CAPITOL
Technology University

CACTUS-1

The High Altitude Handbook is a product by Ian Hastings.

Acknowledgements: HAB payload development was supported by grants from the Maryland Space Grant Consortium (MDSGC).

uploaded July 2019

The HAB Handbook

By Ian Hastings

Contents

Introduction

What is Systems Engineering?

To talk about systems engineering, we must first define what a system is. A system is defined as anything that is composed of other components. For example, a car is a system. It's made up of a chassis, an engine, suspension, etc. Breaking down systems into their components is the heart of systems engineering.

Systems Engineering is a robust method of designing, creating, and operating a system. The method relies on identifying and quantifying the requirements and goals, creating various designs and selecting the best design, and in the end, verifying that the design meets all goals and requirements and goals.

Why use Systems Engineering?

Without the use of systems engineering, large projects suffer where several independent groups made the components had issues integrating with each other and often the system created from the components did not fulfil the initial function and were often unusable. On top of that, missions that don't utilize robust systems engineering principles tend to spend more money, go past their deadlines and don't perform as well.

Example:

For example, if we were building a car, we cannot design and create each component one after the other, we must design two parts at the same time, like the chassis and the engine. If the group who was working on the chassis, changed the dimensions of the chassis for some reason, it may impact how the engine fits into the car. Without systems engineering overhead, this issue might go unseen until final integration where the engine would not fit. This mistake would cause one or both groups to have to go back and redesign and remake their components to make them fit, causing a massive loss of time and money for the project.

However, this situation is easily avoidable, with a systems engineering architecture in place, a systems engineer can monitor the entire process and make sure that each component is meeting its requirements and making sure that each component and interface with other. In this example, a systems engineer would see the change in size of the chassis, and either make the chassis team keep the original design or go to the engine team and give them new interface requirements before the system is complete.

What is HAB?

High Altitude Balloons, frequently abbreviated as HAB, are simply very large latex balloons that are filled with Helium in order to carry a payload to a high altitude. HAB are frequently used by institutions like NOAA and NASA for weather and climate measurements, as the most accurate method of gauging temperature, moisture and pressure of the local atmosphere is to release high altitude balloons to physically measure the data. HAB can easily fly into what is known as "Near Space" Which is a level in the atmosphere where the pressure is so low that humans cannot survive, and the environment is highly similar to being in space. These balloons can fly up to 120,000 feet into the air and frequently reach 100,000 feet. Upon reaching these altitudes, the low pressure causes the balloon to expand so much that it bursts, causing the payload to freefall and parachute safely down to the ground for recovery.

These HAB launches are also used as a low-cost method of getting to near space for science missions. Because it doesn’t cost much, these missions can be launched frequently and with higher risk payloads without fear. These payloads are frequently in the 1-2-pound range and are very small form factor, much like a CubeSat and are controlled by consumer grade microprocessors and sensors, like Raspberry Pi or Arduino. Most launches fly many different payloads at the same time in order to maximize the benefit of 1 launch and each payload is tied to the balloon through the main payload string.

Example:

For example, Cloud360 was a HAB payload designed to take PH measurements of nearby atmospheric moisture as well as take continual 360-degree video in order to corollate the PH to a specific cloud or cloud type. The mission was designed, built, and tested in one month by a group of visiting Brazilian students. The mission flew a total of 4-5 times in total and was fully successful in all but 1 launch. The payload was tied into the main payload string before launch and flew to 100,000 to 120,000 feet above sea level, constantly taking in 360-degree video and using a spinning spool of PH paper to measure the PH of clouds that the payload flew through. In total, the mission cost around 300 dollars to create, and the major cost factor was the $220 360-degree camera that was used. The mission highlights just how easy it is to get good useful science data from a HAB launch for a very low cost.

System Design Process

What is System Level Design?

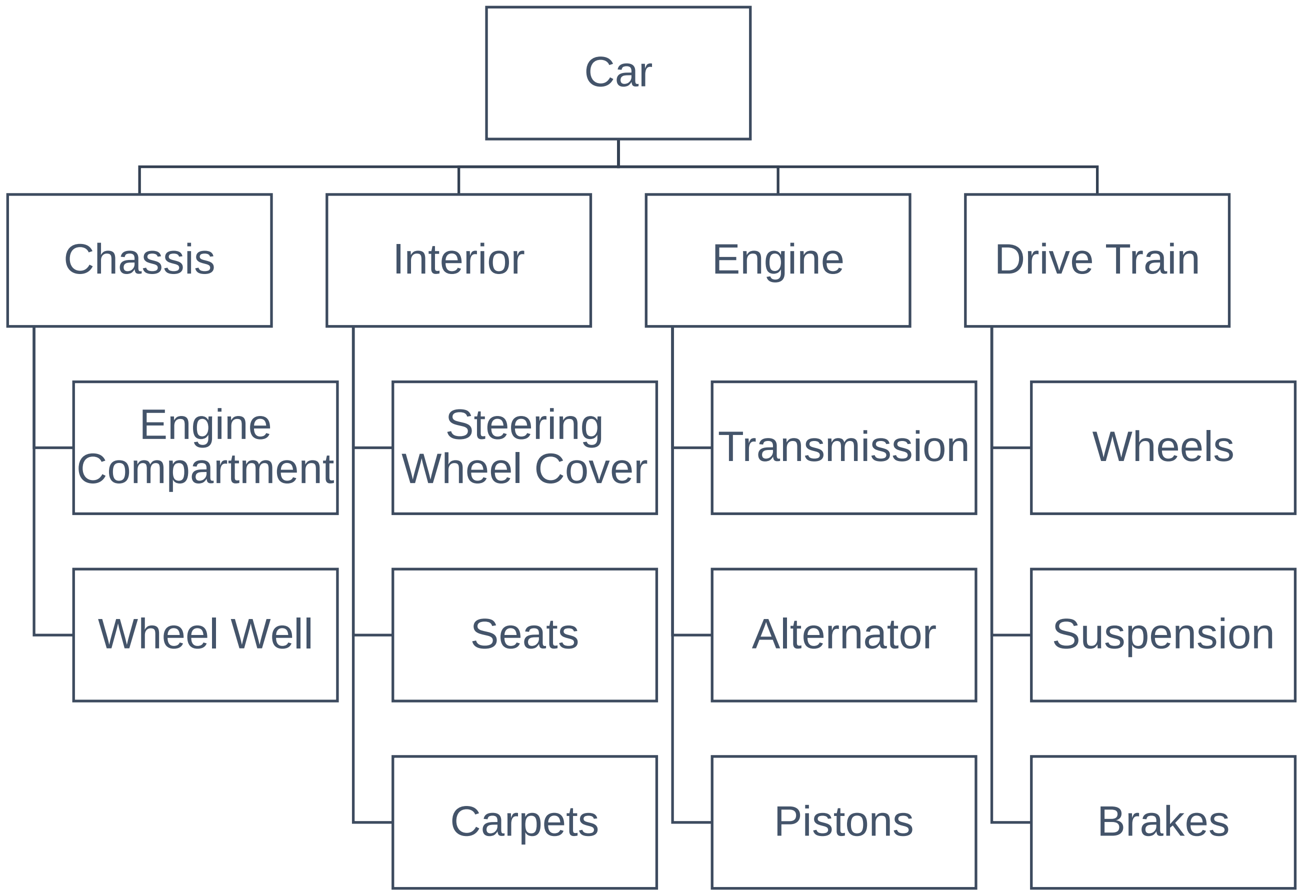
System Level design is the process of breaking down a systems into subsystems. It is extremely useful for complex projects because it allows the project to be separated into subsystems that can be designed and built in parallel to each other, rather than working on a system as a whole. Each subsystem has its own requirements and interfaces that are dictated by the system's requirements and goals.

Example

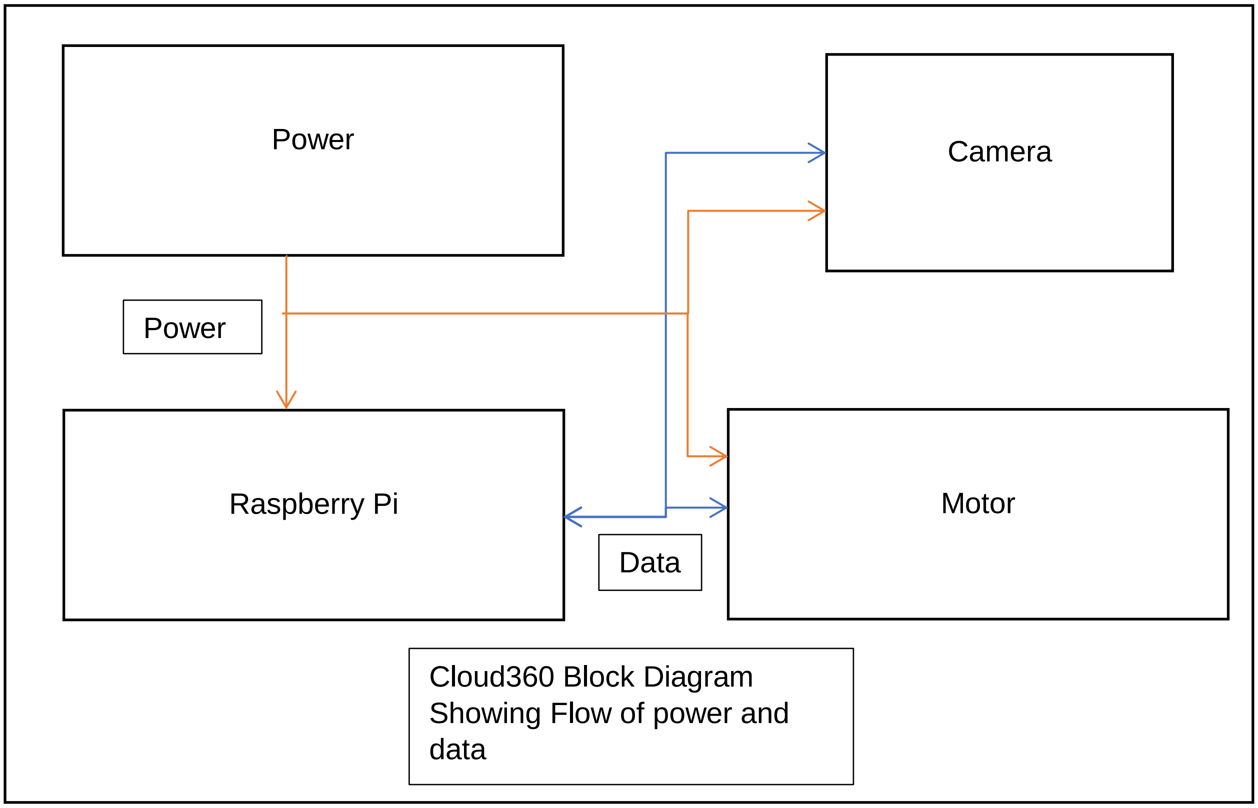
For example, a car isn't designed and built as a whole system. It's much more efficient for them to split up the car into subsystems. You can break the car down into a chassis, an engine, a crew cabin and drive system. Each subsystem will have requirements, like the chassis might have to be a certain size to fulfil the mission requirements. Furthermore, there are interface requirements for subsystems that meet, like the chassis might have to have mounting brackets for the engine to sit in.

System Modeling

Modeling a system is done in two different ways. The first method is the hierarchical diagram of the system from the top down. This gives the team a sense of how the system and subsystems are broken down. At the top of the list is the system itself, the row below are first level subsystems and the following rows.



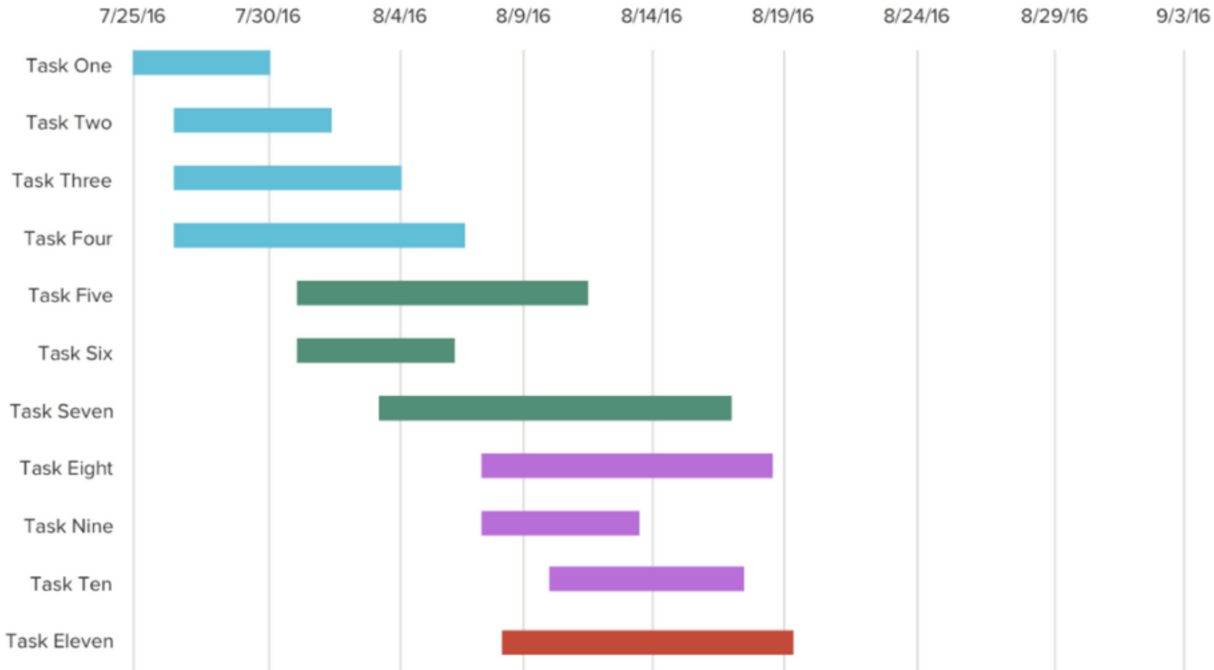
The second method of modeling the system is the interface block diagram. This model shows the individual components and how they interface with each other, with different colored lines representing different types of interfaces, such as mechanical, electrical, or data interfaces. This diagram helps to solidify how each part ties into the whole design.



System Scheduling

Keeping a project on schedule is one of the most important systems engineering tasks. When your flight is a set date it is imperative that each part of the design is designed, built and tested by that date. Most issues with payloads stem from poor scheduling, the “art” of pulling an all-nighter the day before the launch being the cause of many untested payloads flying. Given that, a clear, definitive schedule is a great tool for any team.

The best way to display a complex schedule visually is the Gant Chart. A Gant chart lays out each task on the Y axis of a chart and times on the x axis. This is useful for showing how work on each task overlaps with each other, as one subsystem group is working on their subsystem at the same time as another group for maximum efficiency.



i: Example Gant Chart. The Different Colors Represent the different groups working on tasks. Tasks that overlap with each other show tasks that are done at the same time.

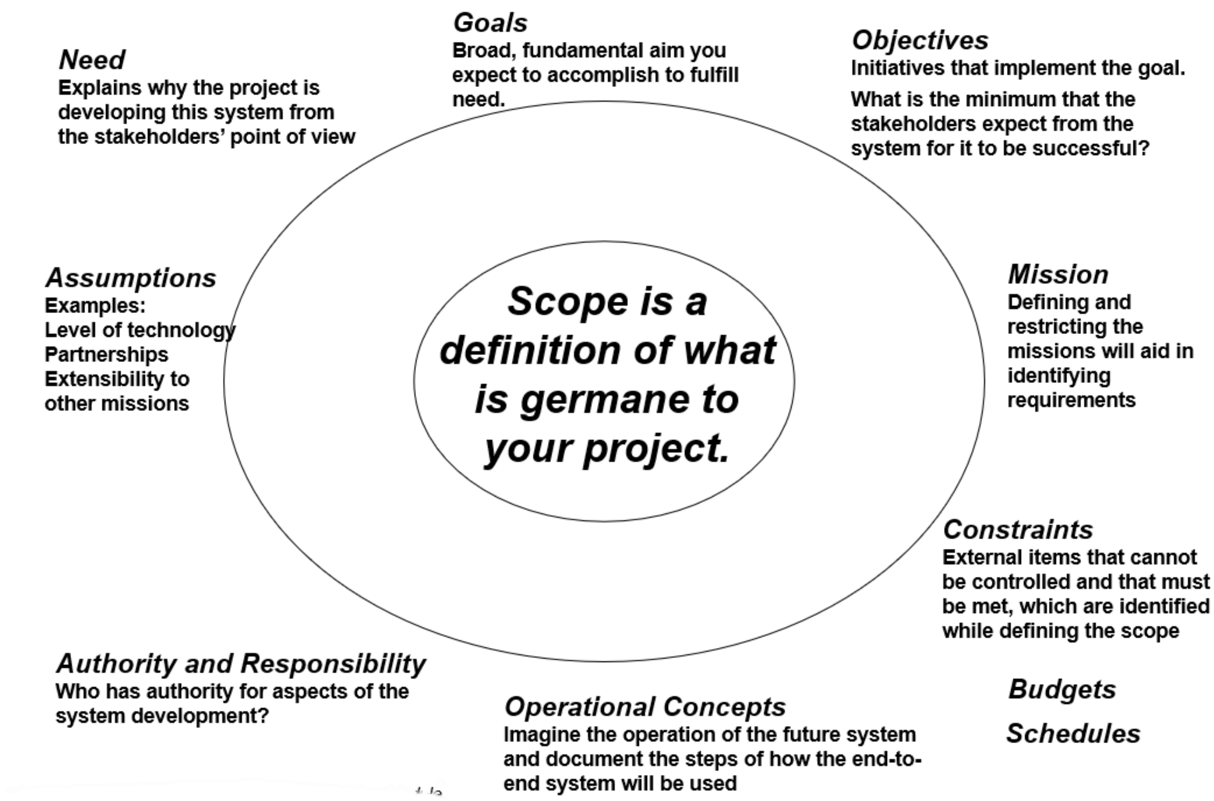
System Interfacing

On top of the previous interface block diagram, an interface control document should be made for every system. It can be as simple or complex as needed for the project, but each subsystem needs to know how it will interface with into the main payload to avoid confusion.

This entails listing the electrical connectors or wires needed to connect components, fasteners used to tie a component to the body of the payload, where holes need to be placed for bolts or wires to come through, and even how the software interfaces with the electronics (i.e. pinouts for the flight controller). Setting this stuff early on will avoid confusion and hasty changes down the road.

Product Realization

This section will outline the high-level concepts of Systems Engineering. Product Realization is the act of defining the scope, or defining what is germane to the project, of the project so that throughout the design process, you can check that the payload is still meeting what it needs to do.



Need

The need is the highest-level concept and lays out what purpose the payload is serving. While it seems trivial, it is important that the payload serves a purpose, however small it may be. If the payload doesn't serve an identifiable purpose, then why should money and time be spent on it?

A Communications payload may be serving the need: "There is a lack of simple, affordable communications devices for HAB projects"

Goals

Goals are the broad aims of the project and is what you wish to accomplish to fulfil the need. It's important that your goal is broad and doesn't talk about the specifics of the project, just what you wish to achieve with it.

For the communications project, the goal could be "Find a cost-effective communications solution for HAB payloads:

Objectives

Objectives flesh out how you intend to achieve the goal of the project and define the success criteria for the project.

Some example objectives for the communications payload are:

- Conduct a trade study to examine different communications solutions
- Build payload with the best solution

- Fly payload on HAB flight
- Send and receive telemetry and commands
- Spend less than 200 dollars on the payload

Mission

The mission is the definition of what the project will be doing. Defining the mission will help flesh out the requirements later.

For example: Fly a basic communications payload capable of uplink and downlink of data 75% of the time the payload is above 2000 feet on a 3-hour high altitude balloon flight.

Constraints

External factors that affect how the payload is designed and made.

For example:

- Be made of non-hazardous materials
- Follow the FCC regulations for RF communications
- Fit within the allotted size of the payload
- Weigh under 1 pound
- Be capable of being made within 2 semesters

Assumptions

These are the assumptions needed for the project to successfully complete its goal.

For example:

- The current communications solution isn't the best solution
- There are communications solutions under 200 dollars
- It is possible for a payload to communicate 75% of the time
- Etc.

Concept of Operations

The concept of Operations is a step by step list of how the payload will work and operate with the end user.

A simplified example:

1. Tie into a HAB payload string
2. Turn on and automatically start the program
3. Send a test communication back and forth from a laptop on the ground
4. Launch on the HAB flight
5. Downlink Telemetry every 5 seconds after receiving a command from the ground
6. Store downlinked telemetry for later examination
7. Communicate until balloon bursts
8. Cease downlink of data

9. Ping the payload from the ground every 15 seconds to see if a connection can be made
10. Continue for duration of flight and store results of ping.
11. Retrieve and disable payload at landing site
12. Examine data to see if the mission was successful

Requirements

Requirements are the list of things that the payload **must** do in order to meet its objectives. When reviewing the project, this is the list that you go through to see if your payload is doing everything it needs to do to meet its objectives. It is useful to phrase all requirements with the word **must**.

A simple requirements list consists of everything that is important for the payload to do all lined up, but a more advanced and useful list consists of different levels of requirements, where a level 0 requirement is more important than a level 1 requirement and so on. Listing your requirements in this way gives a very clear way of seeing what is critical for the mission and what Is less important. If a time comes where you needed to de-scope the project, it is much easier to look at lower level requirements as a way to simplify a project

A simplified requirements list:

- Level 0
  - Must cost under 200 dollars
  - Must weight under 2 pounds total
  - Must meet all launch requirements
  - Must communicate at Near Space
- Level 1
  - Must function for the complete duration of the mission
  - Must be smaller than 1’x1’x1’
- Level 2
  - Must have a data rate of 1 kB/s

With this kind of list, if we’re having issues with the project, we can look to the Level 2 requirements first, and see if we can remove them or modify them to make them simpler in order to make the mission easier.

Risk

When scoping a mission, it is important to quantify and classify risk in the mission so that you can attempt to mitigate dangerous factors. To do this, we create a list of all of the risks to the project and assign them 2 numbers from 1-5. The first number is the impact of the risk, where 1 is a risk with minimal impact, and 5 is a risk with a a potentially mission critical impact. The second number is the probability of the risk factor occurring, where 1 is a factor that is highly unlikely to occur, and 5 is a factor that is very likely to occur.

To visualize these risk factors, we create a risk matrix, where one axis shows the impact of the risk factor and the other axis shows the probability of the risk factor.:

	Consequence				
Likelihood	Insignificant	Minor	Moderate	Major	Critical
Rare	LOW Accept the risk Routine management	LOW Accept the risk Routine management	LOW Accept the risk Routine management	MEDIUM Specific responsibility and treatment	HIGH Quarterly senior management review
Unlikely	LOW Accept the risk Routine management	LOW Accept the risk Routine management	MEDIUM Specific responsibility and treatment	MEDIUM Specific responsibility and treatment	HIGH Quarterly senior management review
Possible	LOW Accept the risk Routine management	MEDIUM Specific responsibility and treatment	MEDIUM Specific responsibility and treatment	HIGH Quartely senior management review	HIGH Quarterly senior management review
Likely	MEDIUM Specific responsibility and treatment	MEDIUM Specific responsibility and treatment	HIGH Quarterly senior management review	HIGH Quarterly senior management review	EXTREME Monthly senior management review
Almost certain	MEDIUM Specific responsibility and treatment	MEDIUM Specific responsibility and treatment	HIGH Quarterly senior management review	EXTREME Monthly senior management review	EXTREME Monthly senior management review

The squares in red show regions that have high impact and high probability, these are to be highly avoided. A mission with a red risk factor is highly unreliable at best. The yellow zone shows medium level risks. These risk factors aren't great but are acceptable if there aren't too many of them. The green squares show the low-level risk factors and aren't that big of a deal.

You can see that even if a risk factor has a high impact to the project, if it's highly unlikely to occur then it can be acceptable, and a frequent risk factor can be acceptable if the impact to the system is low. Given this, if there is a red level risk factor, you can approach it two ways. You can attempt to lower the impact of the risk factor if it occurs, or you can lower the chances of the risk factor occurring in the first place.

Fundamentals of Payload Design

Basic Payload Ideas

Coming up with the idea of a payload can be difficult, because the mission must serve a good scientific purpose in order to justify flying it on the balloon. A mission must be unique and interesting for institutions to want to fund the payload. Missions don't need to be purely scientific, a payload could be an engineering payload that tests a unique engineering idea that hasn't been done before. A good idea for coming up with an idea is to either consult faculty that may have research that they want to have done, get a group of fellow students together to brainstorm about unique ideas, or even adapt a payload that has flown previously to either improve upon it or add additional functionality

Some previous examples are:

- Cloud 360: Utilized revolving PH paper and a 360-degree camera to take PH measurements throughout the flight to see if the PH of clouds were different at different altitudes or changed over time
- Project Hermes: Utilized the iridium network to create a simple one-way communications device that would send short telemetry back to the ground through emails and could be used as a low accuracy tracking device
- TrapSat: Utilized silica aerogel to capture micrometeorites at high altitudes
- HABScope: Utilized an IR camera to attempt to do meaningful IR astronomy from high altitudes because IR astronomy is impossible on the ground
- Magneto: Utilized a 3-axis magnetometer to map the magnetic field fluctuations throughout the flight to try to map the local magnetic field in Maryland.

The ideas can be as simple or complex as you want. Project TrapSat was highly complicated due to the complex procedures needed when dealing with silica aerogel and needed specialized equipment to test for micrometeorites, conversely, a project can be very simple, such as Magneto, which simply utilized a 3-axis magnetometer and only needed a modeling software to correlate the science data. Once you have your idea though, you'll need to start figuring out the design of the payload and what you're going to use to make it.

ii: The Picture on the left shows HABScope, the middle shows Cloud360 on a payload string, and the right shows the inner workings of Cloud360

Materials

Materials that you use to build a payload are highly dependent on the requirements of the payload. Some key requirements for the structure of a payload are:

- Thermals: How well the payload retains Heat
- Durability: How well the payload stands up to damage
- Cost: How much the structure costs
- Material Properties: How well a material stands up to low temperatures and pressure
- Simplicity: How easy it is to design
- Ease of Integration: How easy it is to put the parts into
-

Thermals

High altitude balloons payloads reach incredibly low temperatures, below -30 degree Celsius, and many electronics and materials have strict operating temperatures. Many electronics shut down when they get too cold and many materials get very brittle at low temperatures. So, it's important to design with this in mind. Many payloads with cover themselves in insulation foam that you can get at any local hardware store for very cheap. This is the simplest way to keep your payload from freezing. However, heating elements can be added if passive insulation isn't possible or is insufficient for the project.

Testing of thermals can be done using dry ice inside of a plastic cooler to simulate the low temperature environment of near space. Alternatively, a group can attempt to get access to thermal chambers or thermal vacuum chambers for the same purpose.

Durability

HAB payloads fall from upwards of 100,000 feet in the air with nothing but a small parachute to slow them down. While the parachute slows the payload down significantly, it can still hit the ground with a fair amount of force and can break the exterior structure of poorly designed payloads. Payloads should be designed to be dropped with minimal damage. This usually includes using sturdy fasteners and making sure all cables are tied down to prevent wires coming loose.

Testing Durability can be done by dropping the payload from slowly increasing heights and doing a full function check after each drop. If something breaks or even comes loose, the team should diagnose why it broke and brainstorm how to fix it

Cost

Most payloads have a limited budget, so saving money is important where possible. Efforts should be made to use cheap and available materials, but caution should be taken to avoid spending so little money that you sacrifice the functionality of the payload.

Material Properties

Some materials react differently in near space environments than others. Many materials can become brittle, at low temperatures, as discussed before, but others can react strangely to low pressure environments. Many electronic components can have capacitors explode, sealed containers can explode, and some materials outgas at low temperatures. Outgassing is when gasses trapped within a material escape from the material due to lower pressure. It's important to double check all materials being used to check if they can function at pressures lower than .8 psi. For example, ABS plastic, a common plastic used in 3D printing and premade boxes, can outgas at lower pressures but doesn't lose much of its structural integrity in the process, but it can leave a residue on lenses which can obscure camera footage.

To test if your payload can properly operate within a near space environment, testing within a vacuum chamber, preferably a thermal vacuum chamber which lowers both the pressure and temperature in the chamber, is required. There are many vacuum chambers available to use at CTU that will achieve sufficiently low enough pressures to simulate a near space pressure and to use one contact the Fusion Lab Staff. You may also seek out other institutions with more advanced equipment and see if they will allow, they will allow you to test your payload in their machine.

Simplicity

Simplicity in design is highly important in HAB payload construction. Simplicity can keep costs down and help keep a team on schedule. A payload should meet the requirements of the mission and do nothing more. Designing for every scenario isn't smart design, it’s stupid design. It is far easier and more cost effective to design a simple payload and then add functionality onto it, than it is to design an overly complex payload and must scale it back for cost and time reasons.

Some simple methods to keep your design simple:

- Use passive (i.e. non-powered, non-moving) devices if possible
- Use COTS (Consumer Off the Shelf) or readily available products
- Don’t reinvent the wheel, use other people’s work and knowledge to your benefit
- If you aren’t sure if you need something, test to see if you need it before adding it
    - For example, if you suspect you might have a thermal problem, run a thermal test to see if you really need powered heaters before adding them
- If you think something is too complicated, it probably is

Ease of Integration

It is often easy to forget that you must physically build the payload when you are designing it. So, it is wise to keep the construction of a project in mind when you are working out the design. A good example of this is designing custom mounts and mounting holes into 3D models or figuring out how you’re going to attach a part to the main structure. One pitfall that many people fall into is that they don’t design tolerances for the size of cable connectors or the thickness of wires. When these sorts of things are overlooked, it can cause a lot of confusion and last-minute changes when it comes to integrating all the parts together. To avoid this, keep an integration document laying out how all the parts integrate into each other, more on this later.

Main Structure

The main structure for a payload falls into two main categories, premade structures and fabricated structures.

- Premade:
    - Advantages:
        - Cheap
        - Easy to find
        - Can be extremely durable
        - Can be very good at keeping heat in the payload
    - Disadvantages:
        - Can be difficult to build in
        - Isn’t always good for high altitudes
- Fabricated:
    - Advantages:
        - Extremely easy to integrate parts into
        - Can be easily iterated upon
        - Can be very durable in most directions
    - Disadvantages:
        - Takes a long time to design

- Takes a long time to make
- If errors occur, you must remake the part
- Can be significantly weaker

iii: The picture on the left shows the project Magneto right before launch. Magneto used a premade, hard plastic container. The picture on the right however shows the complex 3D printed frame of Cloud 360. You can see the various built in mounting brackets designed into the structure.

An example of a project that used a premade structure is project Magneto. It was a simple, cheap mission who had a relatively simple goal. Because of this the Magneto team utilized a premade, durable Sterlite container lined with insulation foam. The project's complexity didn't call for the need to create a custom structure and the use of the premade structure gave more time for testing.

One project that utilized a fabricated 3D printed structure was Cloud 360. The project had many different individual parts that all had to integrate smoothly. It had a long period allotted to the development and design of the project and a large budget, so it made sense to use the 3D printed structure. The specially designed structure made it so that the individual components all screwed in and connected with ease and saved time in the long run.

Electronics

Power

Power for a payload is almost always done through pre-charged batteries. The two major requirements for a battery are Voltage and Capacity. Voltage is the output Voltage of the battery and is determined based on the voltage requirements for the other electronics in the system. Capacity is how long the battery will last when power is being drawn from it. Capacity is usually measured in Amp-hours ore milliamp-hours. A 2 Amp-hour battery will last for 2 hours with a current load of 1 amp and a 500 milliamp-hour battery will last for 2 hours with a 250-milliamp load. The below equation can be used to determine how long a battery will last, where Capacity is the given capacity of the battery and Current is the estimated average current of the payload.

$$Duration\ of\ Battery = \frac{Capacity}{Current}$$

This is only an estimate of the duration that It will last though, and as with everything, it is imperative that you test that the payload can last for the full duration of the mission or about 3 hours.

There are several types to use but lithium ion batteries are preferred, especially because alkaline batteries can be damaged or fail to work at near space. For the batteries, lithium ion AAA or AA cells can be used in series to increase the voltage and in parallel to increase capacity (Seek further documentation on series and parallel circuits for more information on this subject). However, lithium ion battery packs are available cheaply on the market and give a consistent voltage output and are high capacity. These packs are rechargeable, which is useful for any mission that intends to fly more than once. These packs range from consumer phone recharging banks to more complex designs. Regardless, the batteries used for the mission should output the correct voltage, have the right capacity for the mission.

Controller

The flight controller is the brain of the payload. Its main job is to control the rest of the payload as the central hub for electronic components. It is responsible for:

- Taking in sensor data
- Sending out commands or controlling electronics
- Logging sensor data onto storage
- Detecting and avoiding faults

There are two main flight controllers used for high altitude balloon missions, the Raspberry Pi and the Arduino. Both controllers can do most of the other things that the other can do and both are flight proven hardware. The major differences in the two systems are:

- Raspberry Pi
    - Is a full computer with a Linux operating system
    - Typically programmed in Python
    - Has a longer start-up time due to having to load the full OS on launch
    - Has specific connectors for Raspberry Pi cameras and displays.
- Arduino
    - Arduino only runs the code that is loaded on it

- Programmed in a C based Arduino language
- Can be more power efficient
- Low start up time
- Has a multitude of “shields” that add advanced functionality

iv: Raspberry Pi

The computer really doesn’t matter to the payload if it can fulfil the necessary requirements of the payload. A preference to a programming language or finding a piece of hardware compatible with one of the controllers is a valid reason to choose one controller over the other.

Sensors

Sensors are the only way for us on the ground to know what is going on with the payload. Sensors take in information from the environment and send them to the flight controller for processing.

External Sensors

External sensors are sensors that are observing things outside of the payload. Common ones are magnetometers, ambient temperature sensors, pressure sensors, cameras. Often times these are useful for gauging how well your payload is doing compared to the external environment or for use as the scientific payload.

For example. Cloud 360 used a 360-degree camera paired with a spooling roll of PH paper for its experiment. These are both external sensors that were used for the scientific objective of the mission. However, the payload also had an array of pressure sensors, temperature sensors, and magnetometers to be able to debug the payload if something went wrong.

In this example, Cloud 360 had an issue where if the ambient temperature reached -20 degrees Celsius, the camera would shut down due to being too cold. The external temperature sensors helped the team diagnose the issue and fix it.

Internal Sensors

Internal Sensors are sensors that monitor the status within the payload. Some common internal sensors are voltmeters and internal temperature sensors. These are useful for monitoring the conditions inside the payload to make sure that you're within the requirements of your payload.

On the payload magneto, the team utilized a digital voltmeter to monitor the output of the battery to make sure it was giving a consistent output for the mission.

Software

Design Principles

When designing software for a balloon payload, regardless of the platform, it must be simple and robust. A complex code tends to have more failure points. Designing your code to restart or resolve errors on its own is highly important, because if an error occurs, you cannot be there fix it or restart during the flight.

For example, in Python this can be accomplished by putting code into "Try" and "Except" blocks, so that if a part of the code fails, which is all too easy to accomplish in Python, it will continue running.

Testing

v: Example Vacuum Chamber

Everything in your payload must be completely tested. In the previous sections we discussed individual component testing. But it’s just as important to test the system to make sure that it meets the mission requirements. A full “day in the life” test is needed to prove that the payload is completely functional and will work on a flight. It entails running the payload through as close to a near space environment as possible for the duration of a flight with no human interaction.

Ideally, this test would be done in a thermal vacuum chamber tuned to the perfect parameters and the payload would be dangling from a shifting payload string. This is nearly impossible to achieve. It can be approximated though, running the payload through a thermal vacuum chamber is the next best thing followed by a simple thermal chamber. It’s important to test that the payload can and will function when it goes to fly.

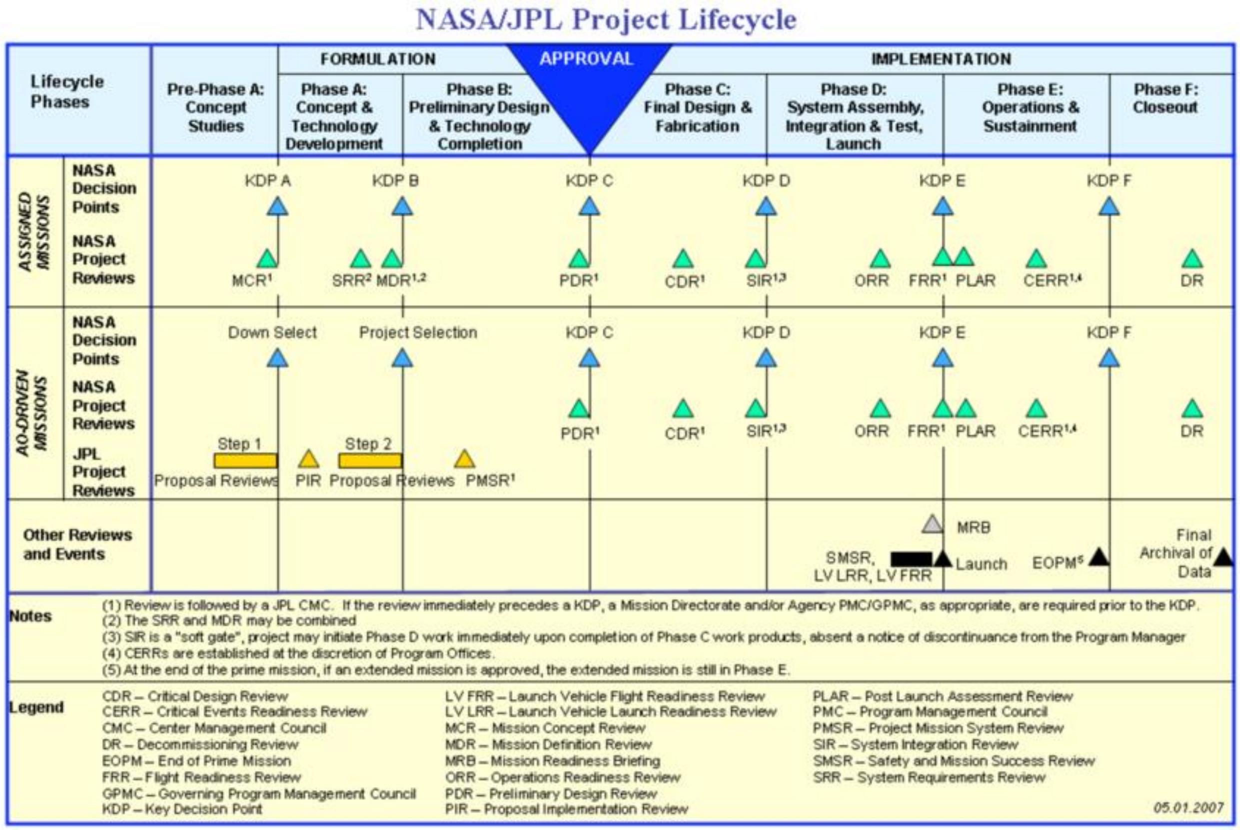
Project Lifecycle

The Project Lifecycle is a tool used by systems engineers to create a step by step design calendar for a project. A typical project is split into around 6 sections, Phase A, which is defining the concept of the project, Phase B, which is designing the payload itself, Phase C, which is the final design and fabrication phase, phase D which is the integration and testing phase. Phase E which is the launch and operations of the project, and phase F which is the decommissioning of the project.

Each phase has certain action items to achieve during the phase, and before being able to move onto the next phase, you much pass a review to show you’re ready to continue. At the university level, it is very useful to have these reviews to show that your payload is progressing well, it can be done with helpful faculty, the balloon payload program intern, and/or experienced students.

At the review, you present all the work that has been done for the phase that you’ve worked on and allow the reviewers to come to decide if you’re ready to proceed. If you haven’t proceeded enough to the liking of the reviewers, you should go back and follow the recommendations of the reviewer.

For various reasons, each project may wish to spend more or less time in a particular phase. Some payloads may need more time in the design portion of the lifecycle and less in the construction or vice versa. However, the length of each phase should be determined early on based on the requirements of the mission.



vi: Full NASA Project Timeline

Purpose of the Project Lifecycle

The Project Lifecycle helps a systems engineer schedule a project and balance the amount of design to the amount of fabrication and testing of the payload. Some payloads focus too much on designing the payload and don't give enough time to make the payload before the deadline, while other payloads jump straight to testing and construction of a payload with little understanding of the goals of the payload. Using a project lifecycle, a systems engineer can choose how much time is dedicated to each part of the design process.

The lifecycle also has a built-in way of dealing with issues early on. The review process can catch design mistakes early. For example, if a project has an issue with the scope,

Phase A

The purpose of phase A is to define the high-level goals and requirements of the mission. This can help clarify the scope of the project and guide the following phases. In this phase you should create most of the high-level systems engineering design from the system design process of this paper.

The review following phase A is the SRR, or System Requirements Review. At this review, the reviewers will be shown the current high-level design of the project, such as the goals, requirements, risk, etc. The reviewers will then discuss with the team whether they're ready or not. If the reviewers don't think that the scope of the mission is fleshed out enough, the team should go back to phase A and follow the recommendation of the reviewer.

Phase B

The purpose of phase B is to refine the physical design of the payload. This is the time to work out theoretical tests, simulations, CAD models, and mock up designs of the payload. This work shows that the mission is technically possible and lays out the design before fabrication begins. During this stage you must lay out the potential cost of the final payload, the materials the final payload will have, how it will be put together, where parts will be sourced, how it will be controlled, a payload testing procedure document, and a flow diagram of any software for the project.

The review for Phase B is called the PDR or Preliminary Design Review. It is one of the most important reviews there are for a project, as it is usually the last review before money is spent on the project. In this review, you must show that the project is capable of being built on budget, on time, and can be reliable. You must show that the design of the spacecraft is fully fleshed out so that you do not waste funding.

Phase C and D

The purpose of phase C is to begin the fabrication of all the needed parts and Phase D is for integrating and testing all of the parts into one fully tested payload. For the purposes of balloon projects phase C and D flow together into one phase. In this phase you essentially follow and build the payload that was designed and go through all the test procedures made in phase B. At the end of phase D, you should have a fully functioning, tested payload that meets all of the requirements defined at the beginning of the project.

At the next review, the FRR, or Flight Readiness Review, you are demonstrating that the payload is fully functioning, fully tested, and meets all the goals and requirements of the mission. The reviewers will assess if the payload is ready for a flight and give you the go/no-go for flight.

Phase E

Phase E is the launch and operations of the spacecraft. For most payloads, this involves tying into a payload string and launching on a balloon into the upper atmosphere. Some payloads may have things to do during the launch, such as a communications payload that the ground team is monitoring and downlinking data from.

The review after this is called the AAR, or After-Action Review. It is essentially a meeting within the group to assess how the launch went. You talk about how if the flight was successful or not, you discuss whether or not the mission met it's success criteria and talk about things that could have gone better or that went better than expected. If the mission didn't meet its mission, then you discuss whether you want to go back to phase D, refine the design and try again or to decommission the payload.

Phase F

Phase F is the decommissioning of the payload. During this phase you tabulate any science data that you have and write any papers related to the payload. It is a great time for documenting the payload for future use and sharing the information gained from the project. A white paper on lessons learned from the project is useful for the group if they intend to work on another project in the future or for other groups in the future that can learn from the mistakes of the project.

Conclusion

Using these tools and strategies, you should have a good idea of how to make a balloon payload. Anybody can do it and all it takes is inspiration, practice, and dedication to make the next big payload.