


TEN SYSTEMS ENGINEERING LESSONS LEARNED

BY JOHN RUFFA

When I was appointed the mission systems engineer of the Solar Dynamics Observatory (SDO) at Goddard Space Flight Center, I was understandably nervous. While I had served in a variety of technical leadership positions on in-house spacecraft development efforts, the all-encompassing systems-level responsibility of the mission systems engineer position seemed daunting.

A full-disk, multiwavelength, extreme ultraviolet image of the sun taken by SDO on March 30, 2010.



A massive plume of dense, cool (only compared with the rest of the solar atmosphere) plasma erupts on the sun's surface, flowing in a loop along a magnetic field line.

Photo Credit: Goddard Space Flight Center/Atmospheric Imaging Assembly

Fortunately, I had the privilege of working with a number of experienced systems engineers prior to SDO and had a strong technical team to help me navigate the many technical challenges we would face. What surprised me was how many *non-technical* issues I would ultimately face on this mission.

Most systems engineering training focuses on the technical issues, often with very little focus on helping the systems engineer understand and learn to deal with the non-technical minefields that are part of every project. Like technical issues, non-technical issues also have the potential to slow or derail progress.

Realize Most Problems Are Non-Technical

This was one of the biggest surprises that I have found as an engineer and the one for which I received the least amount of training and instruction. At the start of SDO, one of the first things we did was identify driving issues—the problems and challenges we considered the greatest threats to mission success. Little did we know that these technical issues were only a subset of our problems.

Early in the SDO development effort, our systems team started formulating the concept for a reliable, high-performance spacecraft-avionics architecture that would serve as the backbone of our solar-science observatory. Many on our team had just completed a successful in-house spacecraft, the Wilkinson Microwave Anisotropy Probe (WMAP). It seemed to make sense to build upon the foundation of this previous effort and pursue a similar approach.

Nailing down the design and getting buy-in from key players was prolonged and painful, however, often resulting in conflict. Throughout the process, I was puzzled why an approach that was so successful only a few years earlier had turned into a nightmare on SDO. It turned out the influence of non-technical issues was greater than I'd known. Just because an approach was once successful with one team did not guarantee success with a completely different team, one with its own mind-set and biases.

These issues can manifest themselves through poor communication, turf battles, conflicting agendas, technical disconnects, conflicting cultures, and conflicting personalities. Anyone who has worked in a team environment is familiar with these problems. Non-technical issues that complicate communication and the open exchange of information make the technical challenges even more difficult.

Understand and Define Your Team Culture

Every team has a culture—an unwritten philosophy of how a team works, communicates, and interacts internally and to the outside. A team's culture helps define its work ethic, its attention to detail (or lack thereof), how well (or poorly) people are treated, whether questions are openly asked or discouraged, whether it is detail (or “big picture”) oriented, and how it approaches troubleshooting and problem solving. Some teams are meticulous, some more casual, some very process-oriented, others less rigid, some open to give-and-take discussions, others more regimented in their communication. Many teams are unaware that their culture can influence mission success.

Early in my career at NASA, I worked with a senior systems engineer who was meticulous in spacecraft testing and troubleshooting, and whose strength in this area contributed to the success of numerous satellites. He strongly espoused the regular use of the formal problem-reporting system to document, track, and close out issues discovered during testing. The engineering team was reluctant to formally document issues in the system. Some of it was laziness, some of it stemmed from the cumbersome nature of the system, and a large part of it was the perception that entering a large number of issues into the system would somehow tag our development effort as being more troubled or problematic than others.

Fortunately, our senior engineer constantly emphasized that the problem-reporting system was simply a valued tool to make sure that issues were properly identified, investigated, reviewed, and closed out in a rigorous manner. Instead of

TODAY, AS I LOOK AT THE ENGINEERS WHO “GREW UP” ON THAT PROGRAM AND NOW HAVE SPREAD THROUGHOUT GODDARD, I SEE THE FRUITS OF THAT CULTURAL CHANGE AND THE EFFECT IT STILL HAS IN HELPING TO ENSURE RELIABLE SPACEFLIGHT HARDWARE.

A great deal of plasma (hundreds of millions of tons) is unable to escape the gravitational pull of the sun after a prominence eruption and falls back down as “plasma rain.”

Photo Credit: Goddard Space Flight Center/Atmospheric Imaging Assembly

making our project seem more risky, he claimed that fully documenting issues would enhance the overall reliability and, accordingly, the confidence we and our NASA center would have in our finished product. He worked with the project manager to change the culture of the engineering team, promoting the proper use of the problem-reporting tool and actively correcting the misperceptions that formally documenting problems would mark the project as troubled. This effort changed the project engineering team culture and the manner in which we investigated, addressed, and closed out issues.

Today, as I look at the engineers who “grew up” on that program and now have spread throughout Goddard, I see the fruits of that cultural change and the effect it still has in helping to ensure reliable spaceflight hardware.

Find a Mentor

On my first flight project, our team presented a spacecraft communication-interface approach we had developed to our NASA review team. Although we were young and relatively new to the world of spacecraft design, we had come up with an approach we were proud of. So it was a huge disappointment when a senior member of our review team quickly demonstrated the complex and cumbersome nature of our implementation. He offered a simple, elegant alternative that was a significant improvement over our “homegrown” concept.

Immediately after the review, I thanked him for his input and asked if we could talk to him about other aspects of our design implementation. This was the beginning of a long and fruitful working relationship. He became a trusted mentor and friend not only to me, but to other members of my team.

Systems engineering covers an astonishingly broad area of mission requirements, design/implementation details, and operations concepts. It is impossible for any individual to possess sufficient experience or expertise to understand the complete system and its nuances and issues. A wise systems engineer will build an

informal list of more experienced engineers as go-to contacts for dealing with the many technical (and non-technical) issues that will inevitably arise. This fellowship of mentors and peers will become one of the most valuable tools in the systems engineer’s toolbox.

Don’t Reinvent the Wheel

When our systems team was assembled on SDO, one of the first things we did was ask ourselves, “Who has done this type of mission before and what can we learn from them?” We sought out knowledgeable people from other missions and picked their brains for helpful implementation details and lessons learned. Even so, we missed obvious mission contacts who, in retrospect, would have helped us tremendously.

For example, while we aggressively pursued information and design details from other solar-science missions, we didn’t contact other missions that used geosynchronous orbits until much later in our development effort. It would have been very helpful to spend more time talking to the geosynchronous spacecraft designers to discover issues they faced that differed from our previous orbital-design experiences.

Engineers often spend tremendous effort trying to come up with a unique solution rather than build on the foundations of others. A wise individual I once worked for was fond of saying, “When you are in college and you copy someone else’s work, it’s called plagiarism, and it can get you kicked out of school. In the world of engineering, this is called good engineering practice, and it often results in awards and promotions.”

Aggressively avoid the trap of “not invented here” that prevents you from tapping the experience of those who came before. You will be the better for it and, in the process, you might further build your informal network of peers and mentors.

Realize That People, Not Positions, Get the Job Done

Selecting the right people for specific positions, roles, and responsibilities will always make the difference when storms

SDO's Atmospheric Imaging Assembly instrument captured this image after a solar eruption and a flare.

(technical or otherwise) hit. This may seem obvious, but it is astonishing how often some leaders are content to fill positions rather than build a team.

Anyone who has worked in a team environment can probably recall an example of a well-intentioned individual who, for whatever reason (lack of experience or underdeveloped interpersonal or communication skills, among others), was a poor fit for a key role on a team. When this occurs, the rest of the team struggles to compensate for the deficiency. This often means either forcing the team to add unplanned additional personnel to augment shortcomings in this key role or learning to “work around” the individual in question.

Having the right person can make a huge positive difference. I recall a time on SDO when the value of talent was recognized and used to augment the existing team. Late in the development effort, we brought in a highly skilled individual to perform technical reviews. After they were completed, rather than let this valuable individual go, I went to the project manager and requested bringing this engineer on full time. I confessed that I hadn't thought through the specific role this individual would fill but emphasized the principle that skilled people are rare, and we should grab them first and ask questions later.

Fortunately, our project manager agreed, and this engineer stayed through the rest of the project, solving many technical issues and performing as a key member of our systems team. Even though we didn't have a particular position that needed filling, we saw the value of a specific individual, realized the potential benefit to the team, and grabbed him.

Tear Down Barriers to Open Communication

On every project there are people who choose not to communicate openly with their counterparts. As a result, communication lines atrophy, slowing or stopping the transmission of critical information and risking technical disconnects. A wise team lead will aggressively address communication issues as they arise. Sometimes all it takes is to

remind people of the need to communicate and the potential consequences of dropped information.

The corollary principle must also be followed—make every effort to promote positive and open communication, whether it is by face-to-face meetings, walking around and touching base with team members, or doing whatever it takes to foster regular, open communication and build positive working relationships.

Recognizing the importance of clear and open communication in solving and preventing problems, our SDO systems engineering team instituted a weekly team meeting. It became a valuable time to not only solve technical issues, but to work through disagreements and differences. In addition, occasionally we would meet to self-assess our team and honestly discuss how we were doing and whether there were areas that could be improved. Outside the meetings, I would make a point to follow up with team members to make sure there were no hidden issues or concerns that were not getting adequate exposure in our group meeting.

These simple actions are not remotely groundbreaking, which is exactly the point: communication does not need to be elaborate or innovative, it just needs to happen.

Talk to the People Who Actually Do the Work

One of my engineers came into my office to talk about a technical problem, quietly indicating that what I thought was a technical issue was really due to issues in the working relationships between key individuals. When I asked why no one had told me about this, he sighed and said, “Of course no one at the working level is ever going to approach the mission systems engineer to have that kind of conversation.”

This was the first time I realized that I had now risen to a place in the organizational chart that created barriers that would impede my understanding of daily issues on the work floor. From that day onward, I started making a deliberate effort to “walk the floor,” asking questions and listening to the answers (whether I liked them or not).

THESE SIMPLE ACTIONS ARE NOT REMOTELY GROUNDBREAKING, WHICH IS EXACTLY THE POINT: COMMUNICATION DOES NOT NEED TO BE ELABORATE OR INNOVATIVE, IT JUST NEEDS TO HAPPEN.

This lesson should not have been a revelation. When I was a young engineer, I struck up a friendship with a senior manager of the engineering directorate at Goddard. Every two or three months, he would give me a call, invite me into his office, and we would talk about how things were going, what I liked about my work and the organization, what I didn't like, and what needed improvement. I learned years later that this was part of a calculated effort on his part to stay in touch with people within his organization. He regularly met with junior members of the department to gain a "boots-on-the-ground" perspective of what was really going on.

On every project, there are the people who are in charge and the people who actually do the work. These key workers often can tell you the most about what the problems really are, what to watch out for, and how to creatively solve problems—and they will figure out quickly if you really want to listen. A team lead who walks the floor will be far better equipped to accurately gauge the issues, understand their impacts, and formulate appropriate responses than one who stays in his office.

Beware "Groupthink"

We admire finely tuned teams that share philosophy and culture and can almost finish each other's sentences because of their excellent teamwork. Therein lies a trap that must be avoided: becoming so well integrated that groupthink creeps in and eliminates valid opposing viewpoints, causing a team to miss alternative approaches or, even worse, miss hidden concerns until they become real problems. The team lead must take pains to cultivate an environment where outside reviews and internal minority opinions are not only acceptable but actually sought out as part of the normal process of doing business.

On SDO, our project management and systems engineering teams worked hard to cultivate an environment where the team took the review process seriously as a valuable tool (rather than a necessary evil) and saw our review teams as partners in developing a successful mission. After our design passed

through the critical design review, our project manager made a habit of updating critical review team members, briefing them on significant issues or changes, even when these fell outside the normal review "gates." As a result, we developed a positive working relationship with our review team and kept them abreast of issues, helping them to be better educated in their review and assessment of our progress.

Internally, we focused on creating an environment where the systems team regularly reviewed and questioned major design decisions and issues. Our weekly systems team meeting served as an anchor to ensure that honest and open discussion occurred, and frank communication also occurred at other project meetings, including design/development meetings and risk meetings. We had no shortage of people willing to challenge the status quo and take on devil's advocate positions. While this give-and-take discussion could sometimes be frustrating, in the end it resulted in a better team and a more reliable mission.

Build and Preserve a Sense of Ownership and Responsibility

One of the biggest challenges for a strong, dynamic leader is to guide team members without diminishing their sense of ownership and responsibility. When we started SDO, many of us were new to our leadership roles and excited about the opportunity to shape this new project. The in-house design teams typically see in-house missions as a prime place for pushing the technical design boundaries in order to advance the state of the art, however, and had their own ideas about design and new technology approaches. This often led to conflicts between the systems engineering and subsystem design teams.

Ultimately, the systems team is the technical conscience of the mission-development effort and has the responsibility to ensure that the trades and compromises made are in the overall best interest of the mission. Looking back, I suspect there were times where our focus and sense of ownership may have unintentionally caused some of our design teams

TIME AND AGAIN, I SEE THE FRUSTRATION SENIOR ENGINEERS MAY HAVE WITH THOSE LESS EXPERIENCED SLOWLY MELT AWAY AS THEY UNDERSTAND THE VITAL ROLE THEY HAVE IN PASSING THEIR KNOWLEDGE AND EXPERIENCE TO OTHERS.

to feel that their own sense of ownership and responsibility was undercut.

When talented individuals start sensing that their ownership or technical responsibility is being eroded or second-guessed, they may fight back, attempting to reassert their roles, or they may recognize the futility of their efforts and become passive. The challenge of the team lead is to prevent both outcomes by not usurping the roles of those underneath him or her, but guiding them in a constructive fashion while preserving the higher-level system goals.

Train Your Replacement

A wise senior systems engineer often reminds me that any job has two primary components: to do your work with excellence and integrity, and to train your replacement.

Until you train your replacement, you cannot leave your current position, since your departure would leave a hole behind. Also, the train-your-replacement mentality creates a fertile environment where the skills of an organization are continually replenished through mentoring and passing of the baton. Finally, having a train-your-replacement mind-set transforms the way we view and deal with other members of our team. Time and again, I see the frustration senior engineers may have with those less experienced slowly melt away as they understand the vital role they have in passing their knowledge and experience to others. Not only does this promote open technical interchange, it also creates a nurturing and team-building environment.

On an earlier mission, when I was ready to take on the new challenge of a systems engineering role, the project manager insisted that I first identify and train an individual to take my place as a flight-component lead. The individual assigned to take my place had far more skill and experience in detailed flight-hardware design than I did, but he had never had the role of coordinating design and testing of a flight component. I was able to work closely with him to broaden his already impressive skills into a new area. In the same way, the systems engineering

lead on the project was helping me grow into my new role. The added benefit of this approach is that the mentoring relationship provides a natural safety net of peers and mentors in the event that a person struggles in a new role.

Be Aware

My list of non-technical issues is almost certainly incomplete. My aim is not to exhaustively catalog all the non-technical threats that engineers may face, but to raise awareness of the impact these kinds of issues can have on a technical-development effort. That awareness is the first step toward developing a mind-set that proactively scans the horizon for these threats, and learning the skills and approaches that help the team mitigate and address them as they occur. The more prepared a team is to identify and address these issues as they arise, the greater the likelihood that they can be dealt with before they significantly damage the team or the development effort. ●

JOHN RUFFA served as part of the in-house Goddard Space Flight Center development teams for the Rossi X-Ray Timing Explorer and the Wilkinson Microwave Anisotropy Probe. Most recently, he served as the mission systems engineer for the Solar Dynamics Observatory, which successfully launched from Cape Canaveral in February 2010.

