the subsystems, orient and navigate in space, and supply the basic communication and processing power to operate a satellite, a commercially produced spacecraft bus developed by Blue Canyon Technologies (BCT) was chosen. This bus – known as the XB1 – is about 1.5U in size, has been flight-proven in other CubeSats before and has a fully integrated suite of sensors for operating and navigating in low Earth orbit (star trackers, reaction wheels, GPS antenna, etc). BCT not only supplied the bus avionics, but also the chassis structure, ground communications radio, solar panels, and other related hardware. The open-source software known as Command and Control of Embedded Systems (COSMOS), developed by Ball Aerospace, commands the spacecraft [3]. BCT uses this software to control and communicate with all spacecraft that utilize their commercial buses. Another use of a commercial product is in a device designed to help align the two spacecraft for an image when in close-proximity science mode. A secondary OSC instrument, the Laser Range Finder (LRF) reflects back off of the DSC to refine the GNC algorithms' relative position estimates. Though the LRF is a commercial-off-the-shelf product, the engineers at NASA Goddard put it through its paces with a casing re-design and extensive environmental testing to ensure its survivability and robustness in space.

In addition to its primary role as the system integrators, Georgia Tech (GT) also is charged with developing two of the major payload subsystems. The OSC and DSC are equipped with 3D-printed cold gas propulsion systems that can produce delta-v's of a small enough magnitude to make the minor position adjustments needed to achieve the science formation. This technology is provided by GT, which has successfully demonstrated the same system but with a different form factor on the Artemis-1 CubeSat BioSentinel [5]. The Payload Avionics Interface Board (PAIB) is another crucial component being developed at GT that will route all the signals and provide power with the correct voltage and current limits to all payload subsystems. In the context of the VISORS mission, the "payload" refers to any subsystem that is not the BCT XB1 bus or related BCT-produced subsystems: the propulsion system, XLINK, PAIB, and science instruments. The PAIB interfaces with all of the payload subsystems but does so most extensively with the XLINK subsystem, which cannot function or turn on without it. This assembly of the XLINK boards and PAIB is known as the Avionics (AVI) Stack. GT is also responsible for the development of the Hosted Software Application (HSA), which is an application stored on the spacecraft bus that connects all of the payload subsystems together in software with the main XB1 software. It also stores Stanford's GNC algorithms and contains the VISORS-specific fault detection and management system, also developed by GT.

In summary, the VISORS mission is enabled by the extensive collaboration between university, commercial, and institution partners. Managing the flow of information between these various locations can be a challenge with so many collaborators. Maintaining two-way communication between the system integrators and subsystem providers is a task that must be approached with great intentionality; an extensive and holistic understanding of the system is key. As such, a short synopsis of all of the subsystems described previously that make up the VISORS spacecraft is provided to assist with developing this system-level understanding:

- XB1 – commercial CubeSat bus avionics and Attitude Determination and Control System (ADCS) suite that enables baseline satellite functionality
- AVI Stack – mechanical and electrical assembly of the XLINK and PAIB payload board stack
  - PAIB – PCB that routes and controls power and signals that pass between the spacecraft bus and payload subsystems for mission-specific satellite functionality
  - XLINK – radio and patch antenna assembly that forms the intersatellite link between the OSC and DSC
- Propulsion ("prop") – cold gas propulsion system that is used to perform maneuvers to establish and maintain a precise formation of the two satellites
- Science Instruments:
  - LRF (OSC exclusive) – laser ranging device used to increase accuracy of the GNC algorithm during science mode
  - Optical sieve (OSC exclusive) – passive optical device used to focus light on the detector
  - DIA (DSC exclusive) – detector camera device and accompanying door mechanism (similar to a lens cover) to captures the high-resolution science images
  - CSIE (DSC exclusive) – avionics that command and control the DIA and temporarily store the images after capture
- HSA (software only) – application installed on the XB1 that contains everything needed for the spacecraft to operate autonomously (including fault detection and management)
- GNC (software only) – algorithms needed to autonomously maneuver and control the spacecraft when performing precision formation flying operations

II. Background and Motivation

An Assembly, Integration, and Testing (AI&T) plan describes the order in which assembly, integration, and test tasks need to be completed and relates it to the overall timeline. It elaborates on what margin is available and where, as well as establishing the major choke points and challenges to pay extra attention to. In addition, it gives a basic description of the testing campaign for the project and describes each test's overall goals. The AI&T plan can be used as a project management and planning tool to direct resources and properly manage schedule delays and margin. It can also be used as a communication tool to clarify project interfaces and avoid misunderstandings between teams.

While the term "AI&T plan" can be used to describe lower-level assembly and integration plans, the focus for this paper will be on describing the higher-level processes plan for the flow of activities. These lower-level assembly and integration details have already been fleshed out through project reviews and documentation like the VISORS project Systems Integration Review (SIR) in April 2023. In this review, an integration plan was presented that outlined the current state of the project's AI&T preparation work, what the plan's major risks are as well as ways to mitigate them, and a detailed step-by-step overview of the main steps of spacecraft integration. While the AI&T schedule has changed much since this review was held, the SIR risk analysis is still mostly accurate. The mitigation strategies developed in this review have been partially or fully implemented since their inception. The SIR will remain the primary source of information for AI&T risk identification and mitigation. The full VISORS spacecraft integration procedures that resulted from the SIR review are extremely comprehensive in how the actual assembly of the VISORS bus and payloads will occur. For the purposes of this paper, *spacecraft integration* refers to the assembly process of the hardware (HW) at a vehicle level. The spacecraft integration procedures have been reviewed extensively by both GT and NASA Goddard engineers on the VISORS team; they are currently undergoing final revisions. This existing body of work includes details like hardware equipment lists, Computer-Aided Design (CAD) views at each step, and torque specifications (among others). The procedures describe the literal nuts and bolts of the spacecraft payload integration process within the payload cavity of the BCT XB1 bus and chassis. Because so much work has already been put into the specifics within the integration process, it is not necessary to rehash them here besides presenting an overall summary of the major payload integration events.

Another set of important terminology to define includes the concepts of *functional tests* and *interface tests*. The terminology is based on the types of testing categories described in Table 2 in [2]. It is important to consider the differences between how these terms are originally utilized and the slightly different but specific way in which they are applied in this VISORS AI&T plan. Here are the specific meanings to these tests when applied to different stages of VISORS testing:

- Subsystem functional testing: determining whether the subsystem operates as expected and meets basic functional requirements when it is tested with a non-flight-like external power supply, component ground testing setup, and with no other extraneous components connected to the subsystem
  - Can also be referred to as acceptance testing
  - Interfaces are non-flight-like since the test's goal is to verify the component's functionality at the subsystem level (i.e. as the subsystem supplier tested it)
  - Subsystem functional testing done at the supplier and also at the system integrator (GT) to ensure no changes in functionality have occurred due to transportation or environmental factors
  - Gives GT engineers the chance to work with the subsystem and understand its function before moving onto interface testing, where comprehensive knowledge of the system and how each part operates is crucial to successful troubleshooting
- System-level functional testing: determining whether the full system operates as expected and meets basic functional requirements when tested at the integrated testbed or fully integrated spacecraft level; depending on the test, an external power supply may or may not be used for the integrated spacecraft
  - All interfaces besides those needed to collect data and provide power are flight-like
  - Performed at both the system integrator and the launch vehicle to ensure no changes in functionality have occurred due to transportation or environmental factors
  - Allows GT engineers to characterize the behavior of the full system
- Interface testing: testing that occurs before, during, and after connecting two components or systems together to verify that the connection between them is functioning as expected and does not change the component's expected function or response
  - Includes safe-to-mate testing, verifying that data packets are being sent and received in the expected format and locations, and confirming that commands are displaying the right response when connected to the new interface
  - A catch-all term for testing that occurs between subsystem and system-level functional testing
  - Incorporates tests verifying the electrical, mechanical, and software interfaces between two components

III. VISORS Assembly, Integration, and Test Flow

An AI&T flow is a visualization of the high-level plan that not only describes the specific activities, but also gives a sense of the dependencies, sequential timeline of events, and resources needed to complete each step of the flow. This flow is an integral part of the AI&T plan because it visually conveys the "big picture" of what needs to occur to complete the project in a way that's easy to follow and track. The AI&T flow diagram will break down the higher-level categories like system-level testing into its basic components and attempt to show their interactions with other components of the same or higher level. The goal for the AI&T flow discussion is not only to provide an ideal order for the events within to occur, but also to provide enough information about component dependencies to manipulate the default flow if need be as long as a component is not blocked by these sub-connections.

First, the overall AI&T process flow will be established at the highest and most abstracted level. At this level, the order in which the events occur is fixed and cannot be re-arranged since each block encapsulates a broad category of activities. The VISORS simplified high-level AI&T flow is shown in Figure 2 below. The boxes marked by the blue flag can all *theoretically* be started simultaneously, as they have no major blockers. The testbed structure and accompanying Electrical Ground Support Equipment (EGSE) can be designed and manufactured as soon as the physical dimensions and primary connections between other subsystems are finalized, which should be early in the design phase. Adjustments to the physical testbed structure can be made with quick turnaround if absolutely necessary through student use of the GT Aerospace Engineering machine shop or employing its professional machinists. The Engineering Development Unit (EDU) creation for each subsystem will oftentimes occur first or as a subset of the flight subsystem development. An example of sequential EDU-then-flight unit development is the PAIB subsystem, which had an EDU version that was manufactured and tested fully before the flight board design was started. The flight boards were designed based off testing of the first iteration EDUs, which informed the need for some major changes to the second iteration EDU and flight board were designed and manufactured simultaneously. Alternatively, the BCT bus EDU was, for all intents and purposes, started at the same time as the flight version but delivered about a year before. In either case, the EDU is most often completed or is available for testing before the delivery of the flight version of the subsystem.

Having the EDU available before the flight unit is essential because of the extensive interface testing that goes into

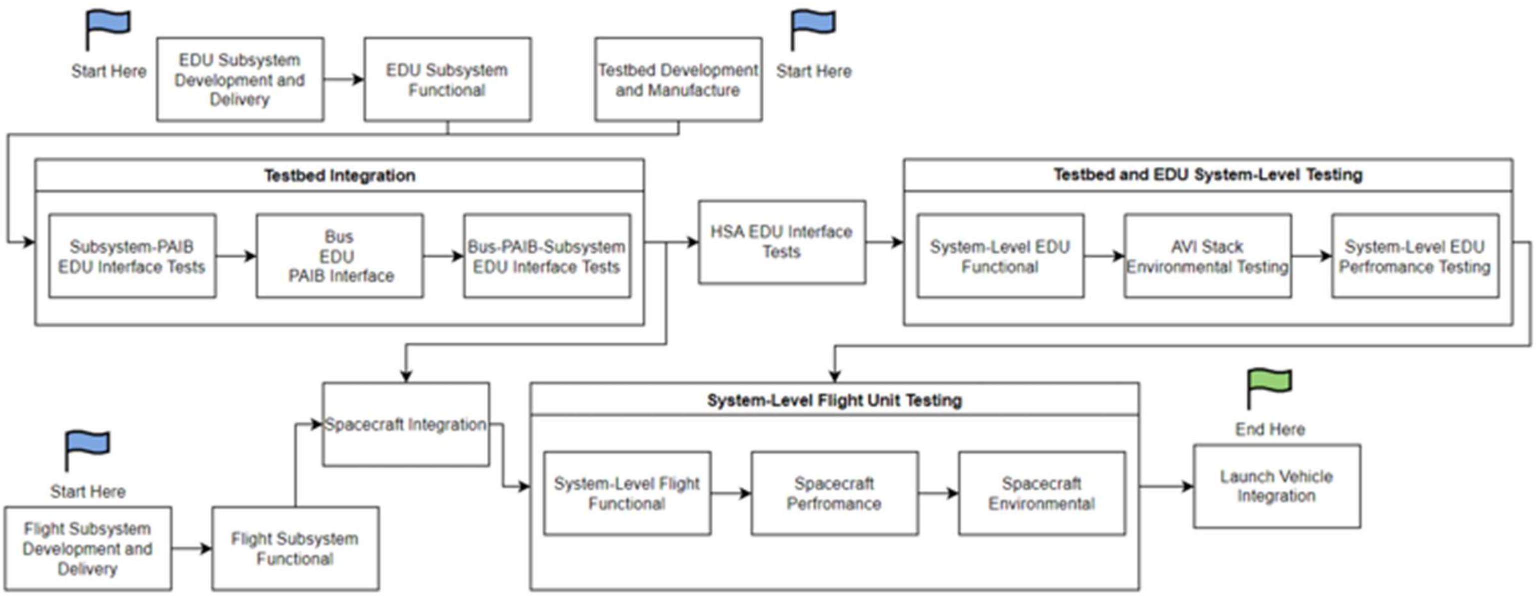

Figure 2. Simplified high-level VISORS AI&T flow.

the Testbed Integration parent block that can prevent integration from occurring if not given enough time. These interface tests often reveal compatibility issues that were not apparent before and can require extensive troubleshooting time. The Spacecraft Integration box is blocked until these testbed interfaces can be tested and verified because of the non-zero chance that hardware changes are necessary to make the connection work properly. With the order shown in Figure 2, the flight system development and functional chain becomes a blocker at this high level only when Testbed Integration is complete and Spacecraft Integration is ready to start. The System-Level Flight Unit Testing is only available after all tests have been verified through Testbed and EDU System-Level Testing. If possible, no tests should be run on flight hardware (FHW) without first testing them on EDUs to ensure no harm comes to it from running an erroneous script or accidental fatal command. It should be noted that the blocker to performing system-level EDU tests is first verifying that the HSA interfaces with all payload subsystems are properly configured and connected. Without ensuring that the HSA is functioning as expected, a full functional with all components in the loop should not be attempted on the testbed with much expectation for success. Once this full functional is performed on the testbed, the AVI Stack EDU can be sent off to environmental testing before the System-Level Performance Testing starts. This will be discussed further in the related subsection.

Now that the highest-level discussion is complete, each block will be broken down into smaller components that have a general defined order but can be more easily manipulated within each large block if necessary. Building this flexibility into the AI&T plan discussion is important for adapting to inevitable delays that occur throughout the process. Delays that may require re-ordering of the elements of the AI&T flow can be caused by the following reasons:

- Subsystem delays that impact the original spacecraft integration window for that subsystem
    - Subsystem provider delays in payload subsystem deliveries
        - Currently, the XLINK, PAIB, and propulsion systems have been delayed in a manner that affected what order they were supposed to be integrated in
    - Delays after subsystems do not perform functions as expected when tested on its own
    - Delays after the subsystems have been received due to compatibility issues between components
        - Problems discovered during interface testing often require time-intensive troubleshooting, forcing a delay in subsystem integration readiness

- Resource availability may push an AI&T event forward or backward in the timeline
    - AI&T personnel availability
        - Impacted heavily by the academic environment (i.e. midterms or final exams)
    - Test equipment availability
        - Thermal vacuum chambers (TVACs) or vibration tables for environmental testing cannot usually be booked or changed last-minute
        - When a certain test occurs is often fixed by the availability of these limited resources
    - Supplier availability
        - If an issue occurs that requires a subsystem contractor/supplier to be involved, the integration team will need to work around their schedule or wait for their responses to resolve issues
- Troubleshooting of AI&T issues may impact whether a test or event can be completed as planned
    - Other tests or activities may be able to run in parallel with issue troubleshooting to maintain forward schedule progress

The sub-components that make up the high-level blocks shown in Figure 2 will be described in detail in the following subsections. To properly understand these more detailed diagrams, the reader must move from left to right and follow the flow of the AI&T plan. The blocks inside of the larger containers will show operations or activities that can be done in parallel through relatively vertical stacking of boxes that are not connected by arrows. Sequential activities will be demonstrated through arrow connections that point in the direction of the dependent operation. Any block that has an arrow leading to the edge of a large block or off into empty space indicates that that completion of that activity is gating access to the next large block or section of the diagram not shown. It should be noted that many of the diagrams contain tests three categories of blocks: those relevant to the OSC only, those relevant to the DSC only, and those relevant to both. If the reader would like to view the AI&T flow diagrams in the context of either the OSC or the DSC, then it is only necessary to complete the activities applicable to that particular spacecraft (reference the introduction for a comprehensive list). Sub-component blocks are colored to show one of two things:

1) Color indicates that the block is related to a particular subsystem AI&T thread and is dependent on the previous sequential blocks in the chain being completed
2) Color indicates that the block is dependent on a previous simplified high-level block being fully completed

A legend showing what each color means will be included on each diagram for ease of use.

A. EDU Subsystem Delivery and Testing

The first high-level parts that will be broken down in greater detail are the top three blocks of Figure 2, which can be lumped together and labeled as a single EDU Subsystem Delivery and Testing block. As described earlier, the VISORS spacecraft can be broken down into a handful of distinct *active* (or powered) subsystems: the XLINK intersatellite link radio, PAIB payload signal and power router, cold gas propulsion system, BCT-supplied satellite bus, LRF laser ranging instrument (OSC only), DIA camera instrument (DSC only), CSIE camera storage and control avionics (DSC only), HSA payload software, and the advanced GNC algorithms that enable the precision formation flying maneuvers. One could consider the AVI Stack to be its own subsystem, just at a slightly more integrated stage than the others. The primary goal of the subsystem EDUs is to have units that electrically (and physically, but not always required) function the same as the flight units so that the expected behavior of the subsystem and the integrated spacecraft can be accurately characterized though software, functional, and interface testing without endangering FHW. In addition, the integrated spacecraft testbed will eventually be used to prepare for and troubleshoot issues that arise during VISORS mission operations.

To prepare this testbed and enable the variety of important duties it will perform throughout the AI&T process, all subsystem suppliers must have some manner of EDU that is developed and delivered to GT for testing. In addition to the EDUs already mentioned, testbed EGSE must be designed to adapt the subsystems to work with the non-flight-like interfaces and Ground Support Equipment (GSE). In the case of VISORS, an adaptor board is required to enable the bus EDU connector to plug into the PAIB with a different connection than the one that will be present on the flight bus and flight PAIB. The testbed also needs to be intelligently designed with structure to securely hold and organize all of the EDUs in a manner that makes moving and testing easier. All of this can happen alongside the development of the rest of the EDUs, once the physical dimensions and key EGSE interfaces are known.

The development of the subsystem EDUs, the software required to operate them, and testbed EGSE/GSE are represented in the blocks on the left-hand side of the split large box in Figure 3. This graphic had to be split into two separate side-by-side columns but is meant to be read as one tall stack of left-to-right process flows. Following the development and delivery of the EDUs to GT, functional tests will be performed on all subsystems. These tests attempt to mimic the procedure and setup that the subsystem suppliers use to check out the full functionality of their system. A particular detail to notice is that after XLINK SW development is completed, the XLINK EDU EGSE is shipped from OSU to GT where it will be integrated with the PAIB EDU to form the AVI Stack EDU. An arrow leading from the PAIB EDU Functional block to the AVI Stack EDU HW Integration step shows that this can only be completed after the PAIB EDU has undergone a full functional and has been verified to route power and data as expected by the GT electrical engineers. Note that the change in color between the XLINK EDU and AVI Stack EDU blocks is meant to show that any future blocks that have the darker orange-red color - which is a combination of the PAIB and XLINK colors - will be dependent on the AVI Stack development. Steps dependent on the XLINK patch antennas only, however, will still be displayed in the light red color to show that it is independent from the AVI Stack. Finally, any blocks that still use the light orange color are referencing specifically the PAIB part of the AVI Stack assembly, independent from the XLINK system. The light orange PAIB-Testbed Adaptor Board Integration box gates the progression from this set of activities in the AI&T plan to the next step in the simplified high-level diagram. Without completing this critical step, the integration of the testbed cannot begin.

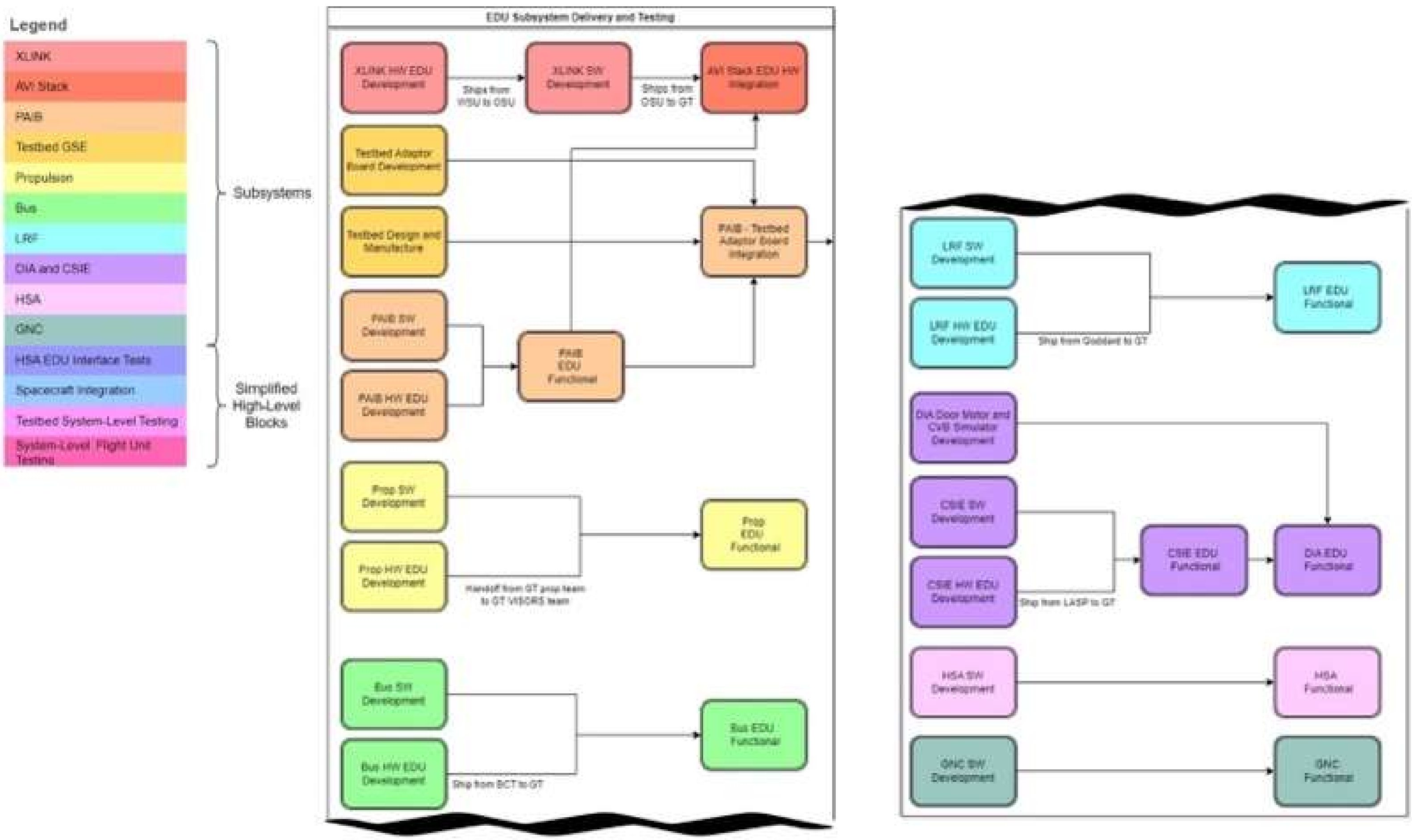

Figure 3. EDU subsystem delivery and testing flow.

B. Testbed Integration

Once the PAIB and testbed adaptor board have been integrated into the testbed structure, the PAIB EDU Interface Tests can begin in Figure 4. A subset of activities within the Testbed Integration umbrella, these PAIB EDU interface tests are all parallel paths with one another. That means that the only gates stopping all of these from occurring (other than the fact that there is only one of the PAIB EDU) is how far along each subsystem is in the previous EDU Subsystem Delivery and Testing high-level blocks: more specifically, whether the subsystem's functional test has been successfully completed. These PAIB EDU interface tests are intended to occur with an external power source to power the PAIB EDU and only the subsystem of interest attached to it. This will verify that the PAIB EDU can successfully power the subsystem and can effectively limit the current and provide the right voltage for that subsystem. It is essentially just conducting a pared-down version of the subsystem functional (done with an external power supply) while connected to the PAIB instead. Once the XLINK version of this test has been completed, the XLINK EDU Functional can occur. Unlike the other HW-based subsystems, the XLINK cannot operate independently of the PAIB; therefore, it is essential to complete that EDU interface test before the functional test of the subsystem can occur.

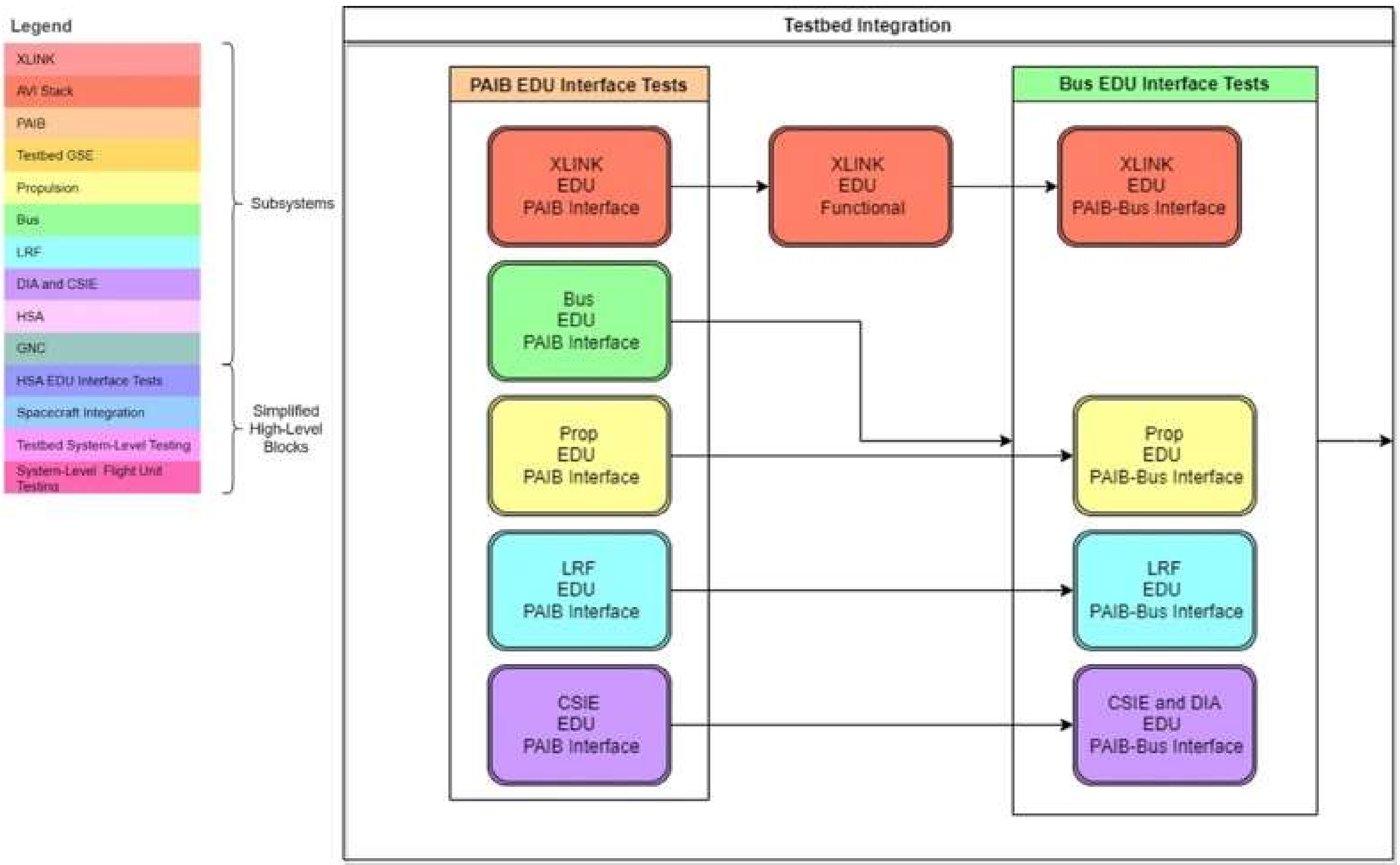

Figure 4. Testbed integration flow.

The Bus EDU PAIB Interface test enables the next suite of testbed integration tests to be performed: the Bus EDU Interface Tests. This dependency is shown visually by both the small-to-medium-box arrow and the green-colored title header in the box on the right. Since the PAIB EDU controls all the payload subsystems, it is the first subsystem that the bus needs to be able to successfully interface with and command to fully integrate all the EDUs. Once the engineers can tell the bus to send a COSMOS command to power and operate the PAIB EDU as designed, the interface has been verified. Then, the other subsystems can be plugged in one at a time (as in, only one subsystem plugged into the PAIB EDU at a time) and one by one the COSMOS-Bus-PAIB-subsystem interface chain can be tested and verified. It should be noted that each of these individual subsystem Bus EDU Interface Tests also rely on the previous link in the chain – the subsystem's accompanying PAIB interface (or the EDU functional for XLINK only) – being successfully completed before these bus-in-the-loop tests should be attempted. Only once all Bus EDU Interface Tests have been completed can the testbed be considered fully integrated, which then unblocks both of the following simplified high-level blocks: Spacecraft Integration and HSA EDU Interface Tests.

C. Flight Subsystem Delivery and Testing

On a parallel process path to the EDU Subsystem Delivery and Testing high-level block – albeit with later completion dates – is the Flight Subsystem Delivery and Testing box in Figure 5. Internally, this high-level block functions very similarly to its EDU counterpart, but with a few key differences. Note the lack of a SW development box for each subsystem like in the EDU section; this is because the EDUs are being utilized to write and develop the SW so the FHW is protected from any untested and accidentally damaging code. Another key difference is that all the subsystem suppliers – with the marked exception of the PAIB/XLINK/AVI Stack – will be environmentally tested before being sent to GT. After the subsystems have been received by GT, they will undergo functional testing just like the EDUs. The flight AVI Stack will not undergo environmental testing, but the AVI Stack EDU will. Due to the criticality of the PAIB and XLINK to the mission goals, it was deemed a greater risk to put the flight AVI Stack through the extra punishment of subsystem environmental testing on top of the system-level environmental tests. As such, the AVI Stack will only need to be conformally coated before it is ready for integration. One other difference between the EDU and flight system subsystem test blocks is the presence of the Bus Flight Acceptance and Aliveness test before the Bus Flight Functional. To confirm that the BCT bus and all the basic functionality was undamaged during the shipping process, a full acceptance inspection and very simple aliveness test was performed on each bus upon receipt at GT. This inspection process documented every existing physical flaw observed on the bus and confirmed with BCT that none were mission-impacting. The difference between the bus aliveness script and the later full bus functional is similar to the difference between checking whether a person has legs compared to whether they can run a mile. The arrow stemming from the Bus Flight Functional block indicates that it is the non-negotiable critical path item that must be completed before the Spacecraft Integration flow can start.

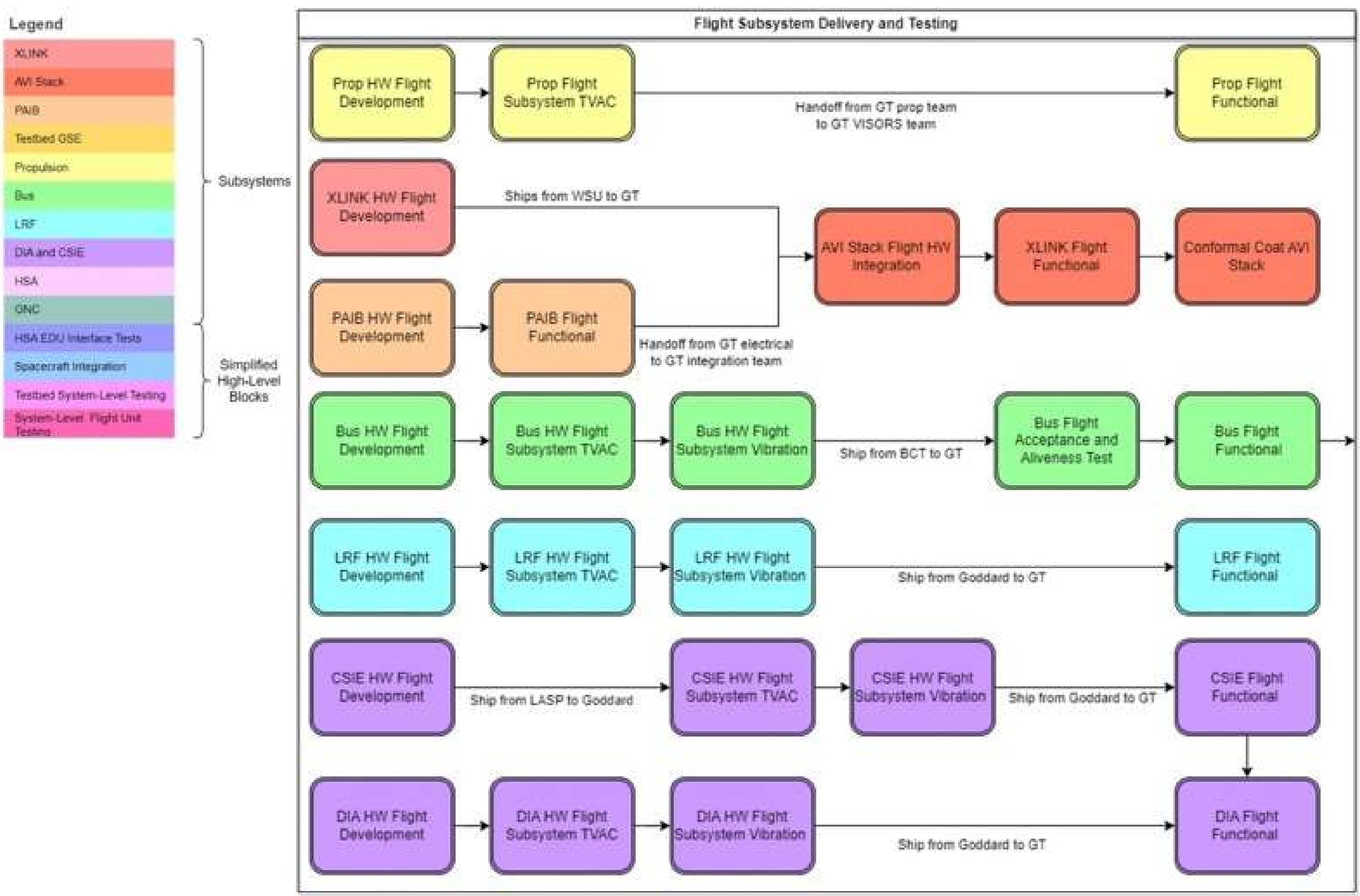

Figure 5. Flight subsystem delivery and testing flow.

D. Spacecraft Integration

Once the Testbed Integration high-level block and the Bus Flight Functional box in the Flight Subsystem Delivery and Testing high-level block are complete, the option to begin Spacecraft Integration opens in Figure 6. How much total progress that can be made within, however, depends on which subsystems have completed their flight functionals (which in turn is dependent on subsystem delivery to GT). There is one straightforward and relatively easy path through that can be achieved if all the flight subsystems are ready when their respective block comes up. It is the path drawn by the primary chain through the center of the high-level block, where the first step is installing the fully integrated AVI Stack into the payload bay. This is the ideal order of integration because of the difficulty of connecting the other payload connectors to the AVI Stack once it is installed in the chassis. It is much easier to plug in the harness already attached to the AVI Stack into the various subsystems after they are later installed into the bus. However, if flexibility is needed in the schedule, this order can be arranged with the cost of added difficulty of execution. There is one configuration in particular that could be especially arduous and should be thoroughly tested with the VISORS 3D-printed simulator before attempting. For either the OSC or the DSC, installing the propulsion system into the chassis without first installing the patch antennas and associated Radio Frequency (RF) cables could face a difficult challenge as installation access is cut off to one side's patch antenna. For the OSC, the LRF is fairly flexible since it has minimal connectors that should be easier to install on the AVI Stack should it be installed after the instrument. However, the OSC Sieve Installation is not flexible whatsoever and must be performed as the final action of integration. This is because the optical sieve is incredibly delicate and installing it anytime sooner would introduce unnecessary risk of damaging it. On the DSC side, the detector is similarly inflexible in the location of the DIA Installation and CSIE Installation blocks. This is because access is greatly restricted for harness staking of the other subsystems once the DIA goes into the spacecraft. Due to some of the harnesses connecting the DIA to the CSIE, the CSIE Installation must occur simultaneously with DIA integration or after.

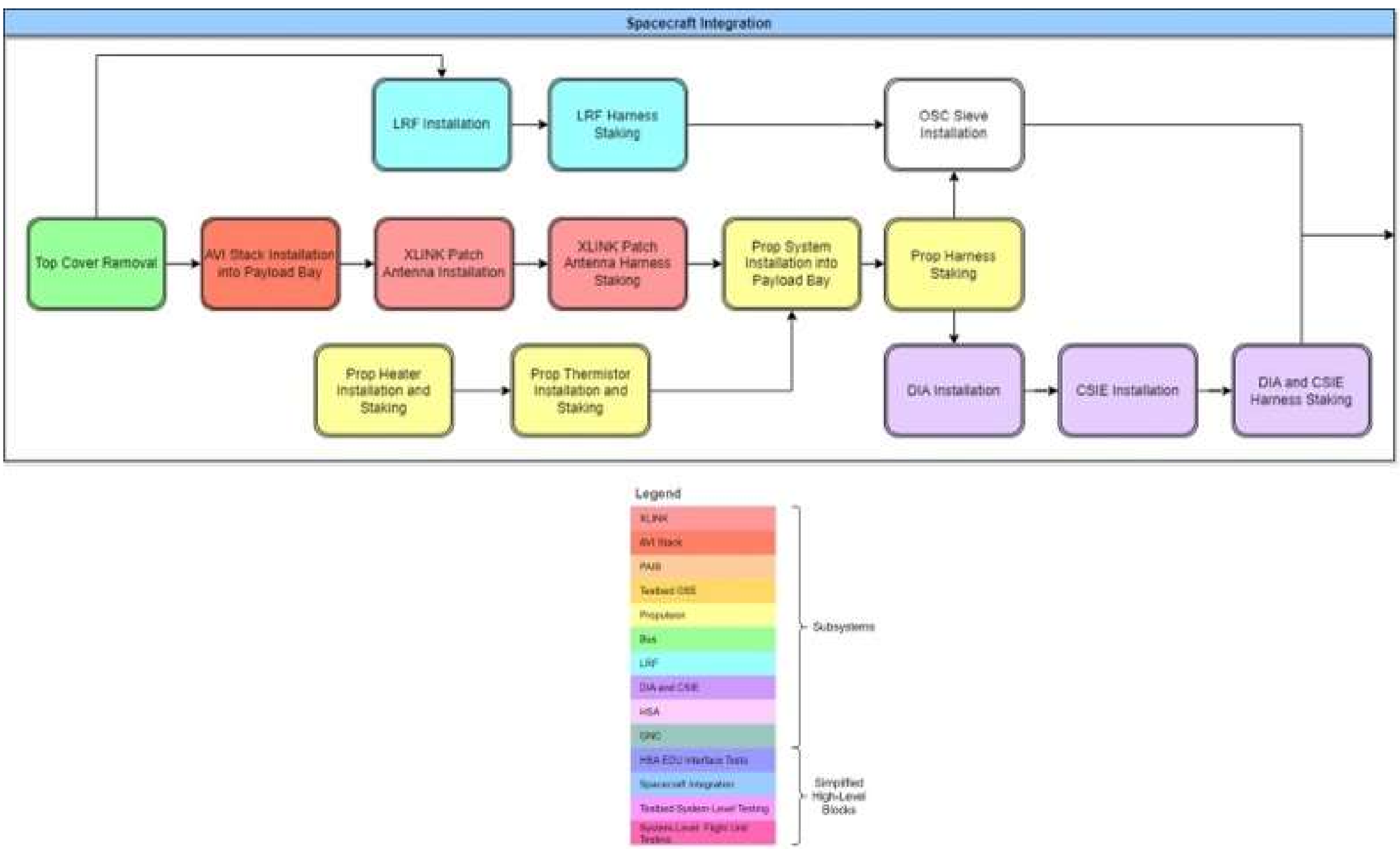

Figure 6. Spacecraft integration flow.

One other important point to note is that the Top Cover Installation and +/- Y Solar Panels Installation do not occur in its seemingly logical place after the science instrument installation within the Spacecraft Integration high-level block. These two sequential blocks are best done later after system-level testing of the spacecraft but before environmental has occurred because it potentially saves the GT engineers a lot of time and risk. A more detailed explanation can be found in the System Level Flight Unit Testing section. It is possible to include either just the Top Cover Installation or both blocks at the end of the current high-level block if it indeed becomes necessary, but it is likely not the lowest risk nor the fastest option.

E. HSA EDU Interface Tests

Before system testing can truly begin, however, it will be necessary to go a bit backwards in the flow to the HSA EDU Interface Test block in the simplified high-level flow as shown in Figure 7 below. In the Testbed Integration block, the interface between the command sender, COSMOS, the bus, the PAIB, and each subsystem was demonstrated. However, the key to the success of VISORS is the satellites' ability to function autonomously since there are very few ground contact opportunities and even fewer that are long enough to perform much real-time commanding. This functionality is enabled by the HSA and GNC SW subsystems, which are hosted on and interface with the BCT bus SW. Because the functionality of these two pieces of software are so critical to the mission, it is necessary to troubleshoot all interface issues that prevent it from responding autonomously in a flight-like manner before a true testbed system-level functional test can be run. The purpose of these HSA EDU Interface Tests is to test out the HSA's capabilities and ensure that it is able to properly interface and control all subsystems as designed. Each subsystem block is dependent on the previous subsystem EDU box in the chain being completed before it can be tested with the HSA. Only once all of these tests are completed can progression in the testbed side of the AI&T flow occur.

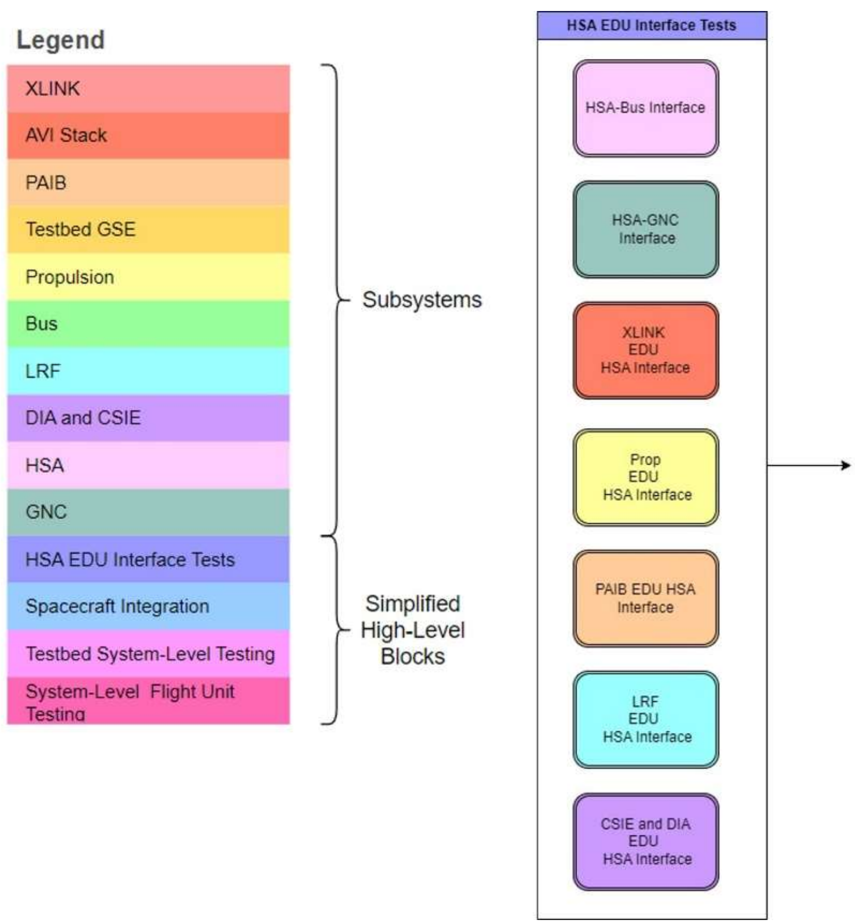

Figure 7. HSA EDU interface tests flow.

F. Testbed and EDU System-Level Testing

Once all the HSA interfaces have been verified, then Testbed and EDU System-Level Testing can begin in Figure 8. The first order of business is to test the full System-Level EDU Functional for the first time. This functional test is where system-level flaws in the VISORS design may first become apparent. Assuming these issues are able to be mitigated and the test is conducted successfully, the team has a decision to make. With the AVI Stack EDU's functionality verified fully now at both the subsystem and system levels, the likelihood of having to make any hardware changes to its design is low. At this point, the AVI Stack EDU can either be removed for conformal coating and environmental testing or it can continue to be utilized for the other system-level testbed tests first before entering environmental testing. The current version of the AI&T flow has environmental testing occurring after just the first test rather than at the end of the testbed campaign so to reduce the impact that any issues discovered during environmental testing may have. This reduces the risk of being stuck in the unenviable situation of not being able to make changes to the AVI Stack design in time to make the launch date. However, it also sets back the ability to enter the System-Level Flight Unit Testing high-level block by the amount of time it takes to perform the environmental testing. When the time comes to make this decision, these risks should be evaluated more closely for which approach is best based on the team's situation.

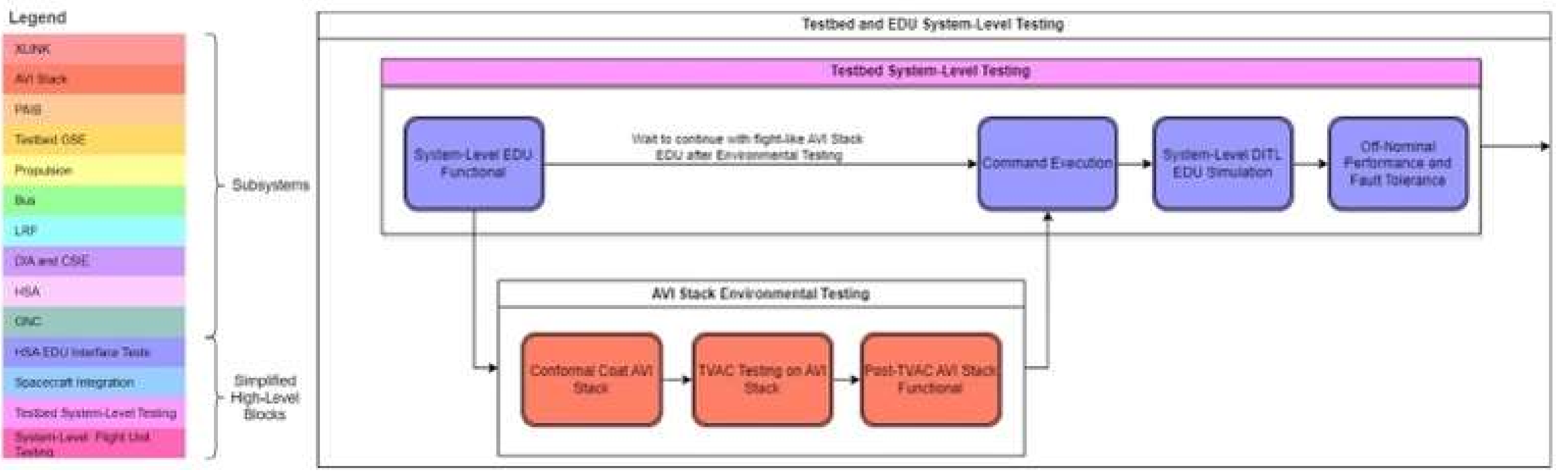

Figure 8. Testbed and EDU system-level testing flow.

The other tests to be conducted on the testbed include the command execution, Day-In-The-Life (DITL), and off-nominal performance tests. All of these tests will also be run on the integrated flight unit, but they must be run on the testbed first to avoid accidental FHW damage. The Command Execution test consists of sending every possible (non-fatal or non-dangerous) command to each subsystem to record its response and ensure it meets expectations. If possible, the first time a command is run on the integrated system should not be in space. Whether or not it is feasible to actually send each and every possible safe command is another story. This test can be scaled back depending on whether the risk posed by the schedule and meeting the launch date is greater than the risk of not testing out certain commands. At the very least, a list of essential commands that are most likely to be sent during the mission should be compiled and prioritized during this test.

The System-Level DITL EDU Simulation tests out the nominal performance of the satellite testbed in a situation that it will likely see on orbit. This simulation will be a comprehensive test of the autonomous functioning abilities of the satellite and will be mostly a software-driven test. It would be designed to put the testbed through a simulated science campaign to see how the satellite might behave when it thinks it's in orbit. A primary test objective would be to test out the formation-enabling capabilities of the satellite, especially the GNC algorithms and how they respond to certain simulated nominal stimuli. The DITL test may not actually simulate a full day or only one scenario; it may simulate a couple different scenarios to test out the satellite system's reaction to various nominal orbits or operation modes. The specific scenarios of this test should be developed and scripted through close collaboration with the operations team. Understanding how the satellite behaves nominally will be invaluable information for the operations team to better prepare, so they should be a driving force behind this test.

Finally, the Off-Nominal Performance and Fault Tolerance tests will allow the team to better understand how the satellite testbed behaves when at its limits. This test is designed to put the autonomous fault detection and management components of the HSA to the test in a controlled, hardware-in-the-loop environment. One of the most consistent killers for university CubeSats are fault protection and/or boot loops. A way to avoid this is to ensure the fault protection system "dead-ends," or arrives at a safe and well-defined fault management end condition that does not continue to loop back on itself again and again. By testing what happens when the testbed receives faulty data, an incorrect or intentionally corrupted command, or out-of-range inputs, the team can verify that the fault management system dead-ends in a safe manner for the satellite. In addition, these tests will reveal information about what the threshold is for making a component or system no longer work as intended. This information will also be critical for the operations team, so they should once again be heavily involved in writing this test.

Once one or all of these testbed system level tests are complete, their FHW counterparts can be run on real satellites in the next simplified high-level block. While it is preferrable to have all of these testbed tests complete before starting to run any single one of them on flight hardware, it is also possible to parallelize the troubleshooting of one test on the testbed with running an already verified testbed test on the FHW. Depending on the remaining schedule margin, parallelizing these tests may be required to make the launch date.

G. System-Level Flight Unit Testing

After a specific test has been run successfully on the system testbed, its twin in the System-Level Flight Unit Testing block in Figure 9 can then be run on FHW (hence the lighter pink coloration corresponding to a dependency on the Testbed System-Level Testing). Depending on the amount of margin available in the schedule, tests like DITL may be able to utilize the testbed to "spoof" the presence of the other spacecraft in the formation to better test the GNC algorithms and other formation-enabling capabilities of the mission [2]. While this would lower the risk that the precision formation flying technologies would fail, it may also increase the chance of missing the schedule launch date. While it is better to plan to perform these more involved tests so the documentation is ready, whether or not it should actually be performed should be weighed against the schedule risk of the mission.

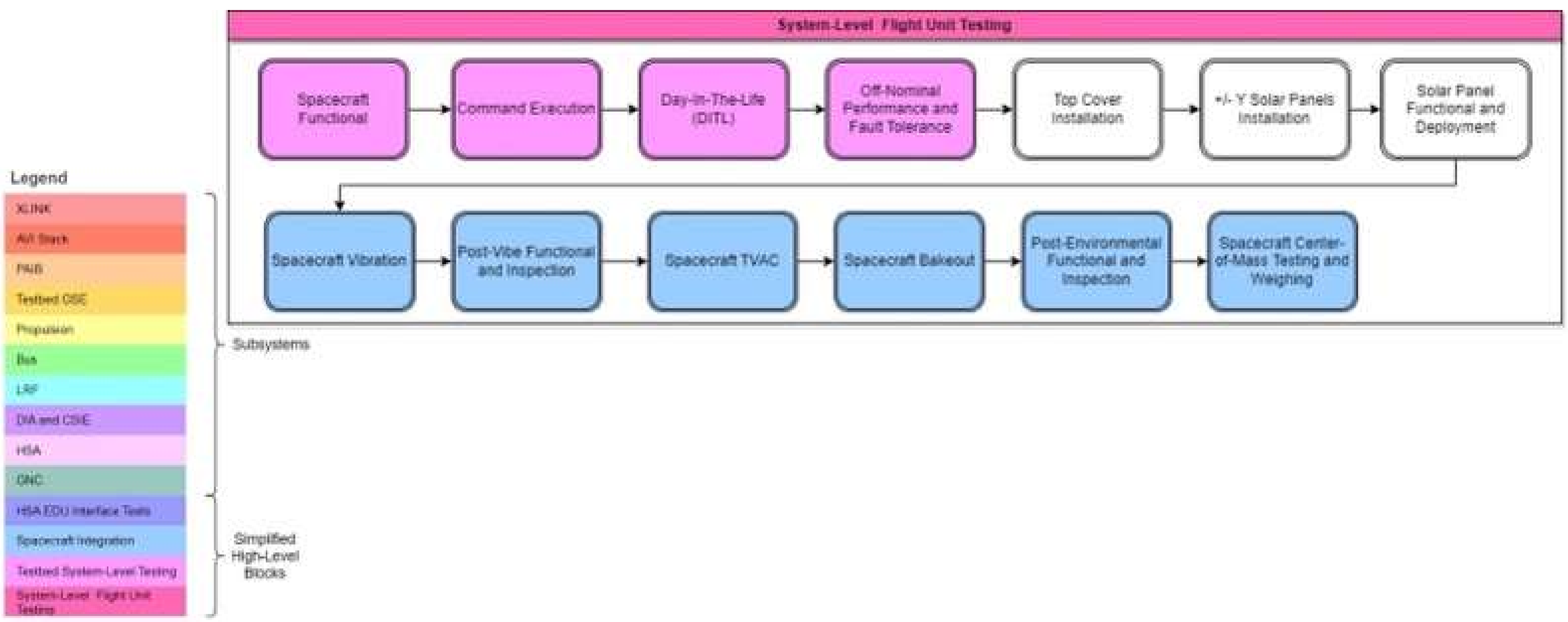

Figure 9. System-level flight unit testing flow.

After System-Level functional and performance testing is complete, the steps that were taken from the Spacecraft Integration block can be run. The problems that are addressed by adding the top cover and solar panels on later rather than during integration are listed below below:

- SW changes are likely to occur during system-level testing and some software needs to be updated manually (i.e. needs access to the internal payload cavity to upload new version to the boards directly)
    - Leaving the top cover off gives more time for the SW team to finalize their SW changes and allow for the FSW updates to the PAIB

    - While an over-the-air update system is in the works, the PAIB SW at least will still need to be updated manually
- Top cover bolt cycling increasing risk of a sheared, stuck, or galled screw
    - If the top cover does need to be uninstalled at any point, it poses a non-trivial galling risk for the top cover bolts by cycling the helicoils
- Solar panels are extremely delicate and easy to break
    - Integrating them right after system testing and before environmental testing exposes them to less risk of accidental cell damage
    - If the solar panels are installed when it is determined that the top cover needs to be removed, it's a very time-intensive and risky process to repeat solar panel removal and installation

Some disadvantages of installing the solar panels and top cover later are listed below, along with suggested solutions:

- Solar panel functionality was not verified in the Spacecraft Functional
    - Solar panel integrity can still be verified in a separate and targeted solar panel functional and deployment test after installation
- System-level flight unit tests won't be flight-like due to the lack of solar panels
    - Solar panels will have no purpose during the system-level tests because the power generation for all tests will come from an external power source regardless
    - No accurate real-life sun simulator resource exists to mimic their on-orbit power generation in a test of the FHW

Spacecraft-level environmental testing follows by the top cover and solar panel installations/tests. Mechanical problems uncovered by environmental testing are often a source of last-minute schedule delays. It is more likely to uncover these mechanical issues during vibration ("vibe") testing, which is one reason to put the test first in this part of the flow. This will give the team more time to respond to any mechanical defects uncovered during the Post-Vibe Functional and Inspection. In addition, the vibe testing will be performed at the Georgia Tech Research Institute's (GTRI) Cobb County facility, which is not a clean facility like the Flight Hardware Lab's (FHL) cleanroom is. It will be necessary to perform a thermal vacuum bakeout to reduce the outgassing risk on orbit after visiting this non-clean facility. By exposing the spacecraft to a certain set of extreme temperatures and pressures that it will be exposed to in space for a set period of time, the risk that outgassing will produce random torques or other disturbances to the spacecraft is reduced. This Spacecraft Bakeout is most efficiently performed during or at the end of Thermal Vacuum Chamber (TVAC) environmental testing, where the temperature and (lack of) pressure are already right where they need to be to reduce the outgassing hazard of the vehicle. For this reason, it makes the most sense to place vibe testing first and follow it with TVAC testing and bakeout, which is of course subsequently followed by a post-environmental functional to confirm that all systems are functioning nominally on the spacecraft after putting it under stress.

A final, crucial test must be performed before the spacecraft is ready to be packed up for launch integration: the Spacecraft Center-of-Mass and Weighing test. By measuring the weight of each spacecraft, the requirement to stay under 12kg can be confirmed satisfied (or not). The location of the Center of Mass (CoM) of the spacecraft has large impacts on the GNC algorithms, which need to contain the most accurate CoM information possible in order to achieve the precision pointing and maneuvering benchmarks that are laid out by the mission requirements. At this stage, a rig for determining the CoM of the spacecraft has yet to be designed or found; however, the most likely place to find one is at GTRI. If the VISORS team is not able to use an existing CoM, resources should be dedicated to designing one from scratch. Finding the CoM is important enough to the function of the mission that it likely warrants investment of precious time and manpower to find.

H. Launch Vehicle Preparation and Integration

The final simplified high-level block is Launch Vehicle Preparation and Integration, shown in Figure 10 below. The spacecraft must be packed up at GT and transported to the launch site with at least two VISORS GT engineers to accompany them upon arrival. At the launch site, the personnel must conduct full vehicle functional tests to ensure that nothing was damaged during transit and to verify the spacecraft are still operating as expected one last time before launch. The final step of launch integration will likely be to Charge Batteries to Launch State before both spacecraft are loaded onto the rocket. At this point, the launch vehicle integrators will likely be the ones to mount the OSC and

DSC onto their payload ejection mechanisms. This simple process flow should be expanded upon further as additional details from the launch providers are learned.

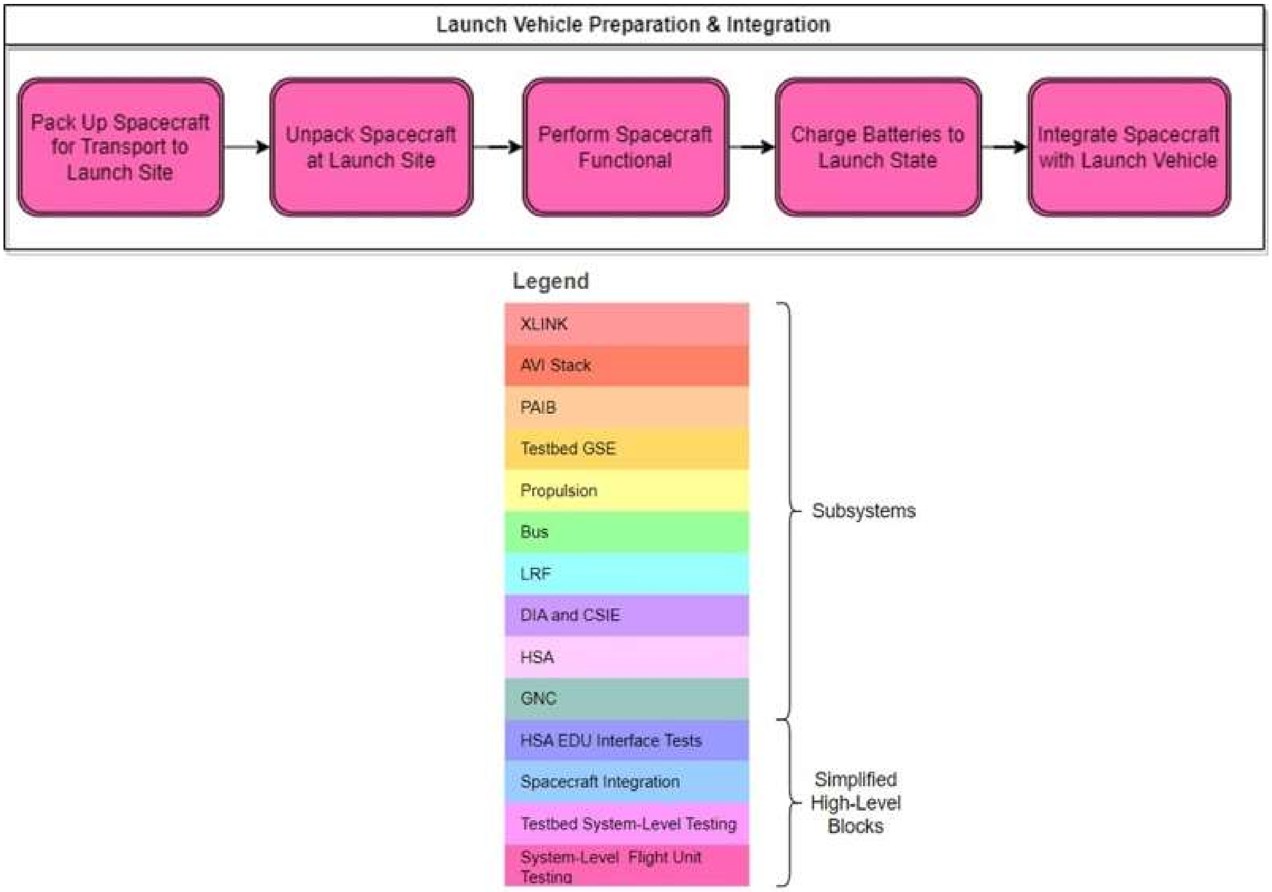

Figure 10. Launch vehicle preparation and integration flow.

IV. Assembly, Integration, and Test Lessons Learned

While it is inevitable that problems will arise during AI&T, it is important that the planning not be one of them. Keeping AI&T plans and schedules realistic and honest is crucial to be able to quickly adjust and minimize damage when a new problem does appear. The author has spent three summer internships, three semesters of graduate school, and a couple years of undergraduate extracurricular experience planning for or performing AI&T and has some advice to offer on this subject. The following are a few additional lessons gathered from these experiences about AI&T:

1. AI&T activities will usually take around two to three times as much time as initially expected, especially if it is the first time performing the procedure.

Whether it is preparing for work in the cleanroom or troubleshooting any interface issues that inevitably arise when attaching or plugging in components together for the first time, AI&T procedures are filled with time sucks that are not captured in the procedure. Troubleshooting these issues takes significant time and effort from the whole team and can be mentally and physically draining. When scheduling AI&T activities, never plan for the time it would take for the procedure to go exactly right. If only a limited time is allotted to perform a procedure that day, ensure that it is one where stopping for the day and starting again later will not compromise the effort if troubleshooting goes over the allotted time.

2. Ensure redundant personnel for every activity and build in time for personnel breaks.

The cleanroom is an overstimulating environment: the loud fan noises, the harsh lights, the difficulty in communication due to masks and loud ventilation system, the smell of isopropyl alcohol, the discomfort of the personal protective equipment, and the constant self-awareness needed for working with and around flight hardware are exhausting. There should be at least two people who can perform every task required during the procedure if possible so formal shifts can be taken. If only one person can perform an activity of considerable length, extra effort should be put into creating a procedure that others who do not have the same depth of knowledge can know what to do and what to look for. In addition, whoever oversees running the procedure at that time should build in time for lunch and personnel breaks so that all people performing integration activities are sharp. Performing AI&T activities while exhausted, ill, or rushed introduces unnecessary and unacceptable risk to the personnel and hardware. This should be communicated to and understood by all personnel before ever running a procedure. While it is up to each person to know and understand their own limits with respect to what is expected during any flight hardware procedure, it is also important for the procedure lead to monitor their personnel and stop any activities they deem are unsafe or done outside of good judgment.

3. Good documentation is important, but so is the schedule.

While performing integration activities, it is important to be well-prepared for the task so as to not unnecessarily introduce risk to both the hardware and the personnel performing the operation. All personnel should be properly trained for the task at hand as a qualification for working on hardware. However, sometimes starting a procedure with imperfect documentation in the moment – especially when working with EDUs or performing integration - is worthwhile for the team to get working with the hardware earlier. The main reason for this is because the AI&T process is like driving quickly on a road filled with potholes; sometimes it is not possible to see and anticipate the issues that come up while plugging in components for the first time until the team is right on top of them (i.e. in the midst of the procedure). This was often found to be the case when working with the satellite subsystem testbed components. If the team must use up schedule margin so they can wait until a full procedure is properly written up with all the finest details documented beforehand, they may find an issue that will heavily delay their AI&T process weeks later than they had to if they had simply started working with the system earlier and documenting as they go. Not only that, but the meticulously written procedure may also need to be rewritten regardless because it did not conform to the reality of working with the system. While this approach of waiting until a full procedure is written to start may be appropriate for a NASA or government project with lots of resources, it becomes more of an ideal rather than a realistic approach for AI&T in a low-budget, rapid development environment. The caveat to all of this is that this approach relies on the personnel understanding the system well enough to safely perform the procedure without a completely formalized, perfectly written test procedure. If the only aspect blocking a test from being performed is a time-intensive finalization of the documentation and not a lack of understanding of the system, then it may be advantageous to simply start working with the components to refine their knowledge and correct the procedure to create a more accurate, realistic procedure. It should be noted that it is not recommended to run a procedure on flight hardware without first testing and troubleshooting the procedure on the EDU, if possible. In addition to the testbed EDUs, this lesson learned was written with the spacecraft integration procedures in mind, as that procedure has taken the team much time to continue refining at the expense of schedule margin that perhaps could have been saved if a lower documentation (but *not* personnel understanding) standard was held to.

V. Conclusion

The VISORS AI&T process is a unique and challenging problem in systems engineering, project management, and planning. An AI&T plan is necessary to guide the team through this difficult process while maintaining an understanding of the major project objectives and goals at this stage in the development. Due to the nature of systems integration, it should be expected that things will go wrong and tests will go awry in a way that is not planned for. A better understanding of the dependencies driving certain parts of the AI&T schedule will allow for a more flexible conceptualization of what must occur to successfully deliver the spacecraft. This paper describes an overall order at which these events will nominally happen and gives specific attention to the major blockers for progressing to each stage of AI&T. A high-level sequence is first presented to provide a big picture understanding of the AI&T flow. Then, each block is broken down into its respective lower-level milestones for each individual subsystem of the VISORS mission. Arrows between blocks are used to show sequential relations between events and coloration is utilized to show overall progress of a specific subsystem. Each major subsystem- and system-level test required to show adequate function, system compliance with mission objectives, and overall system robustness are described in detail. Finally, a few lessons learned are given to provide AI&T planners with the tools to adequately allocate resources and time where they will likely be needed, based on prior experience.

Acknowledgments

The author would like to recognize Dr. Glenn Lightsey for his advisement and guidance throughout the master's program. In addition, the author would like to thank Ebenezer Arunkumar, Michael Hauge, and Conner Awald for their invaluable advice and feedback on the core contents of this paper. Finally, the author would like to recognize Kian Vilhauer and Joseph Gelin; without their support in running VISORS integration, this paper would never have been completed.

References

[1] Lightsey, E. G.; Arunkumar, E.; Kimmel, E.; Kolhof, M.; Paletta, A.; Rawson, W.; Selvamurugan, S.; Sample, J.; Guffanti, T.; Bell, T.; Koenig, A.; D'Amico, S.; Park, H.; Rabin, D.; Daw, A.; Chamberlin, P.; and Kamalabadi, F.; "**Concept of Operations for the VISORS Mission: A Two Satellite Cubesat Formation Flying Telescope**," AAS 22-125, 2022 AAS Guidance, Navigation and Control Conference, Breckenridge, CO, February 2022.

[2] Kimmel, E.; Paletta, A.; Arunkumar E.; Krahn G,; and Lightsey, E. G.; "**Testing Methodology for Spacecraft Precision Formation Flying Missions**," AAS Guidance, Navigation, and Control Conference, Breckenridge, CO, February 2023.

[3] "**Cosmos.**" *Ball Aerospace COSMOS*, Ball Aerospace, 2022, ballaerospace.github.io/cosmos-website/.

[4] "**How Much Did NASA's James Webb Space Telescope Cost?**" USAFacts, USAFacts, 3 Oct. 2023, usafacts.org/articles/how-much-did-nasas-james-webb-space-telescope-cost/#:~:text=The%20Webb%20telescope%20was%20initially,through%20the%202021%20fiscal%20year.

[5] "**What Is BioSentinel**?" *NASA*, NASA, 29 Oct. 2023, www.nasa.gov/centers-and-facilities/ames/what-is-biosentinel/.