

Leading Complex Projects in the DoD

Steven R. Meier, Ph.D.

A large, stylized blue letter 'M' graphic that serves as a background for the main text paragraph.

Many of today's projects require novel approaches to handle increased complexity and large uncertainty. Complex projects are both difficult and challenging even for the most seasoned project managers. Leading these types of projects requires a versatile skill set, the ability to manage the unforeseen, and a strategic vision. Complex projects require more than just management; they require leadership.

Meier, Ph.D., Program Management Professional (PMP), has more than 20 years of federal and private industry experience focused on the defense, intelligence, and civil aerospace communities. Meier is the founder of SRM Consulting, LLC, a consulting firm that specializes in linking strategy and execution to business results. He was a vice president at the Lockheed Martin Corp., and a member of the federal Senior Executive Service at NASA.

Leadership is important in every project but can be even more challenging for complex projects since there is a multitude of variables to manage all at once. Complex projects lie between traditional project management and extreme project management and they:

- Utilize new or unproven technology.
- Consist of independent, interacting elements that require integration.
- Involve two or more stakeholders.
- Entail a dynamic human resource environment.

These traits are common to many Department of Defense projects. First, most DoD projects have a goal of demonstrating unproven technology to meet the increasing needs of the warfighter or to address a new threat. Second, in most cases, DoD projects involve the designing, building, and delivery of a system or subsystem that fits into a larger architecture and requires integration at multiple levels. Third, in these times of shrinking budgets and affordability, many programs have adopted cost-sharing partnerships with other agencies to ease the financial burden. And fourth, many DoD organizations involved in complex project developments have military and civilian personnel who rotate every 2 to 3 years, creating a dynamic human resource environment.

Project Leadership Best Practices

The purpose of this article is to: (1) add to the existing knowledge base of best project management leadership practices, (2) confirm the results of other publications and studies on complex DoD projects, (3) provide seven practices for leading complex projects, and (4) discuss the causes of unsuccessful complex DoD projects.

Specifically, this article identifies seven leadership practices that have been utilized to lead complex ground, air, and space projects to successful outcomes. They include:

- Be decisive.
- Battle overzealous advocates.
- Mature new technology early and in a serial process.
- Experiment early and fail early.
- Stop requirements creep.
- Take great care in managing interfaces.
- Create a software integrated product team.

In the remaining sections of this article, I will discuss each of these best practices in detail and provide data to support each practice.

Be Decisive

One of the most critical roles of a DoD project leader is to make decisions. To a large degree, the success of any project comes down to project personnel making good decisions on a daily basis. Many leaders are reticent to make timely decisions for fear of making the wrong decision, or require additional studies to provide more data to execute a decision. The

inability to make timely decisions contrasts to past and current leaders who were well aware of the critical need to make timely decisions. To quote President Theodore Roosevelt, "In any moment of decision, the best thing you can do is the right thing, the next best thing is the wrong thing, and the worst thing you can do is nothing." Moreover, John Chambers, the long-standing and admired CEO of CISCO, echoes Roosevelt's sentiments with, "Without exception, all of my biggest mistakes occurred because I moved too slowly."

Decisions need to be made in a timely manner to keep a program moving forward and to maintain high motivation levels for the project team. Being decisive does not refer to making haphazard, uninformed decisions but making decisions that are based on data, facts, and experience. Since most defense projects are demonstrating new technologies to provide new capabilities or enhance existing capabilities, there are many variables to juggle such as cost, schedule, technology maturity, requirements, contracts, and staffing. When it comes to decisions that involve assessing several random variables at once, psychological studies have shown that the human brain has difficulty thinking forward with any accuracy. Moreover, these papers provide evidence that the most simplistic statistical models are more accurate than human predictions. Based on this information, project leaders should seek and use data to create simple charts such as a comparison table, histogram, Pareto chart, or scatter plot. These charts will enable the team to view and analyze data and perform a sensitivity analysis to understand how changing one variable affects the other variables. This approach will allow project leaders to predict future project trends and understand how project variables interact.

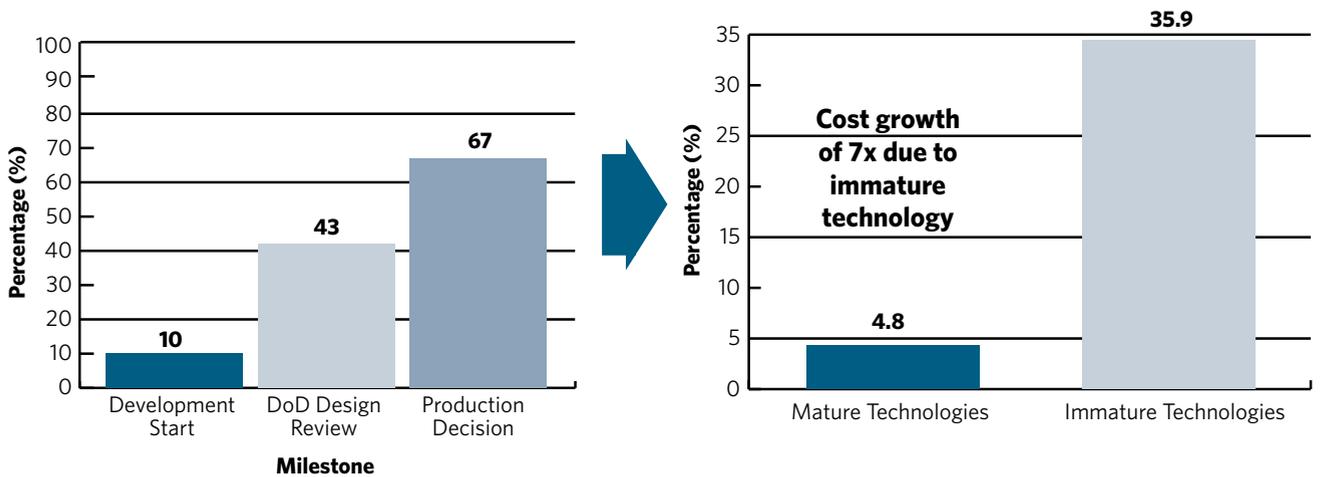
The cost and schedule impacts of delaying a decision can be severe. For example, a complex major defense acquisition program may have 2,000 full-time equivalent (FTE) personnel employed on a contract including government personnel, prime contractors, subcontractors, vendors, and suppliers. Assuming a yearly cost of \$400,000 per FTE, this amounts to an approximate cost of \$16 million per week on the contract. Now if work is stopped for 2 weeks while a decision is pending or being adjudicated, the impact will be a sunk cost of \$32 million and a schedule slip of 2 weeks to the project.

In summary, project leaders of complex DoD projects must make timely decisions to keep the progress moving forward and to maintain high motivation levels. Project leaders also should utilize data and implement simple quantitative techniques and models to understand sensitivities and interactions among several project variables.

Battle Overzealous Advocates

Overzealous advocates are overly enthusiastic individuals who overpromise and underdeliver on projects. While project advocates can have a positive impact, overzealous advocates promise extraordinary capabilities at a fraction of the actual cost and schedule. These advocates can be extremely detrimental to the long-term prospects of a complex project

Figure 1. Program Cost Overruns from Immature Technology



Percentage of programs that achieved critical technology maturity levels at key milestones from GAO-06-391 (2006) from data on 52 programs.

Average program research, development, test, and evaluation cost growth from first full estimate from GAO-06-391 (2006) from data on 52 programs.

since they develop overly optimistic project cost, schedule, and performance baselines. Overzealous advocates can be senior leaders in government who want to gain positive political light, senior leaders in private industry looking to win a large contract that will produce a long-term revenue stream, and government program managers looking to be promoted to senior government or military ranks.

In summary, the best methods to battle overzealous advocates on DoD projects is to ensure team continuity and accountability; identify and document all risks early in the project life cycle; conduct independent review prior to Milestone B; and develop rigorous entrance and exit criteria at Milestone B and other key design points.

Even in the face of contrary facts, overzealous advocates will trend to optimistic outcomes instead of realism. Data to support this viewpoint are presented in a March 2008 article I authored on best project management and system engineering practices for large-scale federal acquisition programs. A few comments from that paper include: "The program suffered from excess optimism," "Frequent turnover makes it hard to establish accountability," "Decision makers need to reexamine decisions as new information is disclosed," and "the prime contractor should not fear retribution for bearing bad news." All these data beg the question: What can be done to battle overzealous advocacy? Here are a few steps that may help:

Mature Technology Early and in a Serial Process

Numerous Government Accountability Office (GAO) reports detail how many major DoD acquisition programs began the program execution phase without verifying that critical project technologies had reached the proper maturity level. As shown in Figure 1, data collected from 52 DoD programs clearly provide evidence that not maturing technology early in the program life cycle had a factor of 7 cost growth compared to programs that had matured critical technologies at the appropriate design milestone.

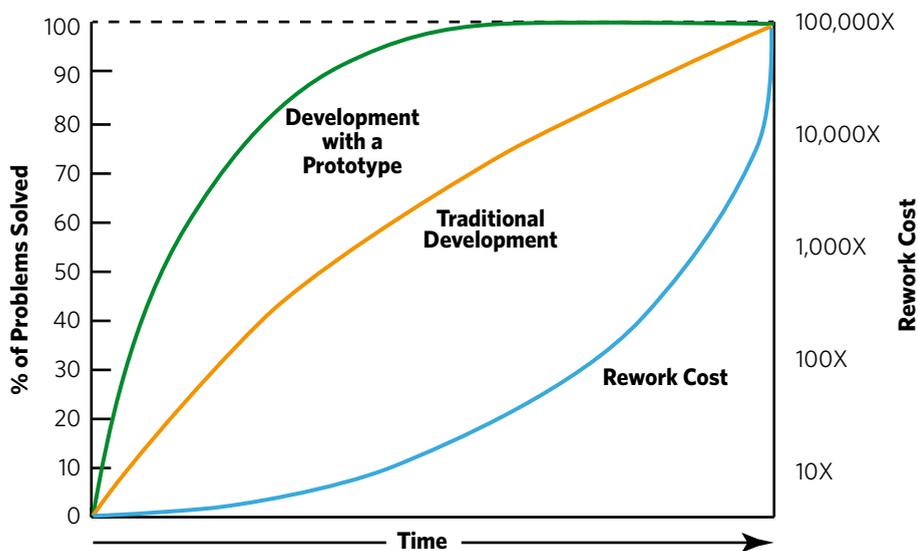
- Ensure the project manager and key team members are assigned to project for 4 to 5 years to establish accountability and continuity for the program office.
- Conduct an unbiased, independent review of the program with outside experts prior to Milestone B and at key design points to counter overly optimistic estimates.
- Develop a detailed end-to-end risk management plan in the pre-acquisition phase—prior to Milestone B—that identifies program risks early in the project life cycle. This is crucial.
- Develop rigorous Milestone B entrance and exit criteria and ensure they are adhered to. Issue liens if the criteria are not satisfied.

To avoid suffering the same fate as many of the programs in Figure 1, a best practice is to mature critical technologies to a Technology Readiness Level (TRL) 6 prior to Milestone B. TRL 6 is defined as "testing a system or subsystem model or prototype relevant environment." By achieving TRL 6, the project will have burned down significant technology, cost, and schedule risk.

Another best practice for technology in complex projects is that it should be managed in a serial acquisition process—not in parallel with system development—in order to lower risk to the project. For example, one of the most ambitious and costly programs in the DoD, the Joint Strike Fighter (JSF), implemented a concurrent development approach and has suffered significant cost and schedule overruns.

Instability in the JSF program has been and continues to be the result of highly concurrent development, testing, and production activities. This has led to retrofitting already procured aircraft to correct deficiencies discovered during testing. The JSF is a complex project that is trying to simultaneously develop and field three aircraft variants for the Air Force, Navy, Marine Corps, and eight international partners. With respect to cost, the JSF project baseline in 2001 was for 2,866 planes at a total acquisition cost of \$233 billion and in 2012 skyrocketed to a total cost \$395 billion for 2,457 planes. Furthermore, the unit cost per aircraft has doubled since start of development in 2001 from \$69 million to \$137 million in 2012.

Figure 2. Prototypes Enable Earlier Problem Resolution



In summary, technology, system, and testing should not be done concurrently. A good rule of thumb is to mature technology to a TRL 6 prior to Milestone B. By meeting this technology metric, a project will burn down significant project risk and reduce the likelihood of cost overruns, schedule delays, and meeting technical performance requirements.

Experiment Early and Fail Early

Thomas Edison eloquently captured experimenting early and often with his famous quote, “Negative results are just what I want. They’re just as valuable to me as positive results. I can never find the thing that does the job best until I find the ones that don’t.” Edison was well aware of the importance of testing early through rapid and frequent experimentation. He was also very aware that failing is part of the process of learning.

Many leading-edge innovation firms exercise Edison’s philosophy by testing out new ideas by rapidly building mock-ups to test features and functions. At their simplest level, mock-ups may take the form of cardboard, clay, papier-mache, or three-dimensional simulations. The idea is to quickly build a visual representation of a product with its desired functions and features—a prototype.

Prototypes can serve multiple purposes. They can:

- Show the design is stable.
- Demonstrate that the user requirements are achievable.
- Serve as a learning tool.
- Provide early information on the system.
- Encourage communication among the customer, contractor, stakeholders, and team members.
- Provide a planned milestone iteration to adjust the design specification.
- Serve as a go/no-go decision point.

Besides serving multiple purposes, prototypes help solve program issues early and burn down risk early in the program life cycle. There are numerous examples in the literature—from automotive climate control systems, software team life cycle approaches, and automotive manufacturing—that demonstrate how prototypes significantly reduce manufacturing development time and effort. This is particularly relevant for complex DoD defense weapons projects that manufacture large quantities of weapons systems.

Figure 2 provides a graphical description of how building prototypes can accelerate problem resolution faster compared to traditional developments and reduce costly rework, which increases by roughly a factor of 10 between project acquisition phases.

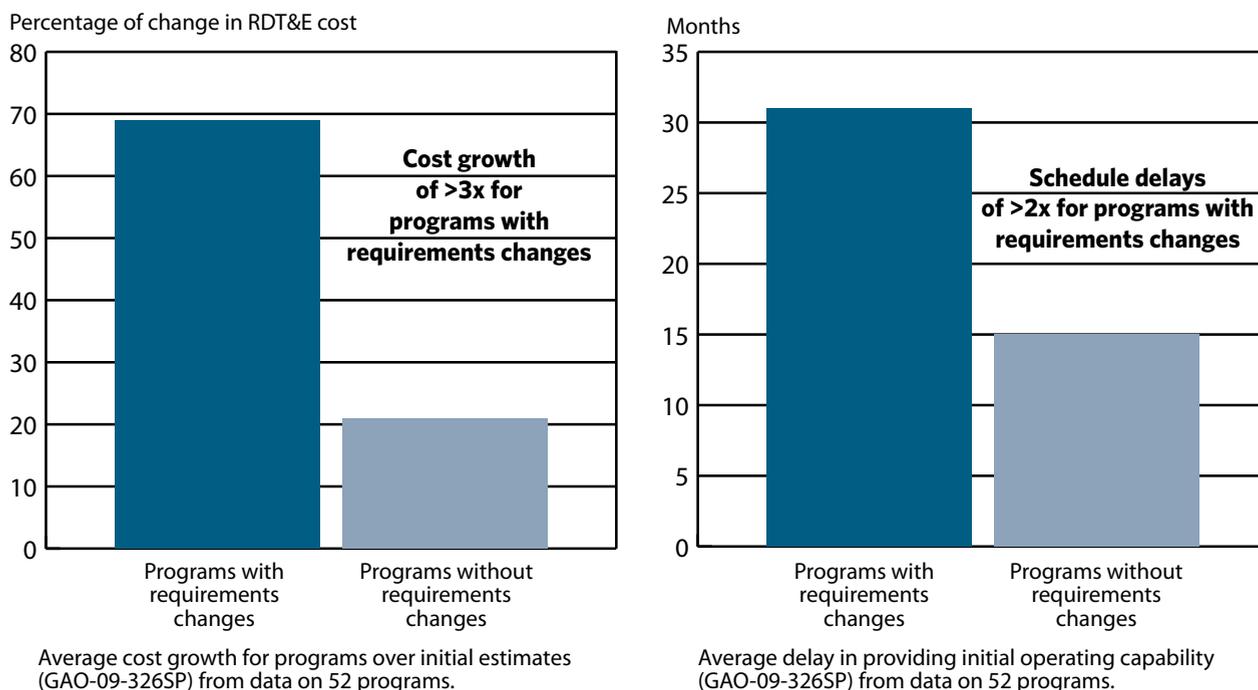
In summary, build prototypes—either hardware or software—to gain knowledge early, to reduce technology development time and effort, and to solve interface issues early. This approach will enable complex DoD projects to avoid costly rework and subsequent cost overruns and schedule delays later in the project.

Stop Requirements Creep

Requirements creep is one of the most cited reasons for cost overruns and schedule delays on DoD acquisition projects. Stopping requirements creep takes exceptional political acumen and a deep understanding of systematic impacts. When pressured to change requirements, it is the project leader’s job to explain to stakeholders that changing requirements in the project execution phase usually leads to program cost and schedule overruns.

In order to support this view, let’s look at GAO data in Figure 3 that show the impacts of changing requirements. Figure

Figure 3. Program Impacts of Changing Requirements



3 (left) provides data on 52 DoD weapons programs that changed requirements and shows that these programs suffered average cost growths of greater than a factor of 3, and Figure 3 (right) shows that the average schedule delay is greater than a factor of 2, compared to programs that did not change requirements.

In general, requirements change for several reasons: too many stakeholders with divergent needs and wants; no project approved requirements baseline at Milestone B; and agencies routinely accepting requirements changes post-Milestone B with no understanding of system impacts.

Requirement changes are a widespread problem in the DoD, and strong leadership is required to combat this trend. The most effective approach to avoid requirement changes is to enact the following steps:

- Have a vetted, approved requirements baseline prior to Milestone B.
- Implement a no-change requirements policy. Stick to it.
- Implement a change control board (CCB) and mandate a cost-benefit evaluation for any requirement change.
- Have a strong, politically astute project champion to help manage stakeholders.

Minimizing or having no requirements changes gives complex projects a chance to deliver a system that meets cost, schedule, and technical targets. In summary, have a vetted set of requirements early in the project; have a government-led CCB; and, most important, have a strong project champion.

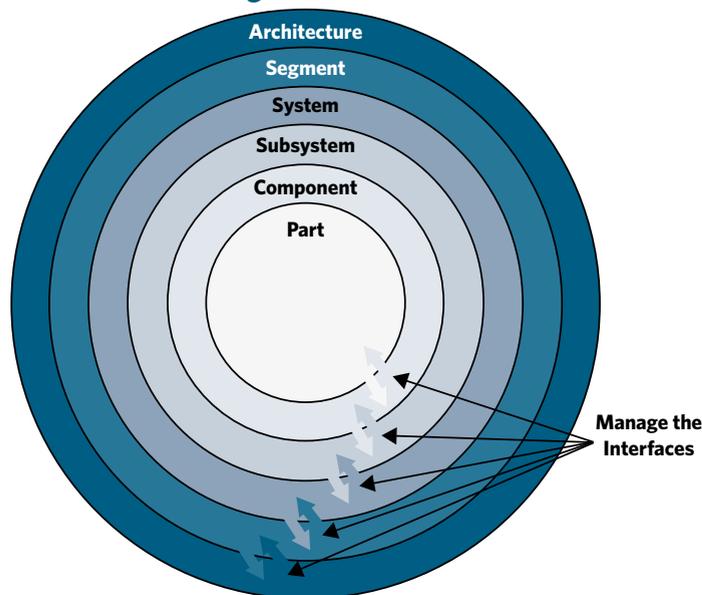
Take Great Care Managing Interfaces

Many complex projects suffer setbacks and failures by not clearly defining technical and organizational interfaces. Interfaces are points where a transfer of information occurs. A technical interface can be an optical, mechanical, electrical, thermal, or data transfer point. Internal organizational interfaces can be among the project managers, system engineers, contracts leads, budget leads, designers, builder, testers, operators, and users, while external organizational interfaces may occur between government agencies, the prime contractor, and subcontractor. As one can imagine, a complex DoD project may have millions of technical interfaces and tens of organizational interfaces that need to be managed (see Figure 4).

On most programs, the technical interfaces are managed by a system engineering integrated product team (IPT). This IPT is tasked with ensuring all interfaces are captured, specified accurately, and documented when changes occur. One best practice is to create a comprehensive, detailed interface control document (ICD) that identifies and documents all project interfaces as well as all unattended and mismatched interfaces. The ICD also should contain a configuration management (CM) plan to document and communicate all interface changes to the project team. There also should be a change control board (CCB) that meets daily or weekly to discuss and communicate interface changes. The organizational interfaces should be captured in a stakeholder communication plan.

Another best practice to minimize interface control and plan for obsolescence is to design in modularity and commonality to the system under development. Modularity refers to

Figure 4. Technical and Organizational Interface Management



designing in volumes on a system that can accommodate future technology 2, 4, or 8 times larger or smaller, while commonality refers to using standard interfaces. Additional benefits of incorporating modularity and commonality are that they will reduce the number of interfaces to manage and reduce switching costs for future systems.

By creating a comprehensive ICD, documenting and communicating interface changes with a rigorous CM process, creating a stakeholder communication plan, and incorporating in modularity and commonality, a project manager will significantly decrease the likelihood of interface issues on a complex project.

Create a Software Integrated Product Team

Functions performed by software continue to increase on many DoD weapons systems. For instance, data from the 2010 House Armed Services Committee (HASC) report show that the percent of functions performed by software has increased considerably over the past few decades on several weapons systems (see Table 1).

The same report provides dismal statistics on the success rate of DoD IT projects: Only 16 percent of IT projects were completed on time and on budget, 31 percent were canceled before completion, and 53 percent were late or over budget with typical cost growths exceeding 89 percent. Even more disturbing is that of the IT projects completed, the final products contained only 61 percent of the originally specified features. This is a poor report card for DoD software development programs.

The leader of a complex DoD project should ensure that software development be treated the same as hardware, with phases and milestones. In addition, the project manager

should ensure that most the efficient software development approach, such as spiral, agile, or waterfall be utilized. This task should be led by a software integrated product team (IPT).

Another best practice for software development is to perform rigorous regression testing for any software change. On one highly successful, large-scale, complex software project for a ground station, the contractor team implemented regression testing on every new or modified line of code and delivered the software system to the ground site with zero errors. Finally, prior to developing software, the project manager should take into account the final system configuration and ensure that the development code and software system interfaces are compatible, the computational complexity is not too high, and that the algorithms meet the system specifications and are scalable.

In summary, treat a DoD software project like a hardware project with phases and milestones; create a software IPT; utilize a development lifecycle consistent with the project’s complexity and requirements; track and document software interfaces; ensure the software is scalable; and, finally, perform regression testing to ensure that a high-quality product that meets all specifications is delivered to the final system.

Summary

There is a quote from Albert Einstein that is very relevant to complex projects: “Any intelligent fool can make things bigger and more complex. ... It takes a touch of genius—and a lot of courage—to move in the opposite direction.” In many cases, it is organizations, agencies, and senior committees that make projects bigger and more complex than they need to be. My hope is that this article will provide leaders of complex projects with the data and the courage to reduce complexity and deliver complex projects within scope, cost, and schedule. &

The author can be contacted at srmeier@srmconsultingllc.com.

Table 1. Percentage of Functions Performed by Software on Several Weapons Systems

Weapon System	Year	% of Functions Performed in Software
F-4	1960	8
A-7	1964	10
F-111	1970	20
F-15	1975	35
F-16	1982	45
B-2	1990	65
F-22	2000	80