Opportunities for Team Development Based on Lessons Learned From Spaceflight Operations

John L. Goodman
Odyssey Space Research, LLC
1120 NASA Parkway, Suite 505
Houston, TX 77058
jgoodman@odysseysr.com

Abstract — **Lessons learned over a career are useful for identifying team development opportunities to ensure mission success and safety of flight. All human and robotic spaceflight is accomplished by teams of people working with technology. Spaceflight is about leading and organizing teams of people to solve engineering problems. Successful engineers identify lessons learned and best practices over the course of a career. These shape and guide how engineers make decisions, perform work, and interact with people. This paper details lessons learned from over thirty-six years of involvement in human space flight at the NASA Johnson Space Center, both in flight operations and spacecraft development.**

TABLE OF CONTENTS

1. INTRODUCTION

Engineers and managers acquire a set of observations and lessons learned over the course of a career. These observations and lessons shape and guide how they approach their work and make decisions. Preserving and sharing these lessons is necessary for team development, flight safety and mission success.

All human and robotic spaceflight is accomplished by teams of people working with technology, so spaceflight is about leading and organizing teams of people to solve engineering problems. Therefore, development of inter-personal, communication, and leadership skills is as important as development of engineering and problem-solving skills.

This paper documents a set of lessons learned acquired by a guidance, navigation, and control (GNC) engineer over a 36-year career as a contractor at the NASA Johnson Space Center (JSC), supporting both human space flight operations and spacecraft development.

978-1-6654-9032-0/23/$31.00 ©2023 IEEE 2023 IEEE Aerospace Conference, Big Sky, MT, March 4-11, 2023.

From 1986 to 2014 the author supported the NASA/JSC Mission Operations Directorate (MOD) conducting plan, train, and fly operations for the Space Shuttle and International Space Station (ISS). This included serving as a Flight Dynamics representative for real-time support in the Spacecraft Analysis (SPAN) room from STS-114 to STS-135. He also supported development of the NASA X-38, Orion, and two Commercial Crew vehicles from the space flight operations perspective [1-6].

Since 2015, the author supported the NASA/JSC Engineering Directorate in the development of the Orion spacecraft and supported two commercial space companies [7-9]. In addition, the author performed real-time support of the Artemis I mission in the Mission Evaluation Room (MER).

Over the course of his career the author has conducted 16 lessons learned and knowledge capture efforts, which resulted in 13 papers or documents that are publicly available [10-22]. However, these papers do not represent the entirety of the author's observations and lessons learned on spaceflight. This paper is meant to be a more comprehensive account.

2. STAY HUMBLE AND AVOID OVERCONFIDENCE

It is human nature for people to take pride in the performance of a seemingly well-designed and well performing vehicle. However, smart and experienced people often/sometimes/can design and fly vehicles that later experience incidents resulting in loss of mission, loss of hardware, and sometimes loss of life. Overconfidence and pride in a seemingly well performing system is at the root of normalization of deviance [23-25].

The importance of humility is one theme of former astronaut Eileen Collins' book *Through the Glass Ceiling to the Stars* [26]. Humble people will admit that they do not know something, and will work hard to learn, while pride can keep a person from learning. Humility is required to know when to listen and not speak, and to ask questions to get people thinking as opposed to giving direction. Developing and practicing good listening skills and keeping an open mind requires humility. Collins defines four steps to handling mistakes, 1) admit the mistake, 2) correct it, 3) make changes to avoid making the mistake again, and 4) move on. Those that are humble enough to admit their mistakes enable other people to learn from those mistakes.

Pride and overconfidence lead to false assumptions that prevent us from recognizing, investigating, and resolving problems. In his book *Shuttle, Houston*, former Mission Control Flight Director Paul Dye describes how the success trap before the loss of *Challenger* made people think nothing could go wrong, since nothing had gone wrong yet [27].

Weak signals are usually present before an incident occurs. To avoid incidents, continual learning about how a vehicle is designed, operated, and performs is required, as well as continually working to make sure problems are resolved, and weak signals of performance problems are investigated (see Appendix A of this paper). But pride makes these weak signals difficult to perceive.

After the fire that claimed the lives of Gus Grissom, Ed White, and Roger Chaffee on Friday, January 27, 1967, Mission Control Flight Director Gene Kranz held a meeting of flight controllers. Kranz's entire speech is in his book *Failure Is Not An Option* but an excerpt reads as follows.

> "I don't know what Thompson's committee will find as the cause, but I know what I find. We are the cause! We were not ready! We did not do our job. We were rolling the dice, hoping that things would come together by launch day, when in our hearts we knew it would take a miracle. We were pushing the schedule and betting that the Cape would slip before we did." [28]

In a January 27, 2004 email to the Space Shuttle Program, then Space Shuttle Deputy Program Manager Wayne Hale wrote about the loss of *Columbia*.

> "Last year we dropped the torch through our complacency, our arrogance, self-assurance, sheer stupidity, and through continuing attempts to please everyone. Seven of our friends and colleagues paid the ultimate price for our failure." [29]

Wayne Hale later expounded on the danger of overconfidence.

> "The best advice I ever got—Tommy Holloway told us over and over—is, 'You're never as smart as you think you are.' If you ever get to the point where you think you've got it under control, you really don't, and you need to be always hungry and looking out for the indications that things aren't going well. It's a difficult thing in a big organization to keep that edge, and it's particularly difficult when things are going well." [30]

In a 2007 opinion piece for MSNBC, Mission Control veteran James Oberg spoke on overconfidence.

> "They need the consequent inescapable ache of fear and the gnawing of doubt that keeps asking, over and over, if they've covered all angles and done all they can. And if their stomachs do not knot up, and mouths go dry, as they confront such decisions — perhaps they need new jobs." [31]

Organizational causes of accidents are due to people; it takes humility to recognize weak signals of a problem, or a chain of events about to result in an accident, then take action to bring attention to the problem, investigate it, and resolve it.

3. STUDYING AND DISCUSSING FORMAL ACCIDENT REPORTS PREPARES PERSONNEL TO RECOGNIZE PROBLEMS

Organizational culture consists of the values, norms, beliefs, attitudes, assumptions, and practices that govern how an organization functions. A strong organizational safety culture encourages intellectual curiosity, skepticism, seeking to understanding systems performance, learning from past failures, and exhaustive debate concerning risk; while avoiding reactive and complacent behavior, as well as unjustified optimism [25]. Establishing a healthy flight safety culture requires training personnel to intuitively recognize symptoms, both organizational and technical, that could lead to the loss of a spacecraft or a failure to meet mission objectives.

Starting in October of 1998, a Flight Safety Awareness Seminar has been held as part of the yearly NASA/JSC Safety Day activities [15]. The seminar's purpose is to increase flight safety awareness through discussion of accidents, organizational causes, and lessons learned. The seminar enables participants to draw comparisons and conclusions between the causes of the accident and their work environments. This allows personnel to recognize potential organizational or technical accident causes such as communication breakdowns, overconfidence, complacency, normalization of deviance, practical drift, poor decision making, and weak signals of performance problems [32].

The organizational and cultural causes of accidents offer insight into communication breakdowns, overconfidence, complacency, normalization of deviance, practical drift, and poor decision-making. Technical causes of accidents provide visibility into systems design, integration, operations concepts, hardware and software reuse, testing methodologies, and choice of technologies.

Discussions among seminar participants of the organizational causes of accidents in various industries (space, commercial aviation, oil and gas, maritime shipping, rail transport, nuclear power, medical, the military, etc.) helps engineers determine how to apply the lessons to their work. An important guideline for discussion is to avoid blaming "stupid people," incidents occur in industries employing smart and experienced personnel.

A workforce that can recognize both organizational and technical causes of accidents through flight safety education will raise the probability of program success. In addition, the seminar empowers personnel to not only recognize but also communicate and act when confronted with a potential cause of an accident. Since 1998, seminar participants have consistently stated that the Flight Safety Seminar is the most informative activity of Safety Day. Programs that similarly review formal accident reports will empower their personnel to recognize and report possible failures before those failures result in mission failures or claim lives.

4. MANAGEMENT MUST PLAN FOR LESSONS LEARNED AND BEST PRACTICES

Lessons are most often learned when challenges or failures occur in a vehicle development or flight program. The best time to identify and preserve lessons learned is when the challenge or failure occurs, not years or decades later [17]. Memories are fresh right after an event occurs, but as time passes people forget and stories change.

In some cases, engineers may be motivated to document and share their own lessons and best practices. One example is the NASA report *Navigation Filter Best Practices* [33].

However, it often is difficult for a lone engineer to persuade a project to perform a retrospective that documents lessons learned. Many engineers who are knowledgeable in lessons learned and experiences consider themselves to be too busy to participate in such an effort and may lack the communication skills to perform such an effort on their own. Lessons learned and best practices must be documented clearly so that they can be understood. Otherwise, those studying the reports may wonder "What do I do with this?" Lesson learned reports do not often inform readers on implementation. Therefore, management direction and support are required for lessons learned identification ("Pause After Learning" events), preservation, and publication [21].

In the book *The Smart Mission – NASA's Lessons for Managing Knowledge, People, and Projects*, the authors (former NASA knowledge management and communications leaders) devote an entire chapter, "Stories: Knowledge, Meaning, and Community," to the use of stories as a vehicle for sharing knowledge and lessons learned. Five advantages to communicating through stories are identified; 1) Most people can share a story without having to be trained to do it, 2) Stories facilitate the reflective leadership that is a characteristic of learning organizations, 3) Stories provide context enabling people to find meaning and purpose, 4) Stories help clarify what is important, and 5) Stories connect people through the sharing of thoughts and feelings [34].

The people who are most likely to implement a lesson learned are the ones who learned it the hard way. It is hard for an engineer to influence decision making with a lesson learned or best practice, unless that engineer has been assigned the authority and responsibility for making implementation decisions for a system. As Hoffman, Kohut, and Prusak point out in the book *The Smart Mission*, peers and management must be sold on ideas through conversation and influence. Connecting with an audience and effective persuasion is a social activity [34]. Advocacy of ideas and navigating program politics to influence key decisions is an art form, as John Houbolt's successful advocacy of the lunar orbit rendezvous decision for Apollo demonstrates [35] [36].

Short term concerns about schedule and cost often prevent the review and application of lessons learned whose benefit will not be seen for a long time. Typical responses to an attempt to influence a project with a lesson learned include "That is not my problem," "It doesn't solve a problem I have right now," or "That is a nice idea, but there is no budget or requirement for that." It is difficult to convince someone to act on a lesson learned when they don't know how much they don't know.

A conflict exists between design engineers, who design for limiting cost and spacecraft mass, and flight operations engineers who are interested in lowering life cycle costs and risk during flight operations by increasing systems redundancy and flexibility. Flight operations personnel understand that cost reductions in the near-term vehicle design, can increase overall life cycle costs in the long term, after flights begin.

Formal mechanisms, implemented by management, are needed for including experienced personnel knowledgeable in lessons learned in decision making processes. One solution to this natural tension was Lockheed Martin Integrated Product Teams (IPTs) created during the development of the Orion spacecraft to address various areas of vehicle requirements and design. Experience of members of the various IPTs included human spaceflight, civilian Earth satellites, national security satellites, and uncrewed launch vehicles. An Operations IPT was also created, whose members had experience in human spaceflight operations (Space Shuttle, International Space Station) and uncrewed launch vehicles [37].

Members of the Operations IPT were embedded in the other IPTs to facilitate the communication and consideration of flight operations lessons learned and best practices into the Orion requirements and systems design. Members of the various IPTs recognized the differing concerns of team members and were free to voice opinions and concerns about requirements, design issues, and different solution approaches. The IPT forums permitted informed decisions based on, in part, lessons learned, best practices, and historical data from previous flight programs. These decisions led to operationally sound systems designs that considered life cycle costs and lessons learned [37].

5. GOOD WRITTEN, VERBAL, AND GRAPHIC COMMUNICATION SKILLS ARE NECESSARY

The effectiveness of engineers is dependent on their ability to communicate in speech, in writing, and with illustrations. Brilliant engineers can solve difficult problems and investigate and resolve problems indicated by weak signals. However, if they are ineffective communicators, their ability to convince other engineers and management of the presence of a problem or solution, is limited. Wayne Hale provides an excellent discussion of this in his article titled "Leading Your Leaders" [38]. Furthermore, ineffective communication makes it difficult for subject matter experts to mentor and pass knowledge on to less experienced engineers. While it is good to stress the importance of STEM education, humanities education that develops good communication skills is also necessary.

Former astronaut and NASA Johnson Space Center Director Dr. Ellen Ochoa stated:

> "For those interested specifically in STEM positions, education is important. Math and science are the building blocks of that training, but so are English, writing, and speaking skills. Being able to communicate effectively in this world is vital." [39]

A great deal of attention has been focused on the application of software to facilitate knowledge capture and management. However, before any knowledge can be stored and retrieved by information technology, it must be transferred from the brain of the subject matter expert to the software application. This requires some communication skill on the part of the subject matter expert. Currently that part of the knowledge management process cannot be performed with information technology. Information technology cannot eliminate the need for human communication skills.

6. CAPTURE KNOWLEDGE THROUGH WELL WRITTEN INTERNAL MEMOS

Properly written internal memos can be valuable sources of insight to personnel who are not intimately familiar with the topic. Such insight is usually not found in textbooks or in the open literature, such as in journal articles and conference papers. Internal memos that contain paragraphs of complete sentences are more informative than presentation charts using bullet points.

Internal memos should include an introduction, body, conclusion, bibliography, appendices, and footnotes. The introduction should provide background information so that the reader understands what drove the work or analysis detailed in the memo and what will be done with the results of the analysis. Illustrations and data tables need clear explanations using complete sentences to be useful; do not expect the reader to understand an illustration or table of data simply by looking at it.

Words such as "concise," "brevity," and "it can be shown" are often used in memos to justify why details, derivations, or explanations are left out of a memo to save the author time. Concise and brief writing is appropriate for journal publications with page limits, but when these concepts are applied to internal memos, they frustrate the reader who is not already familiar with the topic. Taking the time to clearly explain a topic reduces the amount of time future readers spend trying to understand it. If a clear and understandable equation derivation is not available in some other publication that can be referenced, the author should provide a detailed derivation in an appendix. In addition, if the word "imply" is used in the memo, explain what is implied. The reader may have difficulty understanding why something is implied.

The memos that are the most informative to readers are those that were written and re-written over time. Memos that are hastily written to meet a deadline are typically of lower quality and of less value to future readers. The time put into making the memo readable speeds up the learning experience later. The subject matter expert should consider that though they may have been living with the topic for years, the reader may be encountering the topic for the first time. Therefore, what is "obvious" and "trivial" to the author is usually not "obvious" and "trivial" to the reader. Before internal publication, a memo should be reviewed by personnel knowledgeable in the topic, as well as an engineer or scientist that is not as familiar with the topic as the author (see Appendix B - Use a Professional Editor). Feedback and changes from such a review can make the memo more understandable to future readers that may never have the opportunity to ask the author questions.

Ray Ryan, a senior software engineer at Square who was interviewed for the book *The Smart Mission*, advocates capturing ideas in writing to enable knowledge transfer through dialogue. Knowledge captured in writing initiates conversations [34].

7. INFORMAL, IN PERSON, FACE-TO-FACE CONTACT IS NEEDED FOR EFFECTIVE RELATIONSHIP BUILDING AND COLLABORATION

Social media and internet applications have made it easier to communicate with people and have facilitated work by teams in different geographical locations, as well as remote work, which was useful during the COVID-19 pandemic. Yet informal, face-to-face communication is also needed for effective collaboration. In the book *The Smart Mission*, the authors state that a small number of in-person, face-to-face meetings quickly establishes trust between geographically separated people [34]. This can lead to timely resolution of problems. However, a flexible policy permitting remote work is beneficial for government agencies and corporations.

A shift from face-to-face communication to electronic communication has made it harder to reach people. Key personnel may avoid email or social media to increase productivity and therefore cannot be reached in a timely manner. Collaborators then become reluctant to reach out for assistance or information since they don't know if people will get back to them, or if they will be heard. Many people prefer remote work, as they believe they are more productive, but it makes it harder to collaborate. Collaboration and brain storming takes time, and some personnel are so busy that they give collaboration a low priority. Remote work makes mentoring and learning more difficult.

Limited collaboration and lack of informal conversation degrades professional relationships, and makes it harder to read people (i.e., pick up subtle signals from body language, tone of voice, etc.). Reading people is hard enough in person and is more challenging in a virtual team environment.

Lack of in-person contact makes it harder for people to learn how to lead, manage, motivate, and understand people, and how to communicate effectively. In-person contact is necessary to allow the next generation of leaders and managers to develop people skills.

One tactic to force collaboration is for a person desiring collaboration to announce in a meeting that resolving this problem requires interaction with people that are difficult to reach. This gets the attention of leadership, and leadership can facilitate the collaboration.

This issue of remote work was recently addressed in an article that appeared in *The Atlantic* titled "The End of Trust" [40]. Lack of face-to-face contact prevents the building of social capitol, which is necessary for trusting relationships to be built in an organization. *The Atlantic* article states that lack of physical contact can be interpreted as a signal of untrustworthiness. According to the author, many supervisors are unsure if their remote employees are performing as well or are as motivated as those in the office.

A friend of the author of this paper, a PhD who supervises researchers in the medical sector, likes the flexibility of being able to work remotely when needed, but believes there must be a balance between effective collaboration in the office and remote working. In her experience, teams that preferred to work remotely were less productive and collaborated less than teams who preferred to work in the office. Remote workers lacked direction and worked lower priority tasks rather than higher priority tasks. While waiting for other people to complete a task they were unproductive, while those waiting in the office shifted their attention to a different task. Remote workers were accomplishing tasks, but overall relationships between team members were not being built and strengthened when many people were working remotely. New employees found it harder to get started and learn the work culture when large numbers of people were working remotely. Remote work made collaboration more difficult, and it proved to be harder to innovate when people were not collaborating. Yet innovation through collaboration is one of the most important hallmarks of space flight research and development, necessitating, some amount of in person interaction.

8. ENGINEERS NEED BETTER PEOPLE AND LEADERSHIP SKILLS

All human and robotic spaceflight is accomplished by people working with technology. Spaceflight is about leading and organizing people to solve problems. These problems involve both engineering and challenges of human relationships and communication. Many with excellent engineering skills are placed in charge of teams. These leaders are often referred to as “lead engineers,” “technical leads,” or “principal engineers.” Engineers have daily contact with engineering leaders and project managers. These leaders often have more influence on the culture of an organization than line management. Therefore, the attitudes and morale of engineers are influenced more by lead engineers than by line management, with whom they interact less often.

Engineers are often appointed to leadership positions primarily due to their technical expertise, even though they may not be considered suitable for a career in management, yet a lack of people and communication skills can have a negative impact on the team, despite demonstrated technical expertise. Some engineers have a desire to lead and influence decisions, but do not have the people and communication skills for the job. They prefer to focus on engineering problems, not people issues. Yet leadership and management is about people. If an engineer does not like to deal with people or is insecure about communicating and interacting with people, they will not do well in an engineering leadership position, regardless of how excellent an engineer they are. Stressful situations can indicate an engineer’s ability to lead and manage. How do their people and communication skills change when they are under stress? Outstanding engineers who lack these skills may better serve as a deputy or technical assistant to an engineering lead or project manager with better skills.

In the book *The Smart Mission - NASA’s Lessons for Managing Knowledge, People, and Projects*, the authors identify the three dominant paradigms of project management: control, processes, and tools. What is not included in these three areas are the human aspects, learning, collaboration, teaming, communication, and culture [34]. Many engineers in leadership positions are comfortable with control, processes, and tools. However, the human aspects are challenging for them.

In her book *Through the Glass Ceiling to the Stars*, former astronaut Eileen Collins outlined leadership principles that she learned in U.S. Air Force officer training [26], 1) Know your job well and perform it with excellence, 2) Know your team (learn about each person) and communicate with them, and 3) Integrity, honesty, and sharing knowledge are important. Learning what is going on requires listening, inquiring about people’s concerns, and avoiding intimidating anyone. Problems can be avoided, and the stress level reduced, by a leader who provides sage advice without preaching. Communication must be clear, and a leader must check to ensure that people understand what has been communicated [26].

Successful leaders have a wide array of personality types and leadership approaches, but five characteristics of successful leaders, collectively called “emotional intelligence,” have been identified by successful business leaders and university professors [41].

Self-Awareness – Good leaders recognize and understand moods, emotions, and drives, and their impact on other people.

Self-Regulation – Good leaders think before acting to control or redirect disruptive impulses and moods.

Motivation – Good leaders can pursue goals with energy and persistence but are not necessarily motivated by money or status.

Empathy – Good leaders understand the emotional makeup of other people and treat them with skill based on their emotional reactions.

Social Skills – Good leaders manage relationships and build networks to find common ground and build rapport.

To effectively lead teams of engineers, engineering leads must have some desire to improve their emotional intelligence. These leaders must be as serious about improving these skills as they are about improving engineering skills and solving engineering problems. They need coaching and mentoring on leadership and management techniques, and emotional intelligence, just like members of line management. Even if classes and mentoring are available for developing these skills, engineers often claim they are too busy to learn to run meetings properly or develop their emotional intelligence. One sign of a potential engineering lead is a desire to develop these skills. While it is important for all engineers to develop better people skills, it is of particular importance for the engineering lead.

9. MEETINGS ARE IMPORTANT, BUT MUST BE RUN PRODUCTIVELY

Meetings are often regarded as taking up too much time and providing few benefits to a project. People overreact to their frustration with meetings by thinking all meetings are bad. While informal collaboration is necessary and beneficial, occasional meetings are needed to communicate with and hold discussions with a wider audience to facilitate decision making and learning. Sharing of information on management concerns and technical problems improves teamwork and morale. Meetings are an important part of the learning process for Mission Control flight controllers and other engineers. Meetings are where engineers learn to sell ideas using logical arguments supported by data [27].

Training is needed to equip engineers and managers to conduct meetings in a manner that is productive [42]. Keys to holding and conducting efficient meetings include 1) Determining whether a meeting needs to be held; 2) Preparing an agenda; 3) Banning electronic devices to avoid distractions through attempts to multi-task, 4) Determining who to invite; 5) If possible, limiting meeting length to an hour; 6) Clearly stating the purpose and objective of the meeting; 7) Asking attendees who have not spoken for input; 8) Avoiding meetings that merely update status; and 9) Following up on actions given to attendees.

There are also techniques for handling attendees that inhibit progress. In his presentation "Lessons Learned from Fifty Years of Observing Hardware and Human Behavior," NASA space suit engineer Joe McMann provides three options for dealing with difficult people in meetings [43]. First, they can be ignored and left off distribution for meeting notices. Second, a meeting organizer can pay lip service to cooperating with them and give the absolute minimum amount of information. However, these options run the risk of offending the person and prompting them to involve management. Third, they can be asked to help, and the meeting organizer might be surprised at how much they contribute constructively to the effort when they feel valued.

An excellent example for those wishing to conduct effective meetings is Bill Tindall, who was in management at the NASA Manned Spacecraft Center in Houston during Gemini and Apollo. During Gemini, Tindall ran the Trajectories and Orbits Panel, and during Apollo, he ran Mission Techniques meetings as Chief of Apollo Data Priority Coordination. Tindall was well-known and appreciated for his excellent ability as a communicator, both in speech and in writing through his memos called Tindallgrams [44-47]. People who were too busy to attend meetings or read everything in their in-baskets made a point of attending meetings chaired by Tindall and reading his Tindallgrams. Such effective meetings result in greater innovation and more effective collaboration and problem solving.

In engineering parlance, Bill Tindall's meetings and memos had a high signal to noise ratio. They provided useful information that helped people do their jobs better and raised and resolved important issues requiring solutions to ensure mission success and safety of flight.

Tindall was not intimidated by a room full of people. He had confidence in his ability to lead a meeting, guide the discussion, make a decision, and convince others that the decision was the right one. Tindall could identify and communicate the purpose of the meeting, what decisions had to be made, and what information was required to make those decisions. He led the meetings in a way that enabled participants to reach a consensus and make a decision, even when those meetings needed to be forced. He then communicated those discussions and decisions to a wide audience.

In the book *Shoot for the Moon*, James Donovan states that Tindall's contribution to the Apollo Program was his ability to get people representing a wide range of teams and organizations to develop the mission techniques, using plain English. Tindall would typically begin a meeting by asking "Why are we here? What are we trying to do?" Tindall created a meeting climate where anyone was allowed to speak their minds without being judged. Tindall used the Socratic Method, offering an opinion, or a summary of what people had said, and then asking for comments. After a lot of debate, the attendees would have a clear idea of what should be done, or at least an idea of what further work was required to get to a solution [48].

Engineers and astronauts had demanding schedules and routinely did not attend all meetings they were invited to. However, they made time for Tindall's Mission Techniques meetings because they were entertaining and informative. The right experts were in the room, different points of view were expressed, important decisions were made, and then Tindall discussed these issues and the decisions in his Tindallgrams [45].

10. INFORMED EMPLOYEES WORK SMARTER AND HAVE A POSITIVE ATTITUDE

It is good for engineers to understand the challenges personnel face during the Design, Development, Test, and Evaluation (DDT&E) and flight operations phases of a program. The more engineers understand the technical and management challenges, who is doing what, what leadership is thinking, and why they are making decisions, the better informed the team will be, resulting in improved internal communication, more positive attitudes, and improved quality of work. It also improves engineer's ability to identify and communicate potential problems. Engineers and management often don't realize the implications of decisions that are made about requirements, design, hardware selection, and risk acceptance. Engineers help protect the program from itself.

Sharing information is key to the early identification and resolution of engineering problems. People need to be informed and stay informed to recognize, raise, and resolve risks to cost, schedule, mission success, and safety of flight. This kind of insight pulls the team together in a way that reviewing priorities, schedules, processes, and accomplishments do not.

Meetings	Tindallgram Memos	Rarely Mentioned
Held to make decisions, not to update status.	Inform people of what decisions were made in meetings and the rationale.	Accomplishments
Define why we are here and what we are doing.	Define work to be done to support decision making in future meetings.	Schedule
Summarize what was said, ask for comments.	Inform people of management and astronaut concerns.	Priorities
Everyone participates, no fear of judgement.	Provide status on problems and who is doing what.	Process
	Detailed discussion of technical topics.	
	Prompt people to provide input.	

Figure 1. Summary of Bill Tindall's use of meetings and memos.

However, gaining an understanding of the big picture is a challenge. Engineers who are focused on solving specific problems often find it difficult to get access to information that provides context for management decisions, and how the problems in each specific area (trajectories, procedures, guidance, navigation, control, communications, propulsion, power, thermal control, robotics, software, hardware, etc.) impact other systems on the vehicle. Minutes or summaries of meetings are rarely created and distributed; such information is passed by word of mouth and interested and curious engineers must ask for it. The use of email, instant messaging, the internet, and social media has not solved the problem. Even in an age of electronic communication and social media, it is easy to facilitate the silo phenomenon (isolation) by the control of who is on distribution lists and who has access to social media communications. Information technology doesn't necessarily bring people together, and it can contribute to isolation.

In addition, some engineers who are very busy and under stress may limit how much they communicate. They have the attitude of "I have gotten what I need from you, I'm not going to provide you with any more information or status unless I need something from you again." This approach makes other engineers feel they are not part of the team. Meetings that could provide people with information may often be cancelled to give people more time to work tasks. This approach, while understandable, does not give knowledgeable engineers a chance to become informed, to think about problems, and provide input on things that other engineers have not thought about. An engineer cannot identify problems if they are not involved.

Effective communication requires skill and initiative by people following the example of Bill Tindall's leadership during the Mercury, Gemini, and Apollo Programs [45] [46]. Bill Tindall was an excellent engineer, but it takes more than engineering skills to lead teams of engineers, software developers, and Mission Control flight controllers; it takes people skills. While issues involving people can be more difficult to solve than engineering problems, the success of space flight programs depends on people working together to solve difficult engineering problems.

The years 1966 through 1969 were the busiest and most stressful phase of NASA's human space flight program to date. Despite the demanding schedules and workloads, Tindall saw himself as a leader of people and continued to engage them in a constructive manner (Figure 1). Given the number of Tindallgram memos from the Mission Techniques meetings, it does not appear that Tindall was canceling meetings. His priority was sharing information and getting people to work together. He was not in a "heads down, leave me alone so I can do my work" mode of operations. Tindall had a unique talent; he would enter a project characterized by chaos and confusion and bring clarity and organization to it.

Most meetings in spaceflight programs cover four topics, schedule, priorities, process, and accomplishments, but Bill Tindall did not limit his communication to these (Figure 1). He communicated the concerns of management, astronauts, and engineers. He let people know what was keeping key decision makers awake at night. Tindall was aware of what the near- and long-term challenges were and shared his concerns. He provided an overview of who was working on what issues. He explained this to a wide audience, not just the people directly working on the problem. What decisions need to be made? What work must be done to support those decisions? What are other people doing?

Tindall understood that he was developing and figuring out how to fly an integrated system of systems. Lots of people in different organizations had to know what was going on. Problems had to be identified that could not be predicted ahead of time. This required that Tindall inform people so that they could identify additional problems, action items and unforeseen impacts.

Tindall took different NASA and contractor groups in the Apollo Program that had been working in isolation and got them to work together. He got them to share information in the Apollo Mission Techniques forums. Tindall was able to integrate the efforts of Mission Control flight controllers, engineers, software developers, mission trajectory planners, and astronauts. These groups often had different views on engineering issues, and Tindall's ability to work through the disagreements and get to the best answer was needed.

Bill Tindall did not assume that everyone had the same level of insight that he did, and he liberally shared that insight. Tindall gave people insight into meetings and hallway conversations that most engineers didn't have access to. Tindall's written communication, known as Tindallgrams, provided information on what decisions were made in meetings, and what needed to be discussed at the next meeting. He kept the dialog going outside the meetings.

Tindall led and motivated people through the content of his communication, he kept them informed and enlightened, enabling them to work smarter, know that they were part of a team, and kept them from feeling that they were kept in the dark. This was his strength, and Don Eyles, an MIT Instrumentation Laboratory engineer who worked on the Lunar Module software, called Tindall's communication "pure candy" in his book *Sundance and Luminary* [47].

Bill Tindall communicated using methods (dictation, hand-written drafts, typed memos, carbon paper, mimeographed copies that were distributed by hand) that are today considered old school, yet he was also an effective communicator when computers came in the form of main frames using time sharing, and before the internet, smart phones, and social media. Despite the hectic schedule and heavy workloads in the 1960s, Bill Tindall prioritized communication. Tindall dealt with people face-to-face. He was friendly but asked hard questions and demanded that people provide proof to support their explanations, conclusions, and recommendations. He frequently used terms like "copacetic," "swinging," and "cool." Tindall's memos and meetings were more informal and combative than today's more polished meetings and high-tech presentations, but much more effective and informative.

This kind of information that Bill Tindall conveyed can be shared in occasional meetings in a succinct manner; lengthy presentations with lots of detail are not needed. But such short communications do the vital work of keeping all members of a team informed about the big picture.

11. FLIGHT OPERATIONS AND ENGINEERING DEVELOPMENT HAVE DIFFERENT CULTURES BUT COMPLEMENT EACH OTHER

Two different cultures exist in a human spaceflight program. Design engineers perform the DDT&E phases of spacecraft development. Flight operations personnel prepare and execute missions with already developed hardware and software through a "plan, train, fly" process. Furthermore, development engineers are concerned with technology innovation, while flight operations engineers are concerned with process and procedure innovation in the "plan, train, fly" process. Engineers representing both cultures work together to design, build, and fly spacecraft. It is necessary to take advantage of the talents and perspectives of both cultures during the DDT&E phases. The differences in these cultures must be understood for effective teamwork and communication.

NASA and contractor engineers that developed the GNC systems for the Orion spacecraft support Artemis missions in the Mission Evaluation Room (MER) in the Mission Control Building. The MER provides reach-back engineering support to the Flight Operations flight controllers in Mission Control. Most development engineers in the MER were not familiar with the culture, duties, and outlook of Flight Operations personnel. The book *Shuttle, Houston: My Life in the Center Seat of Mission Control*, by former Space Shuttle Flight Director Paul Dye, provides an excellent description of Mission Control and what flight controllers do [27]. Quotes from Dye's book were used to acquaint Orion GNC developers with the cultural differences between engineering development and real-time flight operations. This enabled Orion development engineers in the MER to provide better support to the Flight Operations personnel in Mission Control.

The following seven sub-sections highlight the differences between engineering development and flight operations.

The Nature of the Job

Engineering development is often a slow process of theoretical study, analysis, and testing. The goal is to advance the state-of-the-art and determine the cause of problems and devise solutions.

The nature of flight operations is rapid decision-making during simulations and flight. Existing procedures are used, or new procedures are developed and verified to ensure 1) crew safety, and 2) mission success, in that order. The root cause of technical problems is not determined in real-time but is discovered over a period of hours and days by engineering personnel during the flight, or in the weeks following the end of the mission.

Before the flight of Apollo 11, Flight Directors Glynn Lunney, Clifford Charlesworth, and Gene F. Kranz wrote a memo to the flight controllers that were working the mission. The memo provides an overview of the accomplishments of the Mercury, Gemini, and Apollo missions and how they prepared NASA for the Apollo 11 landing. They described flight operations as follows:

> "Once we are in flight, we are doing or preparing for each flight activity as we have trained for. We also must maintain an open mind to handle any unexpected occurrences on its own merits and ramifications as we see it in real time. And, space flight being what it is, we are sometimes presented with situations which are slightly or grossly different or unexpected and require real-time, on-the-spot resolution. That is the challenge of our business." [49]

Technology Innovation versus Process and Procedure Innovation

Engineering personnel advance the state-of-the-art of technology to solve new engineering challenges and facilitate automation and autonomy. Engineering sometimes sees flight operations personnel as not willing to advance and take advantage of the state of the art.

Flight operations uses developed and applied technology to innovate processes and procedures to conduct spaceflight in a "plan, train, fly" process. They ask, "What can be done with the existing system?" The focus is on the crew and Mission Control human element in "plan, train, fly."

Flight operations sometimes sees engineering development engineers as taking too many risks. In some cases, operations may take longer to accept a new technological solution due to unfamiliarity, and a desire for operational robustness and simplicity.

Distinct Disciplines versus an Integrated System

Designing and building a spacecraft requires interaction between engineering disciplines that are distinct and separate in academia. Most development engineers focus on a particular discipline or sub-system on a spacecraft. They understand their component and the underlying theory in detail.

Flight operations personnel consider the spacecraft to be an integrated system. They are concerned with how each component or sub-system impacts the others when failures occur. Flight operations protects crew safety and mission success, in that order, regardless of the number of failures that have occurred.

Former Space Shuttle Flight Director Paul Dye described what was required to be successful in Mission Control.

> "In order to be successful within Flight Operations, you had to understand not only your own system but how it interacted with every other system on the spacecraft. More than that, you had to understand your system's place in the timeline as well as its relative importance in a crisis. …… But most of those design engineers didn't see the big picture – they didn't understand how their equipment might be used in space. They often thought in terms of what was good for their system or component, but not how it melded with all the other systems to achieve the bigger goal of a spaceflight." [27]

Rationale for Making Changes

Engineering prefers to make changes to hardware and software.

Flight operations has a different view of changes. In the short term, proven procedural workarounds that are low risk and simple for the crew and Mission Control to perform and monitor may be preferred over software and hardware changes. Software and hardware fixes may be longer-term solutions due to long lead times for procurement, development, testing, etc. Software and hardware fixes are often the right thing to do and are usually simplest in terms of crew and Mission Control procedures, as compared to procedural workarounds. Cost and schedule are considered, but except for Mission Control software, cost and schedule are not primary factors.

Familiarity with the Latest Technological and Theoretical Developments

Engineering personnel participate in engineering conferences where they interact with other engineers and academics. They keep up to date on recent technological and theoretical developments and can apply these developments to enable new spacecraft to meet new requirements for missions that could not be flown in the past.

Flight operations personnel are familiar with theory and technology underlying the systems they fly with. They do not often participate in conferences. Many are not cognizant of the latest theoretical and technological developments.

Meetings

Engineering may occasionally limit meetings to the smallest number of participants as possible. This is motivated by a fear that too many people in the room will hinder progress and decision making.

Flight operations considers meetings involving multiple disciplines as necessary to develop flight techniques, flight rules, and Mission Control and crew procedures for complex integrated systems that includes both the spacecraft and supporting ground systems. In the absence of documentation, meetings are often important forums for learning about vehicle systems design, performance, and operation. There is a fear that too few people in the room will result in problems not being identified or will result in decisions that will have a negative impact on successful mission execution.

Response to Accident Investigation Reports

A difference between how engineering and flight operations personnel respond to accident investigation reports has been observed in the NASA Johnson Space Center Flight Safety Seminar [15]. Operations personnel (Mission Control, astronaut training) tend to focus on decisions made and actions taken by the people involved, and what they could have done differently.

Engineering personnel tend to focus on hardware and software design issues. How could a different design reduce the chances or consequences of the accident? This sometimes leads to a conclusion that if systems were more automated the risk of human error would be reduced. This is short-sighted, as failures also occur in automated systems.

In response to a belief that more automation would prevent incidents, flight operations personnel are concerned with the human-machine interface, the challenge of monitoring an automated system, and the need to reduce the risk of automation surprises. Dave Matthews, a veteran Space Shuttle and Orion astronaut trainer expresses it this way. An aircraft pilot or astronaut doesn't ever want to be in a situation where they ask, "What is it doing, and how do I make it stop?"

12. IDENTIFY AND FIX PROBLEMS EARLY

Time and budget can be saved by identifying and fixing problems early in development. When problems are identified early, more options to fix them may be available. Resolution of problems discovered late in development is more difficult, as requirements have been defined, and hardware has been designed and built.

There is often resistance to software fixes as a means of avoiding the introduction of additional errors into code. However, the continuous integration and regression testing used by modern software processes lowers technical risk of making changes to fix problems.

Procedural workarounds are often an attractive option for avoiding modification of hardware and software, therefore lowering risk to cost and schedule in the short term. However, such workarounds can become expensive over the long term, once the vehicle is flying, and can complicate operations conceptions and procedures. Procedural workarounds increase the cost, time, and complexity of training of ground personnel, and in the case of human spaceflight, astronauts.

Ground and on-board software developers should not "throw problems over the fence" and expect someone else to fix them later in the project. The longer the delay, the more expensive they are to fix. Involving operations personnel in the requirements definition and testing of software, before it is officially delivered, permits problems to be caught early. Formal requirements and design documents can help ensure that development and operations personnel have consistent expectations about software functionality and how the software works.

This requires development and operations personnel to be proactive about collaborating and communicating, often in a face-to-face manner, before a problem arises that results in direction from management to communicate.

Development of the Orion GNC system required interaction between NASA and contractor engineering development personnel and NASA and contractor personnel that supported the NASA/JSC Flight Operations Directorate (FOD). Several NASA and contractor engineers that supported the NASA/JSC Engineering Directorate in developing the Orion GNC system had previously supported FOD during the Space Shuttle Program, and were familiar with the culture, mindset, and outlook of FOD. These engineers were better able to understand concerns raised by FOD during Orion GNC development and were motivated to address those concerns. FOD personnel were at times surprised at how quickly changes were made during Orion GNC development based on their input.

13. SOFTWARE PROCESSES MUST BE CONTINUALLY IMPROVED

For software processes to work well, they must be subjected to continual scrutiny. Changes must be made to eliminate activities that do not add value to the process, while changing or adding other activities to improve efficiency and the overall performance of software.

The Space Shuttle flight software exhibited outstanding performance over thirty years of missions [50]. The perfection required to achieve the desired level of safety was extremely difficult to accomplish but was aggressively pursued over the life of the program. The flight software development process was continuously improved, automation was implemented to improve both quality and productivity. Once defects and quality escapes were identified, the process was examined and changed to correct the underlying problems. The philosophical approach was that quality must be built into the software at a known level, rather than trying to add quality after the software was developed. Quality could not be tested into the flight software.

14. OBTAINING OBSERVATIONS IMPROVES THE SOFTWARE PROCESS EXPERIENCE

For a software process to be improved, observations on it must be sought from process participants, such as engineers, management, and computer science specialists. Some engineers find it difficult to provide feedback on software processes. Attempts to communicate their observations and frustrations are sometimes met with attempts to educate them on how software processes work, and they are told that if you strictly adhered to the theory of how the process should work things would go better. This frustrates engineers who feel that their concerns and recommendations are not listened to and taken seriously.

In 2021 the author collected observations from NASA and contractor personnel that performed GNC on-board software development for the Orion Artemis I and II missions. Personnel interviewed represented both engineering DDT&E personnel and spaceflight operations (Mission Control) personnel. The collection, preservation, and communication of these observations is intended to inform decision making for future software development. The motivation to perform this task, and the approach taken, was based on the author's previous experience in the Space Shuttle Program. The author was also influenced by the U.S. Army battle lessons pamphlet *Fighting on Guadalcanal*, researched and written by Lieutenant Colonel Russell P. Reeder in 1943 at the direction of U.S. Army Chief of Staff General George C. Marshall [51-53].

The author divided the observations into the following topics.

Analysis Tools
Coding Style and Standards
Documentation
Interface Testing
Information Technology System and Security Challenges
Off-Line Development and Testing
Personnel, Communication, Teams, and Organization
Software Architecture
Software Development Environment
Software Process
Software Requirements
Software Reviews
Software Testing
Use Cases, Scope Creep, and Automation

The resulting 39-page memo consisted of a section on each of the topics. Engineers were eager to schedule time to share their observations, despite heavy workloads. Contributors were told that written observations were welcome. However, one-on-one verbal discussions were preferred, and no written observations were provided to the author, presumably due to work loading. The memo content was based on notes taken during the one-on-one interviews. Contributors were given the opportunity to read and edit the notes before the text was placed in the final version of the memo. Personnel interviewed were not identified by name.

The observations memo did not make recommendations on how future software development should be approached and advised that the observations should not be defined as "lessons learned" or "requirements." Nor was any attempt made to resolve conflicting observations. Definition of requirements and best practices based on the observations are best left to software development and leadership personnel that are working to define the approach to future software development.

One issue mentioned by many engineers interviewed was frustration with having to work with multiple, complex, non-intuitive information technology (IT) systems and software applications. Teamwork and collaboration are necessary to overcome IT challenges, it takes a village to resolve IT issues. IT specialists are overworked and stressed out, and the advice and procedures they provide often assumes that the engineer knows more about the IT system than they really do. In a recent interview at NASA/JSC, SpaceX President and Chief Operating Officer Gwynn Shotwell stated that Elon Musk likes to eliminate frustrations that employees encounter. This includes lowering the number of mouse clicks that must be performed to accomplish a task [54].

15. ENGINEERS NEED BETTER SOFTWARE DEVELOPMENT SKILLS

To reduce cost and shorten development schedules, engineers often perform software development that in previous decades was delegated to professional software engineers with formal education in computer science. However, many engineers (aerospace, mechanical, etc.) are not required to study computer programming at university. Most have some coding experience by the time they graduate, but their experience is limited to coding for themselves for homework assignments, a master's thesis, or a doctoral dissertation. They have never had to write code that has to be examined and understood by other people as part of a formal software development and testing process. Most engineering graduates are very proficient at using software applications. Yet employers prefer engineers who can code proficiently in software such as C++ or Python, rather than engineers whose software experience is limited to applications that require expensive licenses.

Improving the coding skills of engineers would improve the quality of software developed for spacecraft and supporting ground systems. This in turn would lower cost, schedule, and technical risk by avoiding software problems and re-work to resolve issues. The quality of software cannot be improved simply through testing, but by engineers with good coding skills building quality into the software [50].

Engineering students should receive a basic education in software development processes and testing. This includes writing code that is robust, readable, understandable, documented, testable, and can be maintained by other personnel years after it is written. Some software development skills can be acquired by observing or participating in open-source software development projects that use automation, coding standards, conduct unit testing, issue tracking, and code reviews.

16. SOFTWARE IS NOT SELF-DOCUMENTING

The "software is self-documenting" paradigm assumes that code and code comments can be written to provide all the information that is needed to understand what the code does, and the mathematical theory behind the code. This belief justifies elimination of as much documentation as possible to reduce risk to cost and schedule. This eliminates documentation tasks that many DDT&E engineers would rather not perform. Advocates of this approach cite the difficulties of creating and maintaining documentation but pay little or no attention to the long-term value that documentation provides to enable people to understand software functionality.

While informative task and variable names can be created, not all information needed to understand software can be contained in such names. Careful and enlightening comments in code are essential to facilitate rapid and thorough understanding, but not all information about the software design and the theory underlying the algorithms can be contained in code comments.

Visual or graphical programming methods provide an illustration or flow chart of the software, but such visual depictions are difficult to understand without extensive explanations using complete sentences and paragraphs. One of the reasons Apollo and Space Shuttle detailed software requirements were so informative was the use of paragraphs to provide explanations of software functionality, in addition to the equations and logic. These explanations were often more enlightening than flow charts.

GNC algorithms are based on derivations and studies, which can be referenced in code comments. Code comments should reference informal memos, formal reports, presentations, and meeting minutes that cover derivations of equations, algorithm testing that led to algorithm choices and design, analysis to determine numerical constants used by the algorithm, software architecture, and the rationale underlying decisions made about the algorithm implementation and overall software architecture. Still, well commented source code is not a substitute for configuration-controlled requirements and pseudo-code documents, nor can it educate engineers on the mathematical theory used to develop the software algorithm.

Studying extensively annotated software to understand how it works is a time-consuming task for engineers. It can be difficult for an engineer to understand the big picture of what the software is doing, and how multiple software tasks impact the performance of each other. The worse the quality of the code comments, the harder it is for engineers to gain an understanding of what the code does. Good engineers are not necessarily skilled at writing comments for code, which requires good communication skills.

Software, therefore, is not "self-documenting" and believing the contrary leads only to the delivery of software that is difficult to study and understand. This makes problem resolution and maintenance more difficult, and results in long term risk to cost and schedule.

17. SOFTWARE DESIGN INSIGHT IS NECESSARY

When the Space Shuttle was designed and built in the 1970s, it had the most sophisticated software and avionics architecture ever designed for an aerospace vehicle up to that time [55]. From the first flight of the Space Shuttle in April of 1981 to the last flight in July of 2011, performance of both the Primary Avionics Software System (PASS) and Backup Flight System (BFS) was outstanding [50]. Shuttle development veterans stated that during the Space Shuttle DDT&E phase in the 1970s they thought that if a vehicle was ever lost, it would most likely be due to software or some other avionics failure. Such a loss did not occur. *Challenger* and *Columbia* were lost due to hardware failures. This proves the necessity of software design insight, along with rigorous software development and testing process.

The NASA requirement imposed on the Space Shuttle flight software development contractors called for delivery of "error-free flight software." This seemingly impossible challenge drove the development contractors to work hard to develop robust, error-free software [56].

Many factors could be cited for the success of the Space Shuttle on-board flight software, including robust and meticulous development, exhaustive testing, configuration control, issue investigation, and issue resolution processes [50]. Multiple NASA and contractor organizations participated in various phases of requirements development, test scenario development, test data evaluation, and resolution of both testing and flight performance issues. Engineers (engineering development and Mission Control personnel) and skilled computer science specialists interacted extensively throughout the ~37-year Space Shuttle software development and flight program.

Another success factor was the insight that NASA and contractor personnel had into on-board software requirements and functionality. This insight was obtained through several different kinds of documentation.

From the late 1970s through the last flight in 2011, both experienced and new personnel in the numerous Shuttle Program disciplines (engineering, mission planning, Mission Control, astronaut training, etc.), continuously researched flight software functionality. There were multiple reasons for this research – astronaut and Mission Control training, astronaut and Mission Control procedure development, performance investigation, development of training documentation for the crew and other program personnel (engineering analysts, Mission Control personnel, etc.), simulator and mission planning tool development, identification and development of software upgrades, risk assessment, issue resolution, mission planning, trajectory design, etc.

In his book *Shuttle, Houston*, former Mission Control Flight Director Paul Dye explained the importance of design insight.

> "While we had to make decisions at a fast pace while the Shuttle was in flight, we expected decisions to be based on data – not guesses. That meant flight controllers had to study the history of their systems and remember all the various tests and results that occurred over the decades of Shuttle design, construction, and testing. There was no substitute for knowing your system intimately – it was the price you paid for walking through the Flight Control Room door." [27]

Space Shuttle Program personnel generally believe the existence of informative and understandable flight software documentation saved a considerable amount of time and money over 30 years of Space Shuttle missions. Furthermore, the easy access to accurate documentation explaining detailed functionality increased the knowledge and insight of Shuttle Program personnel, enabling them to work better. This, in turn, enhanced mission success and safety of flight.

The existence of such documentation enabled personnel outside the flight software development organizations to conduct daily research and obtain answers without consulting flight software personnel. If the documentation did not exist, computer science specialists would not have been able to address the many questions and requests for information that arose daily as a part of issue investigation, training, mission preparation, and execution.

Some of these documents were written primarily by engineers (not computer science specialists) who understood the software requirements from such engineering perspectives as guidance, navigation, control, trajectories, redundancy management, propulsion, power, communications, thermal control, life support, etc. This made the documents understandable to other engineers, while documentation written by computer science and software process specialists is often difficult for engineers to understand.

Engineers in future human space flight programs will face the same challenges as those in the Shuttle Program. While software processes will change and (hopefully) become more efficient over time, the need will remain for software insight to ensure issue resolution, effective risk management, mission success, and safety of flight. Future human spaceflight programs have flight software automation, autonomy, and functionality requirements far greater than the Space Shuttle. Advances in computer memory and processing speed permit higher complexity code to be executed faster. Like the Space Shuttle, these vehicles will be flying for decades. Space flight programs need the right kind of documentation - documentation that informs people of how the software works and why it was so designed.

Software development processes appear to be focused on reducing cost and schedule over the short term, during initial software development. The desire to reduce software development cost and schedule results in the elimination of as much documentation as possible, even if it is useful. The limited amount of documentation that is produced is focused on software development and requirements verification, but such documentation is rarely used by engineers after the flight software is delivered for testing and flights, so it is rarely informative. Documentation produced by software development processes usually does not consider the need for software insight by engineers that did not participate in software development. That need extends over the life of the flight program, lasting, in the case of the Space Shuttle, for 30 years.

To ensure safe and successful missions, flight software will have to be maintained and understood by many people. These people are not limited to computer science specialists who performed software development but will include people who were born after the first flight of the vehicle. Furthermore, high attrition rates of engineers and computer science personnel drives a need for some form of formal knowledge capture, even on programs that have been in existence for less than a decade. Despite the desire for short term schedules and delivery dates, long term needs for software insight also require consideration due to the impact of lack of knowledge on mission success, safety of flight, and life cycle costs.

18. A CULTURE SHOULD PROMOTE INNOVATION

Critics often complain that engineers propose too many new ideas, and too many of them want to make changes, urging the claim that making more changes is not realistic and presents too much risk. There comes a time when the changes must stop, and when better is the enemy of good enough. There is some justification for this, from a cost, schedule, and technical risk perspective. For example, Bill Tindall's "better is the enemy of good enough" management approach to Apollo flight software development in the mid and late 1960s was one factor that led to first Moon landing before 1970 [45] [46].

Critics also complain that when engineers are asked to come up with new ideas, they propose solutions implemented and proven on previous flight programs. This frustrates leaders who encourage "out-of-the-box" thinking to address new challenges rather than past ones. Leaders who experience this frustration may become less trustful of experienced engineers who have insight and experience that can help a new flight program succeed.

There is a link between these two complaints; 1) engineers want to change things, and 2) engineers keep proposing old ideas. Engineers who spend much of their careers working in operations and sustaining engineering are less inclined to innovate and propose new ideas. They become reluctant to think creatively and tend to propose proven legacy engineering solutions to be successful in the corporate culture. This leads to engineers who are reluctant to innovate and think of new ways to solve problems.

One of the keys of the success enjoyed by some new space commercial companies is to stress the importance of innovation. Continual innovation and new ideas are rewarded by the performance appraisal process which praise engineers for proposing new ideas and wanting to make changes, rather than receiving criticism. Engineers who do not demonstrate a track record of innovation and problem solving are more likely to be let go by new space companies.

In traditional aerospace companies, schedule pressure does not generally motivate engineers to think creatively. Engineers under schedule pressure tend to use familiar technical approaches that have worked rather than to innovate new solution approaches. This reduces technical and therefore cost and schedule risk. New, innovative approaches that are a product of out-of-the-box thinking are high risk and less likely to be proposed by engineers under schedule pressure, therefore engineers proposing innovative new solutions are less likely to be rewarded by the corporate culture.

However, schedule pressure, when used correctly, can facilitate fast decision making and eliminate red tape. Schedule pressure can lead to the elimination of long and arduous decision-making processes that must be completed before engineers can try a new idea. It is necessary to drastically reduce the number of people who must give their approval before something can be started or stopped, or engineers can pursue a new idea. In summary, the proper use of schedule pressure is to make it easier for engineers to try new ideas and get things done quickly.

An innovative leader worth studying is Dr. William B. McLean, who led development of the Sidewinder missile at the Naval Ordnance Test Station at China Lake, California. McLean understood from his experiences during World War II and later at China Lake, the different phases and organizations involved in developing and applying new technology; basic scientific research, applied research, DDT&E, the users (naval aviators and associated support personnel in the fleet), pilot production, and working with contractors manufacturing items on assembly lines.

McLean identified the characteristics of an organization that are necessary for good performance and innovation, as well as responsibilities that came with a culture where there was freedom to innovate [19]. Implementing those characteristics and protecting the innovation culture from encroaching bureaucracy and organizational politics required hard work.

The most effective and efficient innovation was performed on unofficial projects, mostly executed on personal time. The use of personal time enabled McLean to determine who was really interested in the project. He believed that pursuing multiple solutions during development would result in better systems and ultimately save money, even though there appeared to be duplication of effort due to overlapping responsibilities. Like Kelly Johnson of the Lockheed Skunk Works, Bill McLean preferred small teams, even though more people had to become involved as a project matured and approached production and entry into service [19] [57].

The culture at China Lake enabled McLean to quickly change direction and priorities based on what problems he thought were most important and what ideas turned out to be unworkable. Characteristic of such effective leaders as Wernher von Braun and Bill Tindall, McLean could simplify complicated problems and communicate in a way that allowed personnel from different disciplines to understand the problems [45].

McLean observed that bureaucracy not only hindered the flow of information, but also imposed constraints that impeded research and development. The creative process could not be planned and or scheduled. Formal requirements cannot dictate what research and development could be performed. A strict process for selecting what ideas to pursue can discard ideas that are later recognized as promising. McLean's ideology was more successful at allowing new and successful ideas to emerge from research and development [19].

McLean learned that the first idea for solving a problem was usually the most complex approach. After the original idea was implemented, tested, and engineers become more familiar with the physics of the problem, they could devise one or more alternative solution approaches. A second or third idea was usually less complex and more robust than the original approach.

McLean believed in giving engineers enough time to get to the second, improved solution, rather than stopping creative thinking and development too soon. Once the first idea was working, he wanted to see if a simpler solution approach could be created that presented less technical, cost, and schedule risk since they are easier to manufacture, implement, and usually performed better [19]. Once the simpler solution was proven, the original solution approach, originally thought to be "good enough" was judged "not good enough" in favor of the new.

While McLean believed in being flexible and open to frequent and rapid changes early in a project, he realized that once certification for use and production began, changes should be limited to those absolutely required to fix problems. At that point he would praise engineers for their creative ideas, but state that those ideas would best be applied to the next project.

A great deal of institutional resistance may be encountered by leaders and engineers trying to establish a culture that promotes innovation. Once established, management must protect the innovation culture from attempts to make it more formal and bureaucratic.

In the book *No Time for Spectators: The Lessons That Mattered Most From West Point To The West Wing*, General Martin Dempsey, 18th Chairman of the U.S. Joint Chiefs of Staff, points out that most innovation starts with what he calls "responsible rebelliousness" [58]. He distinguishes between responsible and irresponsible rebelliousness as selfless versus self-serving, inclusive versus exclusive, transparent versus opaque, and needing to make an improvement as opposed to changing for the sake of change. Leaders distinguish between responsible and irresponsible rebelliousness and create a safe environment for responsible rebelliousness, while followers responsibly innovate to solve problems and improve processes.

19. FAILURE IS NECESSARY FOR LEARNING

Study, curiosity, experimentation, and failure are needed to learn and innovate new solutions or identify and investigate weak signals of a problem. Engineers learn by performing development of new solution approaches or investigating problems. Engineers can innovate quickly when they have had past experiences and work that enabled them to think, learn, and try new ideas. However, many talented engineers are too busy to innovate and think creatively.

In the 1940s through 1960s, it was normal for an engineer to work on half a dozen different vehicles in the first ten years of their career. By the 1970s it was normal for an engineer to work on one or two vehicles for most of their career. The later generations of engineers don't have as many opportunities to solve problems, learn, and develop their engineering skills as earlier generations of engineers. The same is true of leaders and managers.

Additionally, in government agencies and traditional aerospace companies, development failures are no longer seen as beneficial: instead, failures during development are too quickly attributed to bad management or sloppy engineering. While bad management and sloppy work can cause technical failures, many failures occur simply through attempts to advance the state of the art by trying a new idea. This kind of failure is needed for learning and advancing the state-of-the-art but is not valued or rewarded in government and traditional aerospace cultures.

However, some new commercial space companies believe failure is necessary for technological advances and problem solving. Failure is good if it is a learning experience and leads to advancing the state-of-the-art. However, failure is bad if it is the result of not following procedures or not implementing a previously identified best practice.

20. LEVERAGE THE STRENGTHS OF YOUNGER AND MORE EXPERIENCED ENGINEERS

Spacecraft development and operations are more effective when leaders can leverage the strengths of both more experienced engineers and younger engineers. Younger engineers are enthusiastic and familiar with mathematical theory and solution methods but are unfamiliar with solving real-world problems that do not appear in textbooks and academic literature. Older, more experienced engineers can quickly identify solutions to problems based on experience from previous projects and flight programs. Youthful enthusiasm is not a substitute for sound judgement based on experience.

Younger engineers can collect large amounts of conference papers, journal articles, and documentation from past spaceflight programs, but have difficulty discerning what is important and what is not, and so are unsure what to do with the information. Experienced engineers can more quickly identify what information and resources are needed, and what problems must be worked.

21. SPACECRAFT DEVELOPERS SHOULD WORK CLOSELY WITH ACADEMIA

Spacecraft development can benefit from a close relationship with university professors whose research interests are in problem areas that spacecraft developers are trying to solve. Motivated students who study under these professors can successfully contribute to spacecraft development as interns. For example, during the development of the Orion guidance, navigation, and control system, doctoral work performed by several interns was directly applied to the Orion vehicle. After graduation, these people were hired as full-time engineers working on

Orion and continued to apply their research experience to Orion development [59-66].

While academia has knowledge and skills that are useful to space vehicle development, there may be a lack of understanding of some of the operational and system level constraints. When working with universities and interns, space agencies and companies need to define the real-world problems and associated constraints to effectively leverage the skills and talents of professors and students.

22. SUMMARY

Successful spaceflight, both robotic and human, is accomplished by teams of people innovating, implementing, and using technology. People problems can be more challenging than engineering problems. Lessons learned are not limited to engineering problems, but also to the areas of motivation, inter-personal relations, communication, teamwork, and leadership. For vehicle development and mission execution to be successful, program personnel must be continually learning and developing their skills in both interpersonal and engineering skills.

There are no silver bullet checklists that can be easily executed to create an environment that facilitates technological innovation, or a good flight safety culture. However, continuous training, examination of accident reports, humility, curiosity, identification and questioning of assumptions, learning from failures, responsibility, fighting complacency, developing leadership skills, process improvement, understanding cultural differences of organizations, and clear communication can help ensure successful space vehicle development and mission execution.

APPENDIX A – TOMMY HOLLOWAY ON FLIGHT SAFETY AND MISSION SUCCESS

In November of 1998 Space Shuttle Program Manager Tommy W. Holloway published a memo titled “The Future” [6] [67]. The first shuttle mission in support of the ISS was to be launched in about two weeks and the flight manifest was challenging. Holloway mentioned several close calls within the Shuttle Program and referred to several space and aviation incidents not connected with human spaceflight. Holloway went on to detail eight points that were required to ensure flight safety and mission success.

1. Each member of the Space Shuttle Team is accountable and responsible for his/her task, function, or project. The Program, Safety, and Mission Assurance, or the phantom “They,” etc., are not. We, individually, are responsible.

2. Individual Space Shuttle Team member skills and expertise must be continually pursued and honed. Thinking we know it all and complacency are enemies.

3. Adequate and thorough analysis is mandatory. Understanding the limitation of the analysis is just as important. Using “similarity” and “gut feeling” or “extrapolation” is dangerous. The Mission Evaluation Room’s motto, “In God we trust, all others bring data,” will serve us well.

4. Adequate and thorough testing in the best possible environment is mandatory. Understanding the limitations of the test is as important as understanding the results. Bad tests are worse than no tests; they mislead you.

5. Individual rigor and discipline to do it right are mandatory. Lackadaisical attitude, lack of attention to detail, and not implementing procedures correctly, etc., are precursors to failure.

6. Take time to do it right, to ensure there will be a tomorrow. Cutting corners and hurrying to do a job are sure ways to fail. If you don’t think you have the time to do it right, take time out!

7. Communication and sharing of data, concepts, and ideas across the Space Shuttle Team are the checks and balances that keep us on track. Not having data is bad; not sharing is worse.

8. Learn from close calls. We should not only investigate the specific close call but review like areas in other systems, processes, and designs.

APPENDIX B – USE A PROFESSIONAL EDITOR

The quality of internal and publicly available papers and presentations can be improved through a review by an editor and graphic artist since these professionals have university-level training and experience in communicating with different types of audiences through word and picture. An editor or graphic artist does not have to understand the science and engineering to improve the quality of a publication; a liberal arts degree equips these professionals to think creatively and critically and communicate with greater clarity to a wider audience. The average engineer would do well to make use of these skills.

However, some humility on the part of the engineer is required. An engineer finished with a publication, can feel proud of their work and resistant to comments to improve it. But a willingness to accept and act on editorial comments, and an understanding that work is not finished until it is well communicated to the intended audience, will improve the publication and develop the engineer’s communication skills.

A peer review by fellow engineers is useful as well. However, not all input from engineers is necessarily helpful; a professional editor or graphic artist can help the writer sort through comments from other engineers and choose those modifications that truly improve the publication.

REFERENCES

[1] John L. Goodman, "Space Shuttle Navigation in the GPS Era," *Proceedings of the National Technical Meeting 2001*, Institute of Navigation, Long Beach, CA, January 22-24, 2001, pp. 709-724.

[2] Leonard Kramer and John L. Goodman, "Ionospheric Instability Observed in Low Earth Orbit Using Global Positioning System," *AIAA Journal of Spacecraft and Rockets*, Vol. 42, No. 1, January-February 2005, pp. 124-131.

[3] John L. Goodman, J. P. Brazzel, and D. A. Chart, Ph.D., "Challenges of Orion Rendezvous Development," *AIAA Guidance, Navigation, and Control Conference*, AIAA, Reston, VA, 2007.

[4] John L. Goodman and Carolyn A. Propst, "Operational Use of GPS Navigation for Space Shuttle Entry," *IEEE/ION PLANS 2008 Conference*, Monterey, CA, May 5-8, 2008.

[5] John L. Goodman, H. R. Mamich, and D. W. Saley, "Orion On-Board Navigation Architecture and Operations Concepts," *Guidance and Control 2008, Advances in the Astronautical Sciences*, Vol. 131, Univelt, San Diego, CA, 2008, pp. 425-444.

[6] John L. Goodman, *History of Space Shuttle Rendezvous, Revision 3*, JSC-63400, Flight Dynamics Division, Mission Operations Directorate, NASA Johnson Space Center, October 2011, p. 7. Available on the Internet from the NASA Technical Reports Server.

[7] Shane B. Robinson, Sara K. Scarritt, and John L. Goodman, "Encke-Beta Predictor for Orion Burn Targeting and Guidance," *Guidance, Navigation, and Control 2016, Advances in the Astronautical Sciences*, Vol. 157, Univelt, San Diego, CA, 2016, pp. 709 to 731.

[8] John Goodman, "Approaches and Fly-Arounds for Spacecraft Proximity Operations," in *Spaceflight Mechanics 2017, Advances in the Astronautical Sciences*, Vol. 160, Univelt, Inc., San Diego, CA, 2017, pp. 903-922.

[9] Thomas J. Fill, John Goodman, and Shane Robinson, "Orion's Powered Flight Guidance Burn Options for Near Term Exploration Missions," *Guidance, Navigation, and Control 2018, Part I, Advances in the Astronautical Sciences*, Vol. 164, Univelt, San Diego, CA, 2018, pp. 525-544.

[10] John L. Goodman, "The Space Shuttle and GPS – A Safety-Critical Navigation Upgrade," *COTS-Based Software Systems*, edited by H. Erdogmus and T. Weng, Springer-Verlag Lecture Notes in Computer Science Volume 2580, Springer-Verlag, Berlin, Germany, 2003, pp. 92-100.

[11] John L. Goodman, "A Software Perspective On GNSS Receiver Integration and Operation," *Satellite Navigation Systems: Policy, Commercial and Technical Interaction*, edited by M. Rycroft, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2003, pp. 119-126.

[12] John L. Goodman, "Knowledge Capture and Management - Key To Ensuring Flight Safety and Mission Success," *AIAA Space 2005 Conference*, Long Beach, California, August 30-1, 2005.

[13] John L. Goodman, *Knowledge Capture and Management for Space Flight Systems*, NASA Contractor Report NASA/CR-2005-213692, NASA Johnson Space Center, October 2005.

[14] John L. Goodman, *GPS Lessons Learned From The International Space Station, Space Shuttle and X-38*, NASA Contractor Report NASA/CR-2005-213693, NASA Johnson Space Center, November 2005.

[15] John L. Goodman and David P. Dannemiller, "Flight Safety Education for Current and Future Space Flight Programs," *Proceedings of the AIAA Space 2006 Conference*, AIAA, Reston, VA, 2006.

[16] John L. Goodman, *Lessons Learned From Seven Space Shuttle Missions*, NASA Contractor Report NASA/CR-2007-213697, NASA JSC, January 2007.

[17] John L. Goodman, *Best Practices for Researching and Documenting Lessons Learned*, NASA Contractor Report NASA/CR-2008-214777, NASA JSC, March 2008.

[18] John L. Goodman, "Apollo 13 Guidance, Navigation, and Control Challenges," *AIAA Space 2009 Conference & Exposition*, Pasadena, CA, September 14-17, 2009.

[19] John L. Goodman, "Innovation Experiences From Vanguard, Explorer 1, Sidewinder, and NOTSNIK," *Guidance, Navigation, and Control 2018, Part I, Advances in the Astronautical Sciences*, Vol. 164, Univelt, San Diego, CA, 2018, pp. 965-978.

[20] John L. Goodman, *History and Archives Contribute to the Success of Space Flight Programs*, NASA/CR-2018-220044, NASA JSC, December 2018.

[21] John L. Goodman, "Where Did that Equation Come From? A Lean and Agile Approach to Preserving Aerospace Knowledge," *70th International Astronautical Congress*, Washington, DC, October 21-25, 2019.

[22] John Goodman, "A Cautionary Tale of a Secret, a Small Team, an Accelerated Schedule, and the Gemini IV Station-Keeping Failure," *Guidance and Control 2020, Advances in the Astronautical Sciences*, Vol. 172, Univelt, Inc., San Diego, CA, 2020, pp. 437-452.

[23] Diane Vaughan, *The Challenger Launch Decision: Risky Technology, Culture, and Deviance at NASA*, University of Chicago Press, Chicago, 2016, pp. xii, xxxi, 62-64, 75.

[24] Diane Vaughan, "The Trickle-Down Effect: Policy Decisions, Risky Work, and the Challenger Accident," *California Management Review*, Vol. 39, No. 2, Winter 1997, pp. 80-102.

[25] *Columbia Accident Investigation Board Report, Volume I*, U.S. Government Printing Office, Washington, DC, August 2003, pp. 97, 130, 180, 181, 183, 192, 196-197, 203.

[26] Eileen M. Collins, and Jonathan H. Ward, *Through the Glass Ceiling to the Stars: The Story of the First American Woman to Command a Space Mission*, Arcade, 2021, pp. 75, 84, 134, 153, 209, 250-251, 254, 284, 285.

[27] Paul Dye, *Shuttle, Houston: My Life in the Center Seat of Mission Control*, Hachette Books, 2020, pp. 82, 100, 231, 270.

[28] Gene Kranz, *Failure Is Not An Option*, Simon & Schuster, New York, 2000, pp. 203-204.

[29] Wayne Hale, "Adjusting Our Thinking," *Wayne Hale's Blog*, January 13, 2009. https://blogs.nasa.gov/waynehalesblog/2009/01/13/post_1231872667823/. Accessed March 28, 2022.

[30] Matthew Kohut, "Interview with Wayne Hale," *ASK Magazine*, NASA, October 17, 2010. https://appel.nasa.gov/2010/10/17/interview-with-wayne-hale/. Accessed March 28, 2022.

[31] James Oberg, "NASA Has To Fight The Forgetting," https://www.nbcnews.com/id/wbna16830696, January 26, 2007. Accessed February 25, 2022.

[32] S. A. Snook, *Friendly Fire – The Accidental Shootdown of U.S. Black Hawks Over Northern Iraq*, Princeton University Press, Princeton, New Jersey, 2000.

[33] J. Russell Carpenter, Christopher N. D'Souza, F. Landis Markley, and Renato Zanetti, *Navigation Filter Best Practices*, NASA/TP–2018–219822, NASA, April 2018.

[34] Edward J. Hoffman, Matthew Kohut, and Laurence Prusak, *The Smart Mission – NASA's Lessons for Managing Knowledge, People, and Projects*, The MIT Press, Cambridge, MA, 2022, pp. vii, 9-10, 21, 52-54, 75-76, 96-97.

[35] J. R. Hansen, *Enchanted Rendezvous*, NASA Monographs in Aerospace History Series #4, NASA, Washington, D. C., 1995.

[36] William F. Causey, *John Houbolt: The Unsung Hero of the Apollo Moon Landings*, Purdue University Press, 2020.

[37] Steven M. Jordan, "Healthy Tension – The Role of Operations in the Orion Spacecraft Design," *AIAA SPACE 2009 Conference & Exposition*, Pasadena, CA, September 14-17, 2009.

[38] Wayne Hale, "Leading Your Leaders," *Ask Magazine*, NASA, Washington, DC, Winter 2008, pp. 8-11. https://appel.nasa.gov/wp-content/uploads/2013/04/435953main_ASK_29_Winter _2008.pdf. Accessed April 20, 2022.

[39] Jennifer M. Ross-Nazzal (editor), *Making Space for Women - Stories from Trailblazing Women of NASA's Johnson Space Center*, Texas A&M University Press, College Station, TX, 2022, p. 382.

[40] Jerry Useem, "The End of Trust," *The Atlantic*, December 2021. https://www.theatlantic.com/magazine/archive/2021/12/trust-recession-economy/620522/, accessed March 2, 2022.

[41] Daniel Goleman, "What Makes a Leader," in *On Emotional Intelligence – HBR's 10 Must Reads*, Harvard Business Review Press, Boston, MA, 2015.

[42] *HBR Guide to Making Every Meeting Matter*, Harvard Business Review Press, Boston, MA, 2016.

[43] Joe McMann, "Lessons Learned from Fifty Years of Observing Hardware and Human Behavior," November 27, 2011, p. 50. Available from the NASA Technical Reports Server (NTRS) on the internet, https://ntrs.nasa.gov/citations/20140016944, accessed September 6, 2022.

[44] Andrew Baird, "How to Land Next to a Surveyor: Bill Tindall and the Apollo Pinpoint Lunar Landing," *Quest: The History of Spaceflight Quarterly*, Vol. 14, No. 2, May 2007, pp. 19-27.

[45] John L. Goodman, "Bill Tindall, Master Integrator of Gemini and Apollo," *45th Rocky Mountain AAS GN&C Conference*, Breckenridge, CO, February 3-8, 2023.

[46] Charles Fishman, *One Giant Leap, The Impossible Mission That Flew Us to the Moon*, Simon & Schuster, New York, 2019, pp. 140-162, 176, 182-183, 396n1, 398n13.

[47] Don Eyles, *Sunburst and Luminary, An Apollo Memoir*, Fort Point Press, Boston, MA, 2018, pp. 170-171.

[48] James Donovan, *Shoot for the Moon*, Little, Brown, and Company, New York, 2019, pp. 282-283, 283-284, 317, 347, 353, 367, 422.

[49] Glynn S. Lunney, Clifford E. Charlesworth, and Eugene F. Kranz, "The Lunar Landing Mission," July 1969.

[50] Christopher Hickey, Andrew Klausman, Brad Loveall, and James Orr, "The Legacy of Space Shuttle Flight Software," *AIAA Space 2011 Conference & Exposition*, Long Beach, CA, September 27-29, 2011.

[51] Russell P. Reeder, *Fighting on Guadalcanal*, U.S. Government Printing Office, Washington, DC, 1943.

[52] Forrest C. Pogue, *George C. Marshall, Organizer of Victory, 1943-1945*, The Viking Press, New York, 1973, pp. 75-76.

[53] Red Reeder, *Born at Reveille, Memoirs of an American Soldier*, Vermont Heritage Press, Quechee, VT, 1994, pp. 209, 212, 227-239, 327-328.

[54] Gwynn Shotwell, President and Chief Operating Officer of SpaceX, interviewed by NASA/JSC Director Vanessa Wyche, NASA/JSC Town Hall with Fireside Discussion, November 9, 2022.

[55] John F. Hanaway, and Robert W. Moorehead, *Space Shuttle Avionics System*, NASA-SP-504, NASA, Washington, DC, 1989.

[56] Jack Clemons, *Safely to Earth – The Men and Women Who Brought the Astronauts Home*, University Press of Florida, 2018, pp. 156-163.

[57] Kelly Johnson's 14 Rules and Practices, http://www.lockheedmartin.com/us/aeronautics/skunkworks/14rules.html. Accessed February 28, 2022.

[58] Martin E. Dempsey, *No Time for Spectators: The Lessons That Mattered Most From West Point To The West Wing*, Missionday, 2020, pp. 191, 205.

[59] S. K. Scarritt, B. G. Marchand, A. J. Brown, W. H. Tracy, and M. W. Weeks, "Finite-Burn Linear Targeting Algorithm for Autonomous Path Planning and Guidance," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 5, September-October 2012, pp. 1605-1615.

[60] C. Spreen, and K. C. Howell, "Automated Node Placement Capability for Spacecraft Trajectory Targeting Using Higher-Order State Transition Matrices," *AAS/AIAA Astrodynamics Specialist Conference*, Columbia River Gorge, Stevenson, WA, August 21 - 24, 2017.

[61] David Woffinden, Renato Zanetti, Kirsten Tuggle, Christopher D'Souza and Peter Spehar, "Re-Evaluating Orion's Relative Navigation Filter Design for NASA's Future Exploration Missions," *Guidance, Navigation and Control, Advances in the Astronautical Sciences*, Vol. 164, 2018, pp. 1083-1102.

[62] Kevin Kobylka, and John Christian, "Methods in Triangulation for Image-Based Terrain Relative Navigation," 31st AAS/AIAA Space Flight Mechanics Meeting, February 1-3, 2021, in *SPACEFLIGHT MECHANICS 2021, Advances In the Astronautical Sciences*, Vol. 176, Univelt, Inc., San Diego, CA, 2021, pp. 2507-2526.

[63] James S. McCabe, "On A Higher Order Method for Anonymous Feature Processing," *AIAA SCITECH 2022 Forum*, San Diego, CA, January 3-7, 2022.

[64] Sergio A. Sandoval, Ping Lu, John T. Hwang, Jeremy R. Rea, and Ronald R. Sostaric, "Multiple Optima in Abort Ascent During Lunar Powered Descent," *AIAA SCITECH 2022 Forum*, San Diego, CA, January 3-7, 2022.

[65] E. M. Zimovan-Spreen, K. C. Howell, and D. Davis, "Dynamical Structures Nearby NRHOs With Applications to Transfer Design in Cislunar Space," *Journal of the Astronautical Sciences*, Vol. 69, Issue 3, June 2022, pp. 718-744.

[66] C. York, K. Howell, and B. Marchand, "State Perturbation Tradespaces in Trajectory Targeting Convergence Analysis," *AAS/AIAA Astrodynamics Specialist Conference*, Charlotte, NC, August 7-10, 2022.

[67] Tommy W. Holloway, "The Future," MA-98-068, Space Shuttle Program, NASA JSC, November 20, 1998.

BIOGRAPHY

John L. Goodman *has worked as a contractor at the NASA Johnson Space Center for over 36 years, supporting both the Flight Operations and Engineering Directorates. His experience includes the Space Shuttle, X-38, International Space Station, Constellation, Commercial Crew, and Artemis Programs. Areas of work include software testing, software development, rendezvous and GPS navigation, burn targeting and guidance, rendezvous profile design, flight safety education, knowledge management, and lessons learned. He has written and illustrated training materials used by engineers and Mission Control personnel on these topics.*