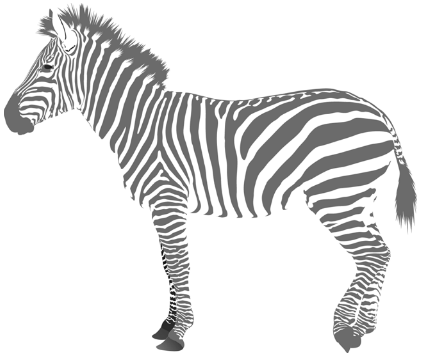
Beautiful Teams



Edited by Andrew Stellman and Jennifer Greene

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Taipei • Tokyo

CONTENTS

CHAPTER THIRTEEN

Defending the Free World

An Interview with Neil Siegel

Few jobs in the software industry are bigger than the chief engineer of Northrop Grumman's 35,000-person Information Systems sector, one of the largest defense and technology organizations in the world. Not many people (ourselves included) outside of the defense industry know much about the world of defense and military projects. Neil Siegel gives us a rare look at the defense industry, and what motivates the people who work in it.

Andrew: Start off by telling me a little bit about Northrop: how you came to be there, and how you came to learn about how teams work.

Neil: Northrop Grumman is one of the key participants in the U.S. defense and aerospace industry, with many important contributions over the years. The company took its current form over the last 10 years through a series of mergers, bringing together a set of synergistic companies with long and proud legacies of accomplishment. We think of ourselves as a “company of immigrants,” respecting those legacies and achievements, while at the same time trying to combine our skills into something synergistic that is better than ever. My current job is sector vice president for Technology and Advanced Systems; I’ve been doing that for about seven years. Before that, I ran one of the businesses within the company, called Tactical Systems. Tactical Systems created vital, life-saving capabilities for the U.S. military—what the military calls “battle command” and “command-and-control” systems, primarily for the Army, but also for the other military services. Tactical Systems was a very successful business organization, which grew on average 25% a year during each of the seven years that I ran it.

Andrew: And what kind of projects did you work on?

Neil: We built equipment that allowed our soldiers to communicate and coordinate on the battlefield, to know where everyone (friend and enemy) was, to make and share plans, and to monitor the execution of those plans. It was a very team-oriented business; 50% of our revenue was derived from vendors and subcontracts, which meant that we had not only to build effective teams inside the company and with our customers, but also with a large number of other companies. Interestingly, the typical contract inside the company at the time had 70% or 80% of the content done by our own company, so this was an organization that depended very heavily on its ability to form teams not just with other parts of our own company but with other companies. And that was a lot of fun, and yet at the same time, very hard.

Andrew: What made it hard? What’s harder about working with teams from a bunch of different companies?

Neil: It’s about achieving alignment of objectives. Everybody has their own set of motivations and goals, and it’s very rare, even inside the company, for two different people or two different organizations to have exactly the same motivations and goals. But this potential for misalignment of motivations and goals is even more exaggerated when you are working with people from other companies. For example, you might team with companies that would actually prefer that they were in charge of the job; for various reasons, you are able to talk them into being your subcontractor to team together for a job, but maybe there’s the thought in their minds through the whole job that they could have just gone off and won this by themselves…and then they would be the prime contractor! In any case, there are always factors like, “We want to have a little more credit”, or “We want to have a little more visibility”, or “We want to have a little more work”, or “Why is she giving the work to this other party? We could do that better.”

So, all those things that are perfectly normal even inside one company are all magnified a little bit when you are partnering with other companies, because every work assignment has to be done as a subcontract, or as a teaming agreement—every aspect of the relationship is controlled by a very formal business arrangement. Every issue is resolved through formal negotiation, and the resolution documented through modifications to the subcontract. Actually, in some sense, there's an advantage inherent to using outside partners, because it forces you into that formality, which gives you the possibility of eliminating ambiguity in what you've agreed to, and ambiguity can be a real enemy to effective team operation. Ambiguity is a major source of conflict, due to mismatch of expectations.

Andrew: That sounds like a delicate balancing act. I get the sense that you had to learn that the hard way. Not a lot of people get the opportunity in their careers to learn a lesson like that. How did you learn that?

Neil: Let me go back a little bit in time to when I was running the Tactical Systems division. We bid and won a large Army program, and we produced a very important warfighting system (one that is still in use); I think at the time we submitted the proposal we had about 10 companies on our team. Some of these companies were providing what you might call components: they were very specialized, they had very clear scope, and as a result there was little conflict, little ambiguity about how we would work together on this job. But others of those companies viewed themselves as generalists who could bring a full range of capabilities to the partnership, and yet to be a part of a team that large, they had to narrow their scope to do just one or two specific things. As a result, some of those companies came to the party with the feeling that they'd subverted their ambitions so as to increase their likelihood of a win, and while that made sense intellectually, emotionally many of them felt like there were additional pieces of work on the job that they could do and would like to do. So, there was really a possibility for misalignment, and even conflict.

It was a very exciting activity—first of all, to talk all these companies into being your subcontractor rather than submitting prime contractor bids competing against you, or teaming with some other bidder. You basically have to convince them that their total expected financial and strategic return is higher because the probability of win is higher. Maybe their work share—the dollar value of work they'll do—is smaller, but the probability of win is so much higher that the total financial package is better for them. That's a hard sell, at times.

And then, when you have so many companies on one team—and in this instance, for various reasons, that was really necessary—then obviously none of these companies could be doing a very large portion of the total job; they start asking, "OK, I'm good at that, but I'm also good at these other things, so why aren't I doing that? Why have you given that work to this other company?" So, it's a real negotiation challenge; you have to get everybody aligned and working toward the main goal (and continuously keep them aligned!), and then you have to get everybody motivated, because this team might have 10 companies on it, but it has to look to the customer as one team.

Andrew: I've never had to deal with a project team as complex as one with people from 10 different companies, but I definitely recognize the need to get people aligned and motivated. How do you do that?

Neil: In our business, we have some advantages and we have some disadvantages. The first thing that we do in the defense business to get people aligned and motivated is sell our mission: we are defending the free world. I was the general manager of this business throughout the dot-com boom. The dot-com boom created a huge demand for technical talent. Recruiting talented software people for our company became a big issue because the dot-com boom was sucking up a lot of talent and paying very high salaries. And so I used that opportunity to recruit shamelessly, based on the value of our customers' mission. I would say things like, "You're a talented software person. You could go to work for a telecom company, and your life's work could be routing telephone calls successfully. Or you can come to work for us and maybe we won't pay you quite as much—although none of our employees starve—but you'll be saving the free world."

Andrew: I can see how that would be a motivator—not for everyone, but definitely for the kind of person you'd want working there. Did it take a lot of work to find those people?

Neil: It's kind of a self-selection process. The people who resonated with that motivation came to work for me. The people who were absolutely out to get the best salary offer didn't come to work for me. I had a strategy of trying to pay just a few percent less than the prevailing wage in our area so that I would be picking off those people who resonated with our customers' mission, playing to one of the natural strengths our industry has: we do something that's important and easy to explain. We also do something that's interesting. I view it as every bit as interesting as the telecom companies or Microsoft—that's a personal judgment—but it's most definitely interesting. It's high-tech, very leading-edge. So, I could really play on the combination of *important*, *interesting*, and *reasonably compensated*.

So, the people who joined my team had self-selected against those criteria. This is important: the first vector of alignment wasn't around our company, and wasn't around me, but was around "here's a way to contribute to national defense." This particular program had very high visibility in the Army—we routinely met with the chief of staff of the Army, and so forth; this made working on this program a very visible opportunity to do something that was clearly important. Furthermore, the Army was trying to do this reasonably fast. When you're trying to do something big in the defense industry, sometimes it takes seven years, or 10 years, or longer; it takes seven years to build an aircraft carrier. But the government wanted to do this particular innovation much faster than normal. So, not only could you do something interesting and important, but you could also get to the job satisfaction of seeing the thing done in about three years.

That gave me an opportunity to create alignment around these bigger-than-us factors like national defense, and the United States Army, and the first-ever digitization of the battlefield, and saving soldiers' lives, and saving marines' lives. That provided a strong message around which to start a process of reaching alignment within our team. I suspect that this might not translate so directly to other fields, but with a little bit of creativity, almost every business has something about their business that is essential and important in some way.

Andrew: I'm actually a little surprised by the fact that what you just said is very similar to what I've heard from open source people: a feeling of working toward a higher cause. Your higher cause is life-and-death, but for a lot of us, I suspect we can still find that kind of motivation in our own projects. What do you think?

Neil: National defense is a very tangible thing, but I suspect that many fields have some aspect that can be important. For example, working for the electric company can be important, because delivering uninterrupted, reliable electric service is important. One can create an ethos within your team that what they are doing is vital to the community. There's probably something in every industry that you can attack from that angle.

Andrew: So, let's get back to those 10 companies you had to balance. How did you do it? I can see how you'd motivate your individual team members, but how do you keep all of these companies working toward the same goal?

Neil: There are some obvious things that they want: they want revenue, they want profit margin, and they want predictability. But those are really, in my experience, weak things to build alignment around. The better thing is, "What are the strategic goals of that company?" If you're dealing with people at a sufficiently high level in your partner companies, you can have discussions about what the strategic goals are and what you can do—things that may or may not have to do with near-term revenue—that can maybe help reinforce a capability that five years from now will be strategically important to them.

So, one spends a lot of time trying to understand the strategic goals of your partner companies. It's a little difficult, in that they're not going to tell you everything, because you're still potential competitors on other jobs. But just ask what makes them tick, what their strategic goals are, and what you can do that will not only satisfy their near-term need of revenue and jobs, but will also create strategic value out of the relationship. Even just the act of asking such a question and listening to their answer, engaging in dialog, is creating an air of respect: a sense that this project isn't just about me and my glory, or maximizing the glory and revenue of my company, but includes them, too. And that's obviously a long, slow process—one conversation is never a basis for building a relationship of trust—but it's a very tangible thing. You can find mutually agreeable things in a teaming relationship that create strategic value for the partners.

And that works for partners, and also with other organizations within our own company. The nature of the discussions with organizations inside our own company is maybe a little different—there's actually a perverse clarity that comes with having these discussions with other companies—but if you look like you're trying to reach beyond the obvious work scope and revenue, and trying to make this work for them in a longer-term strategic sense, that can create a lot of satisfaction in the teaming arrangement, even if the immediate revenue is less than that for which they hoped.

Andrew: I'm really interested in the day-to-day of what it's like working on one of your projects. I assume that you guys have to go through a pretty big information-gathering phase at the beginning of a project, talking to the guys who are actually using what you'll be working on. Can you tell me a little bit about how that works?

Neil: First of all, this information gathering is a huge opportunity for realizing the alignment that I talked about first, the alignment around the mission. Because really, this is a chance to get people seduced and resonating emotionally with the mission of the users. The officers in the U.S. military are, in general, extremely capable and motivated people who are great representatives of their mission. It's wonderful for me to be able to connect our people with them.

Andrew: Are they good to work with? What's it like working with them?

Neil: They're a good customer! They believe in their mission, and know how to talk about that. Of course, the U.S. military is an even bigger bureaucracy than Northrop Grumman; we're a hundred thousand people, they're a million people, and there are some potential disadvantages with that. On the other hand, they understand that the quality of what *we* do can affect *their* life: in a literal life-and-death sense, not just in a sense of whether their milk is fresh or stale, or whether their newspapers arrive on time; their very lives can depend on the quality of our work. So, they're very motivated to make sure that they convey information to us as effectively as possible: what they do, how they do it, what they want to be able to do. It provides great clarity. So, we love for our people to get deeply immersed in our customers' problem. We call that "acquiring domain knowledge."

We like our people, especially our more senior technical people, to become real experts in one of our customers' specific problem domains. When they acquire that expertise, we now have the engineering knowledge, the technology knowledge, and the domain knowledge all resident in one brain…and in my experience, that's when the magic happens. Then we can create these great ideas that in turn get the customers very excited that we can really do something that will improve things for them.

Andrew: "Where the magic happens"—can we delve into that a little bit?

Neil: I have an opinion that, in the typical business—and I've done some commercial IT and other things in my career—they all kind of look the same. There are people who work for our customers who are designated to talk to the contractors, to describe what they need and how they work. Sometimes customers are their own worst enemies. They probably understand quite well what they do today, but sometimes their vision of tomorrow is three seconds from now, not three years. They can have a very narrow view of what they need.

So, we do things like sending our senior employees to spend a year with the National Security Agency or the National Reconnaissance Office, or we bring Army officers to live in our facilities for a year, or we send engineers out to observe maneuvers at the National Training Center, or even send engineers to Iraq and Afghanistan to ride along with units on patrol. Because typically the customers' designated experts, who may understand their current practices, and may or may not have a sufficiently long-sighted vision of the future, probably don't have a grasp of the technology…and therefore don't understand the art of the possible. On the other hand, the private industry people have technical experts who understand a lot about technology, but if they don't understand the problem domain, you need an interpreter in the middle: "He said this, but what he really means is this," and so forth. We used to really concentrate on developing these interpreter–translators. But we realized it takes a long time to turn domain experts into engineers or scientists; we hire a modest number of retired military people because they can come in and tell us about how the organization works. But at the same time, what I've found is that what works better is to immerse our engineers into a customer's problem domain to acquire domain knowledge. Then I don't need a translator; there's no barrier between domain knowledge and technical skills. One person can merge both, develop new ideas, and help visualize the future.

So, that's what we aspire to: getting our technical staff to acquire a pretty good level of domain knowledge about a customer. Now there's no filter or barrier between the domain knowledge and technical knowledge, and that's where the "1 + 1 = 3" ideas for seeing what the future could be occur. That's how you get out of the "future is three minutes from now" bottleneck—you start thinking about how you can use technology to create revolutionary improvements in this mission. That's how we work with customers.

Andrew: Are there any specific practices or tools that come to mind that help you deal with this?

Neil: We have lots of specific mechanisms. For example, we have something called "user juries" where we take ideas out, and walk actual military users through how a system would work: a day in your life after we built this new system. Then, we discuss whether that will help them, and how we would measure such improvement; we try to measure projects not in terms of technical improvement, but in terms of operational utility. There are lots of other mechanisms; it's all about trying to eliminate or minimize the effects of the barrier between domain knowledge and science and engineering.

Andrew: I would love to hear a story from an actual project where you could tell us a little bit about what you were building, and how you actually applied these mechanisms.

Neil: Sure. I'll give you an example. On one of these military command-and-control systems we were building, we had an interesting conceptual breakthrough.

Andrew: I'm not familiar with military systems—what's a command-and-control system?

Neil: An information system for military commanders. It allows them to communicate with each other, and allows them to maintain situational awareness: where everybody is, what their status is. It's how a commander can understand what's going on. Battlefields used to be small areas—a commander could stand on a hill with a pair of binoculars and could see what was going on. But by about 1900, the battlefield started getting so big that you couldn't stand on the hill like Napoleon did. You had to start using technology to "see" the entire battlefield. And the systems that do that are called "command-and-control systems."

A soldier in the field using a command-and-control system built by Neil's team

So, we were building a system like that. And we had in our head a model that was kind of like email: I've got a piece of information and want to send it to somebody, so I get the information into the system, and then I tell the computer the name of the person to whom I want to send it. One of the big "aha!" realizations we had is that that was actually a very low-value way to build one of these military information systems; the users eventually realized that actually they don't want to have to know whom (by name) is the right person to receive a piece of information: they wanted the system to figure out from the intrinsic nature of the information who are the people who would be interested in receiving that information, and then to send it to them automatically. If you're on the battlefield and you see something, and determine it's a potential target—for example, the Army and the Marines use a process called "call for fire," saying, "I want to tell somebody to shoot that tank for me"—maybe I'm just an observer and don't have a weapon or don't have the right kind of weapon or I need to stay concealed, so I can't shoot it, but I want to tell somebody to do so. But I don't know who that should be. So, let the computer figure that out, and automatically get the information to the right person.

So, we finally realized that they want to address a call for fire to the *system* (and *not* to a specific named person), and have the system figure out who they should ask to shoot it. Because there are a lot of different options: you could have the Air Force drop a bomb from an airplane, or you could have a helicopter fire a little missile, or you could have another tank shoot it, or you could have artillery stand 10 miles away and shoot shells at it. And you don't want this person on the battlefield to have to decide which mechanism to use. He might not make the right decision: he doesn't know who's in the area, who's ready to shoot, which kind of ammunition to use. His job is to recognize that there's a bad guy out there, and that we should shoot him. It's somebody else's job to decide what kind of ammunition to shoot, and what kind of delivery mechanism to use.

It took an amazingly long time for us to communicate with each other, and to realize that the "right" approach was for the system to figure out who would be interested, rather than having the originator of the information have to address the information to specific people by name. That's an example of the tension and the suboptimization that can happen when the user is speaking in one language and the scientists and engineers are speaking in another, and how the processes of getting engineers to become domain experts and having user juries and so forth can get you to have the "aha!" moment of "Oh! This is how I can do what you really need!" The users hadn't thought about it that clearly, either, and so we were all struggling with this useless idea of how to build the system.

Andrew: I love that story. One thing that's interesting to me is that a lot of people think of military contracting and associate it with bureaucracy and paperwork, but the way you describe it makes it sound really tough but really interesting.

Neil: I wouldn't trade my job for any other job on the planet. I'm the chief engineer of a 35,000-person organization that does really important and really interesting things. This is a wonderful business for an engineer or a software person. Yes, there's a bit of bureaucracy inside the company because we're 100,000 people, but on the other hand, you can have 99,999 really smart colleagues. If you can harness that size to your advantage, it can be great.

I've worked in small companies. It's a lot of fun, and it's really easy—there's 20 of you. But if you needed an expert in some other field, chances are there isn't one. Here, there's an expert in almost any field that's relevant to what we do. Since I'm the chief technology officer for a big chunk of the company, either I know who these people are, or I know who to call among my counterparts in the other portion of the company to find them. There are thousands of world-class experts inside this company and, if we can get organized, that size works to our advantage. So, when we are at our best, this is a very exciting and wonderful business. It has its bad days, like all businesses, but this is a great and exciting business for software people. And we have a completely insatiable demand for software people and system engineers, with the one caveat that they almost always have to be U.S. citizens.

Knowledge Workers, Techies & Nerds: The Secrets and Science of Hiring Technical People (Dorset House), and *Corrective Action for the Software Industry* (with Denise Robitaille; Paton Press).

Neil G. Siegel is vice president and chief engineer at Northrop Grumman's Information Systems sector. He has been responsible for many successful fielded military and intelligence systems, including the Army's Blue-Force Tracking system and the Hunter unmanned aerial vehicle. Programs for which he was responsible have won numerous awards, including the Crosstalk Award in 2001 for one of the five best software projects across the entire U.S. government, and the 2003 Monticello Award (given in recognition of an information system that has a direct, meaningful impact on human lives). His inventions have led to billions of dollars worth of new business for the company, and more than a dozen patents. Among many other honors, he was elected to the U.S. National Academy of Engineering in 2005.

Tom Tarka worked at MP3.Com for just over two years, and the experience—the people, the place, the work—left an indelible mark on the person he became. Upon his departure, he picked up his travels where he had left off and spent the better part of a year roaming the United States in a 1971 VW Camper, eventually landing back on the East Coast.

After a brief stint as a mechanic, someone took a chance on Tom and his dusty chemical engineering degree and gave him a job in a field that had intrigued him since childhood: energy. He now works for the U.S. Department of Energy on next-generation power systems and global climate change and is a licensed professional engineer.

Tom resides in Pittsburgh, where he owns a home and a small collection of VW buses.

Tony Visconti's career as a record producer spans four decades. He is the producer of 12 David Bowie albums, 10 T. Rex albums, and many, many more by British recording artists. In the past two years, Tony has worked with Morrissey, Razorlight, Alejandro Escovedo, Anti-Flag, and Fall Out Boy. He has written his autobiography for Harper Collins, titled *Bowie, Bolan and the Brooklyn Boy*.

Karl Wiegers is principal consultant with Process Impact, a software process consulting and education company in Portland, Oregon, where he specializes in requirements engineering, process improvement, and project management. Karl is the author of the books *Software Requirements* (Microsoft Press), *More About Software Requirements* (Microsoft Press), *Peer Reviews in Software* (Addison-Wesley), *Practical Project Initiation* (Microsoft Press), and *Creating a Software Engineering Culture* (Dorset House), as well as numerous articles on many aspects of software, chemistry, and military history. Karl has a Ph.D. in organic chemistry, which he regards as the perfect background for a career in software development. You can reach him at *http://www.processimpact.com*.