Managing the Invisible Aspects of High-Performance Teams

RADM Patrick Moneymaker
Ocean Systems Engineering Corp.

Dr. Lynn Carter
Software Engineering Institute

Many of the most important aspects of managing high-performance teams are counter-intuitive, yet we live in a culture that likes to reward its innovators and heroes even when clearly better solutions already exist. From new insights about metrics to the need to celebrate our successes, this paper describes a collection of five management lessons and provides useful references to sources for those needing implementation details.

If it were easy, everyone would be doing it. How many times have you heard this in the context of mastering a difficult task? The truth is, most challenges have a layer of complexity just beneath the surface that, once understood and applied, leads to the solution. The trick is to acquire a knack for seeing behind the everyday circumstances. The famous French aviation pioneer and philosopher, Antoine de Saint-Exuprey, had a key insight that applies: He contended that all that was most important in life was invisible. The invisible or counter-intuitive observations of management principals described here will hopefully provide a deeper understanding of difficulties previously experienced and some new ways to use traditional disciplines.

Thanks to a widely diverse career, I have a valuable perspective on two demanding yet diverse disciplines: military aviation and software development. As a Naval aviator I experienced the demands and thrills of flight training, carrier operations, combat, and ultimately duty as commander and flight leader of the U.S Naval Flight Demonstration Squadron, the Blue Angels. As a senior Navy officer and now as a senior executive of an engineering services firm, I have been responsible for software maintenance and development programs that ranged from the most strategic and complex to the most mundane.

During my aviation years I gradually developed into a seasoned pilot and tactician with each new level of skill supported by a solid foundation.

While involved in technology management I was thrust into a senior management role where I had little training and less experience.

Capability Maturity Model® and CMM® are registered in the U.S. Patent and Trademark Office as a registered trademark of Carnegie Mellon University.

Survival depended on applying my skills and instincts to the new setting. It was in this transitional setting – from maintaining situational awareness in air operations to maintaining effective software development program situational awareness – that I learned universal truths in management and team-building.

In each of the following insights, I will describe the software development context of my observation and my supporting aviation experience. Dr. Lynn Robert Carter, writing in Italics within the shaded boxes, will add references to others' works, including the Software Engineering Institute's (SEI) Capability Maturity Model® (CMM®). The two perspectives will hopefully add clarity to managing the "invisible."

Carter: Behind each of Saint-Exuprey's invisible truths is a wealth of already captured knowledge. I'm fond of quoting Dr. R. W. Hamming in his 1968 Turing Award lecture where he states:

Indeed, one of my major complaints about the computer field is that whereas Newton could say, "If I have seen a little farther than others it is because I have stood on the shoulders of giants." I am forced to say, "Today we stand on each other's feet." Perhaps the central problem we face in all of computer science is how we are to get to the situation where we build on top of the work of others rather than redoing so much of it in a trivially different way. Science is supposed to be cumulative, not an endless duplication of the same kind of things.

I believe that all human efforts should be built on the lessons of those who came before us. To ignore the contribution of these giants shows our ignorance and tends to doom us to avoidable pain. Following each insight, I will provide supporting material and references to assist you in obtaining each insight's full benefit.

Determine Metrics Parameters First

Understand what you want to monitor *before* you ask for metrics. While this seems obvious, all senior managers face the same handicap: There is just not enough time in the day to understand the complexity of *every* project. Far too often senior management makes a general call for metrics (focused on the usual suspects, cost, and schedule) because they do not know any better.

The project manager who is concentrating on the day-to-day execution details is forced to drop everything to develop snapshot graphs showing all is well for the moment. This ill-conceived request causes more harm than good. It diverts the project staff's resources and more often than not gives senior management a false sense of security. A better approach would be an honest one-on-one conversation to determine the senior managements' information gaps so the executives can fully understand the project's critical path, and the project manager can understand the perspective from above.

Once this shared understanding is achieved, and here comes the invisible part, more meaningful "first derivative" measurements can be crafted. What do I mean by first derivative metrics? Simply those samplings that give warning of movement in time to make a correction. In aviation language, it means flying using the vertical speed indicator (VSI) *as well as* the altimeter. The VSI needle moves well before the altimeter, so if the object is to maintain a certain altitude, look for the trend on that instrument first and correct as needed. Likewise, while it is nice to know how much fuel (resource) is in the tank, a fuel flow indicator gives the added insight of burn rate. The fuel flow indicator is the first derivative instrument used in conjunction with the quantity indicator. Together they provide the view just

below the surface data.

In summary, when the call for metrics is sounded, first work toward a shared understanding of critical path, and then agree on what is measured and how. Anything short of these steps is a waste of time.

Carter: In 1984 Victor R. Basili and David M. Weiss wrote a paper that introduced the Goal-Question-Measure Paradigm [1]. The heart of this approach is to start with the organizational goal that needs to be realized. From that goal, what are the questions that need to be answered in order to make corrections and steer to the goal? Given these questions, what measures need to be gathered in order to answer these questions, and how should this data be processed to obtain the needed situational awareness on which a meaningful decision can be made? Finally, given most people's overload, what mechanisms do we employ to ensure we take the time to examine the information and act when we should?

The first challenge in applying this paradigm is the lack of clarity and alignment in an organization's goals. Without this, determining the right questions can be quite difficult. Even when clarity and alignment exist, the second challenge is agreeing to what questions should be asked and how numeric data, if it can be trusted, would support the decision-making process. A solution to all of these technical challenges leaves yet another challenge – the people. How do we change a heroic culture of option-based reactive decision making to one that values and honors the benefit of data and uses the data wisely? The benefits of these measures can only be seen when people both understand and use them. This requires knowledge, skill, and wisdom at all levels in the organization.

For more information, Software Measures and the Capability Maturity Model *[2], may be a useful resource. Those needing more sophisticated measures see* Measuring the Software Process: Statistical Process Control for Software Process Improvement *[3]. Anything less tends to degenerate into something appropriate for a Dilbert® cartoon.*

®Dilbert is a registered trademark of United Features Syndicate, Inc.

Follow a Disciplined Path

Discipline enables creativity and innovation. In one of my debates with Carter on the effects of the CMM on my programming staff, I offered that all code reuse, libraries, and such might inhibit the programmer's creativity. I did not want to have a crew of demotivated code benders who bemoaned the yoke of so much attention to process. Carter offered that quite the contrary was true. As I gave it more thought I began to see examples in military aviation that conveyed the message.

Considering the dynamics of carrier aviation it was easy to see that every step in the process of conducting a combat mission was, in itself, a small module of discipline. Briefing the flight, pre-flighting the aircraft and ordnance, launching from the flight deck, rendezvousing with wingmen, in-flight refueling en route to the target – all of these steps and many more had been practiced until they had become second nature. This training allowed the pilot to relegate much of the "house-keeping" chores to the subconscious, freeing up his immediate consciousness to meet the unexpected. Once in the target area and after the long chain of disciplined steps, quick thinking, creativity and innovation was always needed. Any number of unforeseen events could complicate the briefed plan. After adjusting for changes, discipline would again be needed as the pilot rolled into a steep dive to begin the bombing run.

The previous quote from Sir Isaac Newton is applicable here as well. Just as he had used the efforts of those who had blazed the trail before him, likewise, discipline and ever-improved process give an aspiring master a lofty platform from which to launch his creative assault on the unknown. So, process discipline properly applied, can be a catalyst for monumental discovery. In a world of ever increasing complexity, the need for the leverage of process discipline has never been stronger, nor the opportunity for breakthrough greater.

Carter: It is easy to see how a senior leader and naval aviator could be concerned when first learning of the CMM and the additional actions it advocates. With no formal engineering background and no opportunity to learn how the benefits of using the CMM more than balance the costs, the admiral's reaction and comments were understandable. The challenge was to change his initial perception and begin the process of building an informed sponsor. At the heart of any improvement effort is a continuous effort of sponsorship development and sustainment [4].

Nearly every leader has some aspect of his background where special training and discipline was required. Evoking those experiences and relating them to process discipline has proven very useful to me. The most difficult part is helping them see the benefits that come from separating the routine from the novel. When we learn to see routine problems and develop proven methods for addressing them, we obtain more predictable results and free people to invest their creativity on the novel aspects of their work. While it was true that combat pilots could employ common sense to figure out how to brief each other before a mission, pre-flight the aircraft, etc., what is gained and what is lost? Wouldn't it be better to save their creative energies for the novel aspects of the mission? The problem with common sense is its inability to tell us when it is about to fail us.

Senge's The Fifth Discipline *[6] provides numerous examples of the importance of capturing and leveraging lessons from past experiences. He uses the phrase "the learning organization" to describe a mode of operating that recognizes that common sense and the education people tend to receive are inadequate for the kinds of work we are called on to perform these days.*

At the heart of The Capability Maturing Model *[6] is the notion of a process asset library that contains the organization's standard software process, software life cycles, process tailoring guidelines and criteria, historical data for selection and estimation purposes, as well as documentation for familiarization, training, and use by professionals. Not mentioned in this book is what kinds of processes to capture, what processes to ignore, and the need to consider different representations of these processes and other materials for the different roles to be played and the different purposes for which it will be used. This is not really an omission, since the answer for one organization is likely to be very different from another.*

From my perspective, an organization

must find a balance between allowing the bright and insightful professionals the freedom they need to do their work and leveraging the insight of those that have come before them. This balance protects the organization from painfully relearning lessons. As I tell my SEI Defining Software Process students, if there isn't a horror story or a pot of gold story behind a process asset, why burden the organization with it? There is no need to formalize common sense when it works just fine. The challenge for every leader is to become familiar with the horror and pot of gold stories and ensure everyone sees how the organization's disciplines are really only there to help us avoid the former and leverage the latter.

Stand on Firm Ground

High performance teams need strong infrastructure. You have a big, important project, and you have brought in the best and brightest to form a team. Should you expect harmony with all of this excellence? Will they be one big, mature, happy family? It is unlikely.

Even with the ideal Myers-Briggs® personality match up, the most painstaking selection process, and a tailor-made project for your team, the need for a rigid but porous environment cannot be overstated. Rigid because you cannot stretch to a goal on a shaky ladder. Porous because one person's good idea or word of caution, even whispered, needs to be heard by the whole team.

The Blue Angels handle this need for a communication-rich structure by employing an extremely thorough briefing and debriefing process around "the table." Representatives from all departments in the squadron meet before every flight.

All administrative, logistical, and operational details are discussed. If a member has a question it is resolved on the spot. It is here that pilots mentally rehearse each maneuver, study the overhead photography of the airfield, and put on their game face. Because they take the time to do this *every* flight, they can then disperse at a show site with its hundreds of thousands of spectators and feel confident in the plan. During the performance, every aspect from manning the aircraft to deplaning after the show is videotaped. This videotape is the debriefing focus, which is

conducted immediately after each show. These sessions are nothing short of exhaustive, but every detail is reviewed, and all comments are heard. So, it is a framework of structure and ritual that allows these top performers to be constantly "plugged in."

Within the software development domain, a similar construct is achievable. No, there does not have to be videotaped coding sessions, but the structure suggested by the CMM coupled with a little customized ritual to get people talking and in front of the issues can be very powerful.

Carter: A painful mistake we often make is assuming that everyone is like us. One place where this assumption can be most awkward is in decision making. The book Type Talk at Work *[7] provides a fascinating insight into 16 personality types, and how they influence success on the job. In particular, they describe a critical difference between how two different personality types – what the Myers-Briggs Type Indicator® calls "introverts" and "extraverts" – process information leading to a decision. Extraverts tend to process immediately and verbally, wanting to engage everyone in this process so a decision can be made. Introverts tend to process over time and quietly, wanting to think it through before speaking. Placing people with these two tendencies in the same room at the same time and asking them to make a decision often brings less than optimal results. Without care, a "decision" can be reached without the full benefit of all the team member's insights and experiences.*

Each Blue Angel demonstration begins and ends with time at the table. Before each demonstration, issues and concerns are raised and resolved. This is very similar to the Intergroup Coordination Key Process Area in the Capability Maturity Model *[6]. At the heart of these pre-project meetings are risk management activities. What concerns (risks) do we see? Which are likely to occur and warrant action? The better the group is at surfacing the risks, assessing their probability and impact, selecting which ones to address, and choosing mitigation or contingency actions, the better the project will be.*

After each flight, the Blue Angels reflect on what happened. From minor issues to surprises, the team tries to determine what

should be done differently to improve things the next time. A number of authors have written about taking time to reflect on what we have done as an integral part of becoming better, yet few people actually take the time to reflect. In Habit 7, Sharpen the Saw, Steven Covey [8] relates a story of an exhausted wood cutter so intent on cutting down a tree that he cannot see the benefit of taking time to rejuvenate and sharpen his dull saw. Donald Schön [9] helps us see two kinds of reflection: reflection-in-action and reflection-on-action. The former addresses adjustment professionals make in real-time based on the dynamic and unpredictable realities of our world. The latter is done postmortem in order to set the stage for a better performance next time.

The Peer Review Key Process Area [6] strives to reveal issues as early in the life of the project as possible. Studies have shown that the longer a defect remains in a system under development, the greater the cost to remove it. Leveraging the differing perspectives of the team can uncover the genesis of defects well before it becomes obvious to the professional doing the work. In fact, the notion of continuous peer review is described in the method called "pair programming" [10], where the notion that the right two people working as a pair can often produce more than twice as much as either working alone.

Halt the Hero Cycle

Heroes are often the result of someone else's mistake. When a pilot plans a flight, he takes a host of things into account: distance to destination, fuel required (some cushion if the destination weather is bad), winds aloft, terrain elevation, and on and on. Why? Because anything short of a successful landing at the intended location is unacceptable.

What if the pilot were handed a flight plan that did not offer him a fuel reserve or worse yet, not enough fuel to get to the destination? Would he even take off on such a foolish plan? Thankfully the laws of physics are broadly understood and protect the pilot who would protest.

In the business world, a proposal for software development may not have had the due diligence required to get safely to the destination. Or margins were shaved to arrive at a desired cost. Yet time and again programming staffs assemble for project flights with no weather forecast and questionable fuel reserves. And their superiors expect them to land at their destination on time.

The invisible truth here is that heroes end up filling the gap in resources by over-extending themselves and their teams. If they survive, they are rewarded with a promotion, and the cycle is perpetuated.

There is another subtle point to be made about the plight of the hero. The project manager under pressure to deliver will rarely listen to a good idea if it is outside his implementation plan. He has literally no room to maneuver, since taking time to listen to something potentially helpful takes time he does not have.

So, how is the cycle broken to let remedy enter? The answer is not hard to see if you look outside of the project management environment where there is some discretion. It takes commitment, resources, and patience.

My firm has the pleasure of assisting the Software Engineering Process Office, a part of the Space and Naval Warfare Systems Center, San Diego, Calif., which recently earned CMM Level 3 certification. The staff, led by Beth Gramoy, has enjoyed such commitment from Dr. Kolb in her front office and has offered on-sight training, pilot programs, and generally carried this corporate goal into reality. It is this kind of organic resource and backing that is required to change an organization's culture.

Carter: In his book, Principle-Centered Leadership, *Stephen Covey [11] points out that there are numerous systems that fail to yield to pressure. These systems proceed based on natural laws or governing principles, whether we know about and appreciate them or not. Most pilots and their leaders know the maximum speed and range of their aircraft, so few pilots have to contend with unreasonable task assignments with regard to these basic aspects. A small minority of developers works for executives who know their teams' productivity capabilities per code type. The majority of managers act as if the teams' productivity is*

something that can be negotiated and influenced by applying appropriate motivation and pressure. Yet none would consider negotiating with the crew of a Boeing 747 in an effort to push its maximum speed past the sound barrier.

Over and over, I hear leaders talk about achieving the SEI's CMM Level 3 as a critical business goal. When I ask them why they believe this is a critical goal, it becomes clear they do not really understand the principles on which the CMM was built or the natural laws it was designed to leverage. The main principle beneath the CMM is that continuous improvement and leveraging positive experiences help an organization avoid the black holes from which common sense is unable to protect it. Leaders striving to reach Level 3 without honoring this and other critical principles may obtain that level and may be allowed to compete, but neither the intent nor the real benefit of the model will be realized. The result is more work and paper with no real benefit, which damages the approach in the eyes of those who are forced to operate in such an environment.

Talking with the people who work with Gramoy is a wonderful experience. Here is a fine example of people who have taken the time to understand the hidden factors and discover the basic principles and natural laws. They are embarrassed by the praise, for the more they learn, the more they realize that they need to learn. Such professionals and leaders are a joy to find.

Enjoy the Success

Software engineers must learn to celebrate. Hidden behind every space shuttle launch is a grueling schedule that begins when the landing wheel stops rolling from the previous mission and ends when the launch button is depressed to start the next mission. In the intervening months between shuttle launches, every component and sub-component of the Space Transport System is checked and tracked. There are literally millions of details that come together at launch time. Then after all the painstaking attention to detail, all the diligence, all of the mind-numbing rehearsals, there is a celebration of the human spirit like no other on the planet.

While attending a shuttle launch I happened into a conversation with the bus driver who was taking us to the viewing

site. All along the trip he would spout facts and trivia about NASA and the shuttle program. His enthusiasm was so contagious that the entire bus was lifted to a higher level of excitement and anticipation. I learned that he was a NASA engineer who had retired but still enjoyed the thrill of the launch. He embodied true passion and delayed gratification. He never once mentioned the tedious preparations for launch, the personal sacrifice, and long hours. However, he gushed with pride and the sense of accomplishment.

When I was flying with the Blue Angels, I was routinely struck by the audience's general lack of understanding of what it took to put on a demonstration. Some thought we improvised our maneuvers on the day of the show. Some thought the planes were somehow mechanically locked together in the diamond formation. Very few realized that every maneuver was the result of months of intense practice and that each maneuver fit together precisely to form the entire demonstration.

Often there would be an aircraft "gripe" that would require nationwide logistical coordination for the delivery of a part. The part would be delivered and installed, and the aircraft test flown before the next day's show. And yet, time after time, the show would go on. The maintenance crew proved to be magicians on a regular basis, pulling the rabbit out of the hat just before show time. Then the diamond formation would thunder into the air, eight engines in afterburner, rising majestically into the vertical and complete a loop on take off. Each member of the maintenance crew would follow the formation skyward and experience a penetrating satisfaction. All the pain was erased as each member of the team drank in the beauty of what they had helped achieve. It was a celebration, brief and pure.

Where there is no excitement at completion there is no passion. Conversely, where there is a cause worth celebrating, the passion is inexhaustible. But how can a project manager recreate a shuttle launch or a flight demonstration? The reality is (the invisible part) that within *all* human enterprise there are the seeds of satisfaction and accomplishment waiting to be nourished by a manager who is also a leader, one who can connect human

endeavor with purpose. When a team can boast of delivering code on time and within cost, time after time, it will truly differentiate itself in the marketplace. That alone is worth a celebration.

Carter: The beginning and the end are the most critical points. How these are handled are excellent predictors as to what will happen in between. One speaker I heard made an interesting observation that almost every good book on project management recommends including a project postmortem to capture lessons learned. He goes on to ask why those same books do not advise starting each project by reading lessons learned.

Acknowledging our humaness and employing proven methods to obtain our best is a critical job of every leader. The Blue Angels have a lot to teach us about how to build and leverage a high-performance team. It starts by finding precisely the right people to be on the team. This is accomplished by involving those from last year's team who will continue on into this year in the selection process. It continues with team-building rituals and practices. Not the meaningless waste of time we have come to abhor from ill-prepared facilitation "professionals," but activities specifically designed and used to build team respect and trust.

One of the best sources of materials available for building generic high-performance teams can be found in the Sibbet and Drexler Graphic Guide to Team Performance [12]. While not every item in this guide will be appropriate for every team, it provides a wealth of tools and just enough context to help you understand why you might need to use one.

From Newton, we see the reverence for those mathematicians who came before him and his belief that they helped him reach new heights. From Saint-Exuprey, we learn that the most important things in life are seldom easily seen. From the admiral, I have learned that there is more to be gained by leveraging the advances and accomplishments of others even when their domain appears to be unrelated. I believe there is much more the Blue Angels have to teach us all, if we will only take the time to see beyond the obvious to the invisible, and honor the lessons of those giants who have been this way before us.

Conclusion

At the end of the day, we will always find more goals than resources to support them, more data than understanding, more complexity than order. This is the nature of growth. The key is to keep one's bearing amidst continuous change, and always be respectful of the lessons of those who have gone before us. Hopefully the insights we have shared have pierced the veneer of visible reality and offered a view of important underlying issues that challenge each of us. Effective metrics needs a shared understanding, creativity needs discipline, and strong cultural foundations stabilize high-performance teaming. We need to focus on the systemic ills that surround our heroes and find the cause within our task worthy of excitement. ◆

References

1. Basili, Victor R., and Weiss, David M., A Methodology for Collecting Valid Software Engineering Data, IEEE Transactions on Software Engineering SE-10, Nov. 6, 1984, pp. 728-738.
2. Baumert, John H, and McWhinney, Mark S, *Software Measures and the Capability Maturity Model*, CMU/SEI 92-TR-25, Pittsburgh, Pa. Software Engineering Institute, Carnegie Mellon University, September 1992.
3. Florac, William, and Carleton, Anita *Measuring the Software Process: Statistical Process Control for Software Process Improvement*, (ISBN: 0-201-60444-2) Addison-Wesley Computer and Engineering Publishing Group, 1999.
4. Gremba, Jennifer, and Myers, Chuck, *The IDEAL^SM Model: A Practical Guide for Improvement*, 1997, www.sei.cmu.edu/ideal/ideal.bridge.html
5. Senge, Peter M., *The Fifth Discipline*,(ISBN: 0-385-26094-6) Bantam Doubleday Dell Publishing Group, 1990.
6. Paulk, Mark C., et. al., *The Capability Maturity Model*, (ISBN 0201-54664-7) Addison-Wesley Publishing Company, 1994.
7. Thuesen, Janet M. and Kroeger, Otto, *Type Talk at Work*, (ISBN 0-440-50699-9) Bantam Doubleday Dell Publishing Group, 1992.
8. Covey, Stephen R., *The 7 Habits of Highly Effective People*, (ISBN: 0-671-70863-5) A Fireside Book, Simon and Schuster, 1989.
9. Schön, Donald A., *The Reflective Practitioner*, (ISBN: 0-465-06878-2) Basic Books, Perseus Books Group, 1983.
10. Williams, Laurie A., and Kessler Robert R., All I Really Needed to Know about Pair Programming I Learned in Kindergarten, *CACM*, May 2000, pp. 108-114.
11. Covey, Stephen R., *Principle Centered Leadership*, (ISBN: 0-671-79280-6) A Fireside Book, Simon and Schuster, 1990.
12. Sibbet, David, and Drexler, Allan, *Graphic Guide to Team Performance* (ISBN: 1-879502-09-7) The Grove Consultants International, 1994.

About the Authors

Lynn Robert Carter, Ph.D., is a senior member of the technical staff at Carnegie Mellon University (CMU) in a dual appointment at the Software Engineering Institute and the School of Computer Science. The focus of his nearly 12 years at CMU has been on assisting technical workers, managers, and leaders in benefiting from adopting software technologies in the context of organizational overload. Previously, Carter worked for firms including Tektronix, Motorola, and a leveraged buy-out from GenRad.

Software Engineering Institute
Carnegie Mellon University
3857 East Equestrian Trail
Phoenix, AZ 85044-3008
Voice: 480-598-1247
E-mail: lrc@sei.cmu.edu

RADM Patrick Moneymaker USN (Ret.) is president of Ocean Systems Engineering Corporation overseeing full-spectrum engineering services that the firm provides for the Department of Defense, including software development, C4I systems integration, and program management. Moneymaker retired October 1998 from the Navy as rear admiral having held information technology positions that included J6, US Strategic Command and Commander, and Naval Space Command.

RADM Patrick Moneymaker USN (Ret.)
OSEC
3142 W. Vista Way
Oceanside, CA 92064
Voice: 760-722-3222
E-mail: pmoneym@osec.com