Research Article

A Framework for Studying Cost Growth on Complex Acquisition Programs

Morgan Dwyer,[1,*] Bruce Cameron,[1] and Zoe Szajnfarber[2]

[1] *Massachusetts Institute of Technology*, 77 *Massachusetts Ave*. 33-409, *Cambridge*, MA, 02139
[2] *The George Washington University*, 1776 G St NW *Suite* 101, *Washington*, DC, 20052

Received 22 March 2015; Revised ; Accepted 18 October 2015, after one or more revisions
Published online in Wiley Online Library (wileyonlinelibrary.com).
DOI 10.1002/sys.21328

ABSTRACT

The government acquisition system is consistently plagued by cost growth and by attempts at acquisition reform. Despite these persistent challenges, the academic community lacks a methodology for studying complex acquisition programs both in-depth and longitudinally throughout their life cycles. In this paper, we present a framework for studying complex acquisition programs that provides researchers with a strategy for systematically studying cost growth mechanisms. The proposed framework provides a means to identify specific technical and organizational mechanisms for cost growth, to organize those mechanisms using design structure matrices, and to observe the evolution of those mechanisms throughout a program's life cycle. To illustrate the utility of our framework, we apply it to analyze a case study of the National Polar-orbiting Operational Environmental Satellite System (NPOESS) program. Ultimately, we demonstrate that the framework enables us to generate unique insights into the mechanisms that induced cost growth on NPOESS and that were unacknowledged by previous studies. Specifically, we observed that complexity was injected into the technical system well before the program's cost estimates began to increase and that it was the complexity of the NPOESS organization that hindered the program's ability to effectively estimate and to manage its costs. © 2015 Wiley Periodicals, Inc. Syst. Engin. 00: 1–16, 2015

Key words: cost growth; system architecture; DSM

1. INTRODUCTION

Cost growth is an endemic issue (e.g., see Younossi [2007], GAO [2009a], and Hofbauer et al. [2011]) that plagues the government's acquisition of complex systems and that has been the focus of numerous studies that attempt to identify cost growth root causes that can be corrected by acquisition reform. Past studies often attribute cost growth to an acquisition program's requirements [DSB, 2003; GAO, 2003, 2006b, 2011, 2012a; Younossi, 2008], to its use of immature technology [GAO, 2003, 2006b, 2007b, 2011; GAO, 2012b; Azizian et al., 2011; Katz et al., 2015], and to its unrealistic cost estimates [Augustine, 1997; DSB, 2003; Nowinski and Kohler, 2007; Younossi, 2008; Blickstein, 2011; OSD, 2013]. However, despite identifying these and other root causes (e.g., see Augustine [1997], Coleman, Summerville, and Dameron [2003], and OSD [2013]) and targeting them during reform efforts, the performance of the acquisition system remains poor. For example, in 2012 alone, the National Aeronautics and Space Administration (NASA) and the Department of Defense's (DoD) space system program costs increased by $2.5 billion and $11.6 billion, respectively [GAO, 2006b, 2012a].

*Author to whom all correspondence should be addressed (e-mail: morgan.dwyer@aya.yale.edu).

Given the persistent problem of cost growth on acquisition programs, the government could benefit from an improved understanding of the specific technical and organizational *mechanisms* that induce cost growth in the acquisition system.

In this paper, we propose a framework that enables researchers to study complex acquisition programs and to enhance the acquisition community's current understanding of the cost growth that they often exhibit. In the framework, we use an in-depth analysis of a program's history to identify cost growth mechanisms, to organize those mechanisms in design structure matrices (DSMs), and to observe their evolution throughout a program's life cycle. The proposed framework has a distinct advantage over other methods (e.g., see Brown, Flowe, and Hamel [2007], Malone and Wolfarth [2013], and Brown [2014]) because it enables the researcher to analyze a program in the detail that is necessary to identify specific mechanisms for cost growth. However, in addition to preserving the complexities of a program's history at any point in time, the framework also enables the researcher to study a program longitudinally and to observe how cost growth mechanisms evolve throughout a program's life cycle. In the following sections, we review literature that motivates our approach for studying complex acquisition programs and then continue by defining the framework's six steps. Next, we illustrate the utility of the framework by applying it to study the National Polar-orbiting Operational Environmental Satellite System (NPOESS), a joint acquisition program that experienced significant cost growth prior to its cancelation. Finally, we conclude by suggesting opportunities to apply the framework to study cost growth on other acquisition programs and to enhance our proposed methodology in the future.

2. LITERATURE REVIEW

The concept of *architecture,* which Crawley, Cameron, and Selva [2015] define as "an abstract description of the entities of a system and the relationships between those entities," organizes our framework. While the field of system architecture traditionally focuses on the architecture of technical systems [Maier and Rechtin, 2000], organizational theorists often study organizations as systems [Katz and Robert, 1978] which can also be defined in terms of their components and component relationships. To study the systems acquired by the government and the organizations that acquire them, we recommend that researchers analyze *both* a program's technical and organizational architecture (as in Sosa, Eppinger, and Rowles [2007]).

Studying a program's architectures also provides the researcher with a unique perspective on complexity, since complexity is an inherent property of a system's architecture [Simon, 1962]. Although numerous definitions and measures of complexity exist, authors generally agree that complexity is induced by a system's individual components, by their organization in the system, and by their interactions with each other. Furthermore, researchers have demonstrated that complexity correlates with a system's cost [GAO, 2003, 2006b, 2007b] and that as a system's complexity increases, so does the risk that its development costs will grow [GAO, 1992; NASA, 2012]. In our framework, we adopt the concept of complexity as a useful abstraction and theoretical construct with which to identify, classify, and organize mechanisms that contribute to a program's cost.

Three types of complexity are particularly useful for studying a program's technical architecture: architectural, design, and process complexity. Architectural complexity refers to the number of components [Meyer and Lehnerd, 1997], the pairwise interfaces between components [Meyer and Lehnerd, 1997], and to the larger arrangement of components in a system [Sinha and de Weck, 2013]. Design complexity refers to the complexity of the individual components in a system; oftentimes technology readiness level (TRL) is used to approximate component design complexity [Bearden, 2003; Dubos and Saleh, 2011; Sinha and de Weck, 2013]. Finally, process complexity is not a function of a technical system itself, but rather, is a function of the external processes by which a system is developed. The idea of process complexity was discussed by Sussman, who introduced the concept of "nested complexity," which refers to a complex technical system that is "embedded within an institutional system that exhibits . . . complexity all on its own" [Sussman et al., 2009]. The government system acquisition process has been institutionalized by strict requirements to control quality and manage risk; therefore, we use the concept of process complexity to capture instances when those requirements are levied on the system development process and increase its cost.

To understand the complexity of government acquisition organizations, researchers can glean insights from public administration and organizational theory. Theorists often define organizational architectures in terms of an organization's components and the interdependencies between them [Galbraith, 1974; Nadler and Tushman, 1997]. Importantly, it is also theorized that organizations perform most effectively when those interdependencies are aligned [Nadler and Tushman, 1997]. Drawing from public administration theory, we observe that key characteristics of the government bureaucracies that acquire systems are their authority structures, their missions, their budgets, and their unique expertise [Weber, 1952; Downs, 1972; Wilson, 1989]. Thus, we suggest that in order for a government organization to perform effectively and efficiently, the key interdependencies between its components—authority, responsibility, budget, and expertise—must be aligned. When these interdependencies are not aligned, the organization is unnecessarily complex, its performance declines, and its costs grow.

3. PROPOSED FRAMEWORK

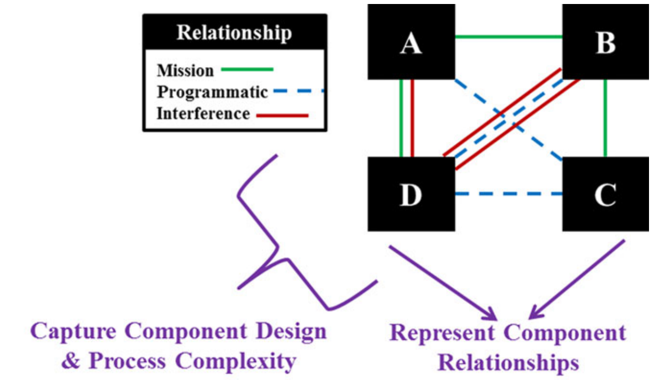
Our proposed framework is motivated by the above understanding of organizational and technical complexity and by complexity's relationship to a program's architectures and cost. The purpose of the framework is to provide a means for researchers to organize, analyze, and discuss the multitude of data that must be collected in order to study a complex acquisition program both in-depth and longitudinally. Specifically, the framework provides a means:

- to capture and categorize technical and organizational mechanisms for cost growth,

- to assess the relative impact of those mechanisms,
- and to enable the technical and organizational evolution of a program to be observed.

To achieve these goals, the proposed framework contains six steps wherein a program's organizational and technical architectures are represented and metrics that assess their complexity are calculated. In the final step, the evolution of complexity is observed by plotting the complexity metrics against time.

To represent a program's architectures, we use DSMs. DSMs are typically N×N matrices that are used to represent product, organizational, or process architectures or some combination of all three [Eppinger and Browning, 2012]. Our framework defines two separate DSMs to represent a program's organizational and technical architectures and the complexity mechanisms within them. Previous studies [Sosa, Browning, and Mihm, 2007; Suh et al., 2009] have demonstrated the utility of using DSM-based metrics to study the evolution of architectures; therefore, to observe how a program's technical and organizational architectures evolved, we calculate complexity metrics using the data contained in the DSMs.



	Design	Process		Architectural			
				A	B	C	D
Component A	5	2	A		1	1	2
Component B	4	1	B	1		1	3
Component C	2	0	C	1	1		1
Component D	1	0	D	2	3	1	

Figure 1. Example technical architecture DSM.

3.1. Step 1: Collect Process Data about the Program

To construct the DSMs and to identify the complexity mechanisms within each architecture, we conduct semi-structured qualitative interviews [Eisenhardt and Graebner, 2007; Eppinger and Browning, 2012] with program staff and triangulate our interview data using primary and secondary source documents [Eppinger and Browning, 2012]. The goal of data collection and analysis is to gain an in-depth understanding of the mechanisms that induced cost growth at any point in time during a program's life cycle. Technical cost growth mechanisms are those that make a system's cost increase both expectedly and unexpectedly. Organizational cost growth mechanisms are those that hinder the organization's decision-making process by limiting the options that are available for each decision, by reducing the quality of the organization's decisions, or by hindering the efficiency with which the organization is able to make decisions. To identify cost growth mechanisms, we recommend that researchers analyze data using process tracing [Langley, 1999; George and Bennet, 2005] and in the subsequent steps of our framework, we describe how the resulting process data can be characterized according to its relationship to a program's technical and organizational architecture, abstracted using the concept of complexity, and represented using DSMs.

3.2. Step 2: Represent the Technical Architecture

First, all major technical components are represented in the technical architecture DSM (DSM_T) using a traditional DSM format that assigns system components to corresponding rows and columns in an N×N matrix. To also represent each component's design and process complexity, DSM_T has two additional columns that contain this information. In this way, DSM_T is able to represent all three of the complexity types that were discussed above:

- ***Design complexity***, which is a function of the technical maturity of each component.
- ***Process complexity***, which is a function of the constraints or conflicting requirements that are imposed during the component development process.
- And ***architectural complexity,*** which is a function of the interactions and relationships between components.

As shown in Figure 1, architectural complexity mechanisms are represented using traditional DSM notation where +1 is added to indicate the presence of *any* relationship between two components. Three relationship types are captured—mission, programmatic, and interference—and the presence of each relationship adds +1 to the corresponding DSM_T entry; components can share more than one relationship and each relationship type adds +1 to the corresponding entry in DSM_T.

Mission relationships between components include physical, data, or design interfaces as defined in Smaling and de Weck [2007]. Physical interfaces mean that two components are physically attached and may also share other relationships (such as data or power). Two components have a data (but not a physical) relationship when they communicate at a distance and two components have a design relationship when they are designed to enable parts sharing (e.g., they are designed to maximize commonality).

Programmatic relationships indicate that components share management resources such as budget, schedule, and staff. Although this relationship is not purely technical, we include it in the DSM because programmatic relationships can induce cost growth: specifically, even though two components may not share a mission interface, they may still interface programmatically because the budget, schedule, and staff assigned to one component can impact the resources that are allocated to the other. For example, if a component's costs grow but the program's budget is fixed, management may decide to prioritize one component's development at the expense of others, whose budgets will be reduced and schedules lengthened; this decision will ultimately increase the total nonrecurring cost of the lower priority components [Committee on Cost Growth in NASA Earth and Space Science Missions, 2010]. To capture this behavior, when two components have a programmatic relationship, +1 is added to DSM_T.

Finally, +1 is added to DSM_T entries to account for *interferences* between components. As noted by Selva [2014], Alibay and Strange [2013], and Rasmussen and Tsugawa [1997], components can interfere electromagnetically, mechanically, and optically. Interferences induce complexity because they must be actively managed and compensated for during the system development process.

As also shown in Figure 1, DSM_T includes two extra columns that contain a design and process complexity score for each component. The design complexity score captures the degree of cost risk associated with the component's development, since as its technical maturity increases, so does the certainty with which a program can estimate its development cost [Dubos and Saleh, 2011]. Brady and Nightingale [2012] previously demonstrated the utility of including technical maturity in a DSM when they developed the technology risk DSM to assess development and operational risk in NASA systems. Although their approach used a standard TRL system to categorize component maturity [Brady and Nightingale, 2012], other rating schemes can also be applied. For example, AIAA categorizes a component's design maturity according to a component's location in the traditional development life cycle [AIAA, 1999].

While there is no formal scheme to categorize a component's process complexity, we suggest that +1 can be added to account for each process complexity mechanism. Process complexity penalties should only be levied to capture macroscopic changes to the system development process that affect multiple components. For example, a key process complexity mechanism that affects joint programs is conflicting mission requirements or engineering standards. This complexity mechanism captures the costs that emerge when a program's schedule is delayed because engineers are forced to reconcile requirements conflicts or to use multiple requirements baselines. Another process complexity mechanism is a function of a program's oversight model. If the government's oversight of a system's development is high, +1 should be added to the process complexity score. For example, if a component is developed under a system engineering and technical assistance (SETA) oversight model instead of a total system performance responsibility (TSPR) oversight model, its process complexity score should increase by +1. In both examples, penalties are included to capture major changes—like the addition of new mission requirements, engineering standards, or oversight requirements—that occur during a program's life cycle: process complexity penalties are not intended to capture microscopic changes that only affect individual components' development processes. Finally, although process and product architectures have shared DSMs differently in the past [Eppinger and Browning, 2012], because process ultimately affects the cost of the technical system, our framework represents both complexities using a shared DSM.

3.3. Step 3: Calculate the Technical Complexity Metric

After the DSM_T is defined, we use it to calculate a technical complexity metric that accounts for design, process, and architectural complexity's impact on a system's nonrecurring cost. Ultimately, our metric is an estimate of the technical architecture's life cycle cost, with penalties applied to its nonrecurring cost that account for design, process, and architectural complexity mechanisms.

Equation (1) shows the general form of the complexity-corrected life cycle cost metric (L_{cc}), which includes the nonrecurring costs with complexity penalties applied (N_{cc}), recurring costs (R), and other costs (O) that can include launch or operations costs. As shown, L_{cc} is normalized by the cost of a reference system.

$$Lcc = \frac{N_{CC} + R + O}{Baseline\,Complexity}. \tag{1}$$

The formula for N_{cc} is derived from a complexity metric that was used to study spacecraft architectures for planetary exploration [Alibay and Strange, 2013]; however, by classifying mechanisms in terms of three complexity types—design, process, and architectural—our metric is generalizable to systems other than spacecraft as well:

$$\begin{aligned} Ncc = \sum\nolimits_{i=1}^{N} \Big\{ \Big(\sum\nolimits_{j=1}^{N} DSM_{T(i,j)} - 1 \Big) * W_A \\ + PC_i * W_P + 1 \Big\} * C_i(DC_i). \end{aligned} \tag{2}$$

As shown, N_{cc} is calculated using each component's design complexity (DC_i) and process complexity (PC_i), DSM_T, and the cost penalty weighting for each complexity mechanism (W_A for architectural complexity and W_P for process complexity). Component cost (C_i) is estimated using system-specific parametric cost-estimating relationships and is corrected for design complexity ($C_i(DC_i)$) either by adding a penalty to component mass prior to estimating its cost (as in Selva [2012] and Alibay and Strange [2013]) or by adding a penalty after its costs have been calculated (as in Leising et al. [2010]). The weightings applied to correct for process and architectural complexity are also system-specific and should be determined on a case-by-case basis.

This particular complexity metric was selected for several reasons. First, by using parametric cost estimating relationships, the metric captures fundamental hardware and software costs. The metric then adds penalties to account for the complexity mechanisms that are present in a system's architecture. Although these mechanisms may not ultimately increase a system's cost, their presence places the system at greater *risk* for cost growth if additional risk *margin* is not included in the program's budget. In this way, our proposed metric calculates the basic cost of the system and adds a complexity budget [Sinha, 2013] or a complexity *margin* on top of that cost. The reason for adding this margin is simple: if past programs had recognized and budgeted for the complexity of their systems,

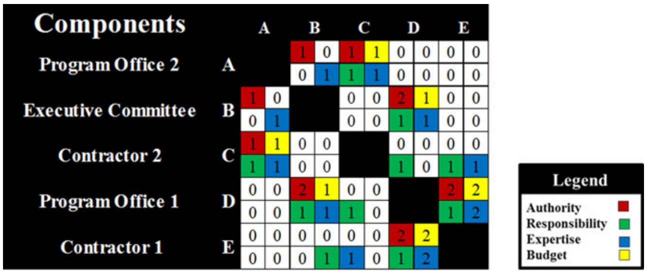


Figure 2. Example organizational architecture DSM.

their costs may not have exceeded their budgets. The process that we follow to identity and account for this cost risk is derived from the concept of cost-risk subfactors that is employed by the Jet Propulsion Laboratory [Leising et al., 2010, 2013].

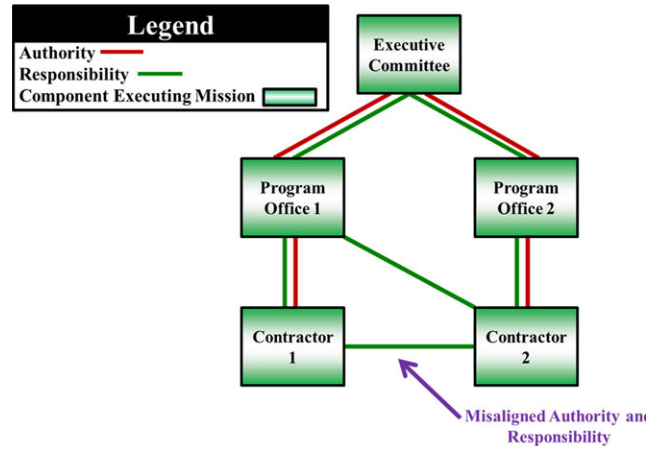


Figure 3. Misalignment of responsibility and authority.

3.4. Step 4: Represent the Organizational Architecture

The organizational DSM (DSM_O) uses a traditional DSM format that assigns the organization's components to corresponding rows and columns in an N×N matrix; organizational components refer to distinct subunits in an organization that include government agencies, user communities, program offices, and contractors. As shown in Figure 2, DSM_O maps four distinct interdependencies between components and indicates interdependency strength; to be consistent with the organizational complexity metric defined below, a score of +2 is used when components' relationship is weak and +1 is used when the relationship is strong. Consistent with the key characteristics of government organizations that were discussed above, the four interdependency types are defined in terms of:

- ***Expertise*:** When a component has expertise (E), it has the knowledge and experience to make decisions effectively. DSM_E is shown in blue.
- ***Responsibility*:** When a component has responsibility (R), it is responsible for delivering a technical system that executes a mission. Responsibility will also be referred to as *mission* responsibility when it is necessary to distinguish between multiple agencies' unique missions. DSM_R is shown in green.
- ***Budget*:** When a component has budget (B), it is responsible for funding the decisions that it makes and the technical system for which it is responsible. DSM_B is shown in yellow.
- ***Authority*:** Finally, when a component has ***authority*** (A), it is able to make and sustain effective decisions. DSM_A is shown in red.

Although the four relationship types are depicted separately, the two relationships that contribute most significantly to the organizational complexity metric (defined in the next step) are *responsibility* and *authority.* An example mission responsibility relationship between two component contractors is illustrated in Figure 3. These contractors share a mission responsibility relationship because the technical components that they produce share an interface; in this way, responsibility relationships between contractors *mirror* [Henderson and Clark, 1990; Baldwin and Clark, 2000] the program's technical architecture. However, in addition to this mirroring, mission responsibility relationships on government programs extend throughout the program's organizational hierarchy and ultimately connect agency leaders, who Congress holds responsible for mission execution, to the contractors that agencies hold responsible for a system's development.

DSM_A represents the organization's authority structure. As shown in Figure 3, Program Office 1 holds a contract for component 1 and Program Office 2 holds a contract for component 2; additional decision authority relationships between components are also illustrated. Figure 3 also notes that although there is a responsibility link between the component contractors, there is no authority relationship. This misalignment of responsibility and authority is critical to the organizational complexity metric that is discussed below. Finally, in addition to the misalignment of responsibility and authority, additional misalignments between authority, responsibility, expertise, and budget can also affect an organization by eroding authority. Although these relationships can be conceptualized in terms of their impact on decision authority, we represent them in separate DSMs because they are key characteristics of government acquisition organizations.

3.5. Step 5: Calculate the Organizational Complexity Metric

To assess organizational architectures, we use a separate metric and define organizational complexity (*OC*) to be a function of the number of components in an organization, the interfaces between each component, and components' mission responsibility and decision authority. We suggest that as an organization's complexity increases, it becomes more difficult for the organization to make effective and efficient decisions; as a result, complex organizations are more likely to enable cost growth by selecting costly options for each decision and to induce cost growth by making decisions slowly.

The proposed organizational complexity metric, given in Equation (3), is derived from a structural complexity metric proposed and validated by Sinha and De Weck [2013]. To apply Sinha's metric to assess complexity in organizations, we developed a new weighting scheme to account for the organizational complexity mechanisms that were identified in our qualitative data. The final organizational metric is shown in Equation (3) where W_A corresponds to the weighting scheme, N corresponds to the number of components, and $E(DSM_R)$ corresponds to singular values of the DSM_R matrix. Finally, like L_{cc}, OC is normalized by a reference organization:

$$OC = \frac{N + \left\{ \sum_{i=1}^{N} \sum_{j=1}^{N} W_{ij} * DSM_R(i,j) \right\} * \left[\frac{1}{N} \right] * E\left(DSM_R(i,j) \right)}{Baseline\ OC}. \quad (3)$$

The process for calculating Equation (3)'s weights begins by adjusting DSM_A to account for factors that erode decision authority. For each factor, a score of +1 is added to each affected decision authority link. The primary authority erosion factors stem from misalignments between responsibility, authority, budget, and expertise. Specifically, we suggest that misalignments between authority and budget and responsibility and budget erode authority by hindering an organization's ability to consider cost when making decisions. Additionally, misaligned authority and expertise erodes authority by reducing an organization's ability to make informed and effective decisions. Finally, in addition to including misalignments that erode authority, the metric can also include additional case-specific factors by simply adding +1 to each interface where authority is eroded.

Once DSM_A is adjusted to account for authority erosion factors, DSM_A and DSM_R are compared and misalignments between the two are identified. If authority and responsibility between two components are misaligned, an additional penalty is calculated and added according to the following process: first, the adjusted DSM_A is transformed into a graph, where components are represented by nodes in the graph and authority links are represented by edges. Edge lengths correspond to the values in the adjusted DSM_A. Next, when a mission responsibility link between two components exists, the weighting, W_A, between those components is calculated by determining the shortest path length between components i and j in the authority graph. The length of the authority path between two components with shared responsibility is intended to simulate the efficiency and effectiveness of organizational decision making.

Specifically, effectiveness is simulated by accounting for authority erosion factors; as more factors erode a particular link's authority, the length of its edge in the DSM_A graph increases to capture the challenge of making decisions in the absence of strong authority. Efficiency is simulated by calculating the length of the path that has to be traversed to find a single node that has authority over two nodes that are connected in DSM_R but not in DSM_A. In this way, the metric captures the delay that occurs when two organizational components have a responsibility but no authority relationship. In the absence of authority, the responsible components must elevate decisions up the organizational hierarchy to find a senior component with the authority to make decisions that affect both of the responsible subordinates. For example, in Figure 3, Contractor 1 and Contractor 2 share responsibility but not authority. Therefore, the contractors must elevate any decisions that affects both of them through their individual program offices to the Executive Committee: the only component in the organization that has authority over both contractors. The process of elevating decisions through both program offices and of waiting for the Executive Committee to make decisions is less efficient than if the contractors could negotiate and implement a decision on their own.

3.6. Step 6: Observe the Evolution of Complexity over Time

The utility of calculating a single complexity metric to represent a program's organizational and technical architectures is that it enables changes in those architectures and their relationship to the program's reported cost growth to be observed over time. Specifically, we suggest that complex acquisition programs can be studied by dividing the program's history into *epochs,* or periods of time when the program's organizational and technical architectures are unique and stable [Dwyer and Szajnfarber, 2014]. One organizational and one technical architecture should be defined per epoch, a complexity metric for each should be calculated, and complexity should be plotted as a function of time and compared to the program's cost during each epoch. Plotting the complexity metrics in this way enables the researcher to compare the program's own cost estimates to the complexity that was inherent in its architectures.

4. CASE STUDY: THE NPOESS PROGRAM

To study cost growth on the NPOESS program, we applied the above framework to generate unique conclusions about the program's history and evolution [Dwyer and Szajnfarber, 2014; Dwyer et al., 2014; Dwyer, 2015]. NPOESS was a collaboration composed of the DoD, the National Oceanic and Atmospheric Administration (NOAA), and NASA that was intended to develop a constellation of environmental monitoring satellites for low Earth orbit. NPOESS was managed by an integrated program office (IPO) that was staffed by representatives from all three agencies. The IPO reported to the program's executive committee (EXCOM), which was a triagency board composed of the NOAA Administrator, the NASA Administrator, and the Under Secretary of Defense for Acquisition, Technology, and Logistics. NPOESS was established in 1994 and canceled—due to cost growth, schedule delays, and management issues [The White House, 2010]—in 2010. In the intervening years, the program's history can be organized into six epochs which include the following events:

- **Epoch A** (1994–1996): NPOESS was established by converging NOAA's Polar-orbiting Operational Environmental Satellite (POES) program and the DoD's Defense Meteorological Satellite Program (DMSP). By converging these programs, the government predicted that it could save $1.3 billion over the life cycle of the joint program [Gore, 1993]. Importantly, these early cost estimates assumed that the NPOESS system would be only moderately more capable than POES or DMSP.
- **Epoch B** (1996–1999): The NPOESS system requirements were defined and as a result, the system's capabilities increased. Also during this period, the IPO managed

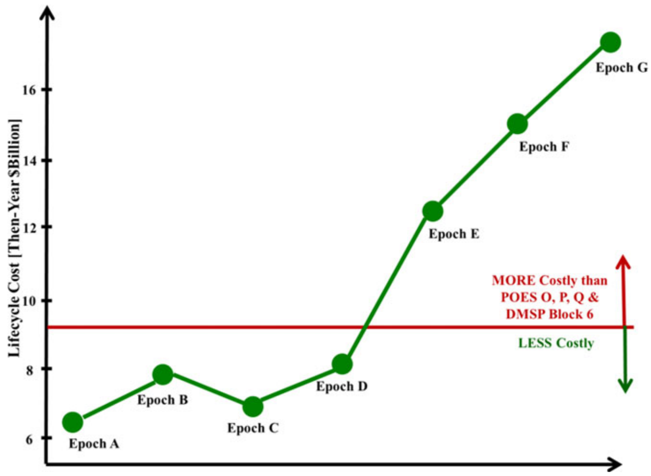


Figure 4. NPOESS life cycle cost growth (data taken from COBRA [2000] and GAO [2002, 2004, 2005, 2007a, 2010, 2012c]).

risk reduction contracts for the system's key instruments but delayed selecting the prime contractor that would serve as the system's integrator.

- **Epoch C** (1999–2002): The NASA-managed NPOESS Preparatory Project (NPP) was established to provide additional risk reduction for key sensors and to execute a climate science mission for NASA. To meet the needs of the system's new climate science users, instrument requirements were updated to enhance performance.
- **Epoch D** (2002–2005): The prime contractor was selected and the risk reduction contracts that were previously managed by the IPO were transferred to the new prime. After the prime assumed responsibility for all of the system's components, the program's cost estimates began to increase.
- **Epoch E** (2005–2007): Costs grew so significantly that the program breached the Nunn-McCurdy threshold and had to undergo certification.[1] As a result, several instruments and spacecraft were canceled and a Program Executive Officer (PEO) was added to streamline decision making between the IPO, NPP, and the EXCOM.
- **Epoch F** (2007–2010): Several instruments were added back to the program and despite its new PEO-based authority structure, management challenges and cost growth persisted until the program was canceled in 2010.

Figure 4 illustrates the cost growth that occurred before NPOESS's cancelation. A key characteristic of this plot is that the program's early cost estimates were relatively constant and that it was not until Epoch D that its cost estimates began to increase; this perspective is echoed by several studies that attributed the program's cost growth to poor management by its prime contractor [GAO, 2004; DoC, 2006]. Other studies attribute cost growth to the EXCOM's failure to meet and make decisions with sufficient frequency [GAO, 2005, 2006b, 2009b; NPOESS IRT, 2009]. Finally, despite focusing on cost growth, past studies fail to acknowledge that—like cost—the program's technical and organizational architectures also evolved throughout its life cycle.

[1]The Nunn-McCurdy Act requires defense programs to be recertified by Congress if their cost growth exceeds a specified level.

4.1. Step 1: Collect Process Data about the Program

To apply our framework to study the evolution of NPOESS's architectures, we collected a mix of qualitative and quantitative data that included over 75 hours of semi-structured interviews [Eisenhardt and Graebner, 2007; Yin, 2009] with over 50 representatives from the program. As recommended by Eisenhardt and Graebner [2007], we sampled interviewees from multiple levels in the program's organizational hierarchy and we triangulated [Yin, 2009] our data using over 150 primary and secondary source documents. Using this data and our proposed framework, we generated a new perspective on the evolution of the NPOESS system and organization and demonstrated that previous studies' understanding of program events is incomplete. Specifically, by focusing on the program's architectures, their complexity, and complexity's evolution over time, we were able to gain a unique understanding of the mechanisms for cost growth that most significantly affected NPOESS.

4.2. Step 2: Represent the Technical Architecture

The NPOESS DSM_T contains three types of components: spacecraft, instruments, and ground processing systems. The operational NPOESS constellation used three common spacecraft that were assigned to different longitude of ascending node crossing times and one unique spacecraft for NPP. The DSM_T also contains 12 different instruments that were assigned to one or more spacecraft and components that represented the system's algorithms and ground system. Although the ground system ultimately executed the algorithms, we represent the components separately so as to capture the critical role that the components' *interface* had on the program's subsequent cost growth.

Instrument design complexity is categorized using a ranking system similar to TRL but that is specifically focused on the instruments' relationship to heritage designs and is based on the approach recommended by [AIAA, 1999]. Spacecraft designs are assigned a complexity penalty of +1 for each type of complexity mechanism that is contained in the architecture. The process complexity mechanisms that affect each component are assigned penalties in the same way. Finally, the architectural complexity mechanisms that impacted NPOESS include design and programmatic relationships, physical interfaces, and electromagnetic, optical, mechanical, and reliability budget interactions between components.

We identified the complexity mechanisms contained in the DSMs by examining qualitative data *within* each epoch. Analyzing data within epochs enables the researcher to gain an in-depth understanding of the cost growth mechanisms that can affect a program at different points in its life cycle. Table I illustrates the detail with which we collected data within

Table I. Example Qualitative Data That Are Represented in Epoch B's Technical Architecture DSM

Complexity Type	Instance	Description
Architectural	EMI Interference	An active radar altimeter was hosted on the same spacecraft as instruments that were sensitive to electromagnetic interference (EMI). To enable the altimeter to be hosted alongside these sensitive instruments without interfering with their performance, instrument designs had to be augmented to add extra shielding.
	EMI Interference	Two additional EMI-sensitive instruments were also hosted on the same spacecraft. As a result, neighboring instruments' designs had to be augmented to prevent them from disturbing these more sensitive instruments.
	Mechanical Inference	A large conically spinning microwave imager-sounder induced jitter on the NPOESS spacecraft. To prevent this jitter from disturbing other instruments, the bus design included a special mechanical isolation system. Individual instrument designs were also augmented to add additional shielding that protected them from mechanical perturbations.
	Reliability Budget	When a spacecraft hosts multiple payloads, different instruments can fail at different times and if a mission critical instrument fails, then the entire spacecraft may have to be replaced. To avoid this, each of NPOESS's four mission critical payloads had to perform at the same high reliability so that they would all fail at approximately the same time. Increasing the instruments' reliability increased their cost.
	Mission Relationships	The program used a common spacecraft bus across the constellation; therefore, the bus was designed to accommodate every instrument even if it ultimately flew without them. This induced cost because instead of tailoring each bus to meet the reduced requirements of a single orbit's payload, the common NPOESS bus had to be designed to simultaneously meet the requirements of all of the constellation's payloads. As a result, each of the bus's subsystems had to be designed to meet the instruments' most stressing requirements.
	Mission Relationships	In order to fly common instruments and buses across the constellation, both components had to be designed and demonstrated to be compatible with multiple orbital environments. The resulting system was ***over-designed*** for any particular orbit and the complexity that was induced by adapting instrument and spacecraft designs to be common ultimately increased their non-recurring costs.
	Mission Relationships	Compared to Epoch A's technical architecture, four new instruments were added to the system during Epoch B.
Design	Instrument	Compared to NOAA and the DoD's heritage systems, NPOESS levied more severe requirements on its system. These requirements impacted the designs of the cross-track infrared sounder, the microwave imager-sounder, and the visible/infrared imager radiometer by making it impossible for NPOESS to leverage either agency's heritage instrument designs.
	Instrument	Despite levying more stringent requirements on the infrared imager-sounder, the program had the opportunity to leverage a NASA heritage grating spectrometer. Instead of using this instrument, the program required that its infrared sounder be a Fourier transform instrument and as a result, the program had to develop an entirely new instrument.
	Instrument	Each agency levied unique and driving requirements on the visible/infrared imager-sounder and this increased the instrument's design complexity. In particular, the instrument's design was affected by requirements for high radiometric accuracy, high resolution imagery, low light imagery, constant resolution across the instrument's cross-track scan, and climate science-quality data like that which had been produced by NASA's heritage instrument.
	Spacecraft	Several subsystems in the contractor's standard bus had to be re-designed to accommodate the high resource requirements that were levied by the NPOESS payload.
	Spacecraft	To accommodate the large amount of data that was generated by the program's instruments, a new IEEE 1394 data bus called Firewire had to be space qualified for use on NPOESS.

epochs and it maps this data to the complexity mechanisms that are represented in Epoch B's DSM. For the qualitative data that we used to construct DSMs for other epochs, refer to Dwyer [2015].

4.3. Step 3: Calculate the Technical Complexity Metric

The process for calculating N_{cc} begins by applying design complexity penalties (taken from AIAA [1999] and Nigg, O'Brian, and Abbott [2011]) to the instruments' mass. Instrument cost is then calculated using parametric equations given in Larson and Wertz [2011]. Next, again, using parametric equations from Larson and Wertz [2011], complexity-corrected instrument mass is used to estimate spacecraft bus mass and bus nonrecurring cost. Finally, the remaining architectural, design, and process complexity penalties are applied to the nonrecurring cost estimates according to Equation (3). The penalties, W_A, applied for bus design complexity, process complexity, and architectural complexity are all taken from Leising et al. [2010, 2013].

Recurring system costs are also estimated using parametric equations from Larson and Wertz [2011] and the launch cost for each spacecraft is also included in L_{cc}. The NPOESS L_{cc} does not include ground system or operations costs because the NPOESS ground system architecture remained relatively constant throughout the program's life cycle. Instead, we model changes to the ground system using complexity penalties that are levied on the components of the space segment with which they interfaced. Finally, each NPOESS L_{cc} is normalized by the analogous nonrecurring, recurring, and launch costs for POES and DMSP. By using POES and DMSP as reference systems, we are able to compare NPOESS's complexity to the complexity and cost of the systems that it was designed to replace.

4.4. Step 4: Represent the Organizational Architecture

The NPOESS DSM_o contains all three government agencies, the EXCOM, both the NPOESS and NPP program offices, several councils of user groups, and the system's prime and subcontractors for the spacecraft, instruments, algorithms,

Table II. Example of the Qualitative Data That Are Represented in Epoch D's Organizational DSM

Complexity Type	Instance	Description
Additional Authority Erosion Factor	Ineffective Authority Delegation	NOAA, NASA, and the DoD shared authority on the EXCOM. However, the EXCOM did not meet frequently enough or with a full quorum of members to make the decisions that the organization required of it.
	Ineffective Authority Delegation	Several councils of user representatives interfaced with the NPOESS program and were authorized to approve changes to the program's requirements or capabilities. However, these user groups lacked an authority structure that enabled them to make decisions: essentially, because no user had authority over the others, any user could veto decisions. As a result, the user groups could not agree on any decisions to reduce the program's technical baseline even though the program's costs were growing and its budget was fixed.
	Contract Structure	NPOESS used performance-based specifications that levied requirements primarily on the system's final data products and that allowed the contractors to derive intermediate design and process requirements. When the government's needs changed and it wished to levy requirements on designs or processes, it had to issue contract change orders, which were often costly.
Authority Eroding Misalignment	Authority & Expertise	The NASA engineers that supported NPP instrument development had technical expertise but no official authority over the instruments. As a result, NASA often used its expertise to dispute the IPO's decisions. This eroded the IPO's authority over its contractors and slowed the contractors' progress as they attempted to satisfy all of their government customers.
	Responsibility & Budget	Although the NPP instruments executed NASA's climate science mission, their development was funded entirely by NOAA and the DoD. The misalignment of responsibility and budget caused NASA to inappropriately weigh performance and risk vice cost when making decisions regarding NPP's instrument development.
Misalignment	Authority & Responsibility	The Tri-agency Steering Committee (TSC) was responsible to the EXCOM for overseeing the program; however, they had no authority to make decisions on the EXCOM's behalf. As a result, the TSC induced additional oversight but did not improve the program's performance because they had no authority to direct its activities.
	Authority & Responsibility	Although the agencies shared authority over the NPOESS prorgam, each agency executed its oversight responsibility individually; as a result, the program's oversight costs increased three-fold. Despite this increased oversight, because the agencies shared authority, no one agency could unilaterally respond to the issues that it uncovered during its independent oversight of the program.
	Authority & Responsibility	NASA was responsible for NPP's missions and by extension, for the instruments that collected data in support of those missions. Despite this responsibility, the IPO--not NASA--held authority over the program's contractors. In order to execute its responsibility for NPP's instruments, NASA engineers often engaged with IPO engineers and informally influenced the instrument development process. When NASA engineers were unable to influence IPO managers, they raised issues to NASA management, which ultimately discussed the issues at the EXCOM: the only component in the organization with authority over the rest. Elevating issues to the EXCOM took a long time, as did waiting for the EXCOM to convene and to actually make a decision.

and ground system. As with the technical architectures, we identify the complexity mechanisms contained in each organizational DSM by examining qualitative data within epoch data. Table II provides an example of this data from Epoch D and illustrates how we connected qualitative data to the complexity mechanisms contained in the DSMs; for data from other epochs, refer to Dwyer [2015].

4.5. Step 5: Calculate the Organizational Complexity Metric

NPOESS organizational complexity is calculated using the process described in Section 3. In addition to the authority erosion factors noted previously, several case-specific authority erosion factors—including limited contract structure and ineffective authority delegation—were included in the NPOESS DSM_{O}. *Limited contractual authority* refers to the limitations that contracts can place on the government's authority to direct the contractors' activities without incurring the additional cost of a contract modification. On NPOESS, the government used performance-based contracts that limited its ability to direct vendors' design and development activities without modifying their contracts at an additional cost. *Ineffective authority delegation* refers to organizational components—like the NPOESS EXCOM—that held authority but that often failed to execute that authority by making timely decisions.

4.6. Step 6: Observe the Evolution of Complexity over Time

Figures 5 and 6 illustrate how we applied our framework to observe the evolution of complexity on the NPOESS program. First, Figure 5 depicts a miniature version of each epoch's DSM_T and its corresponding L_{cc}. This figure suggests that even when cost estimates account for complexity, the program's initial technical architecture was less costly than the separate POES and DMSP systems. However, as shown

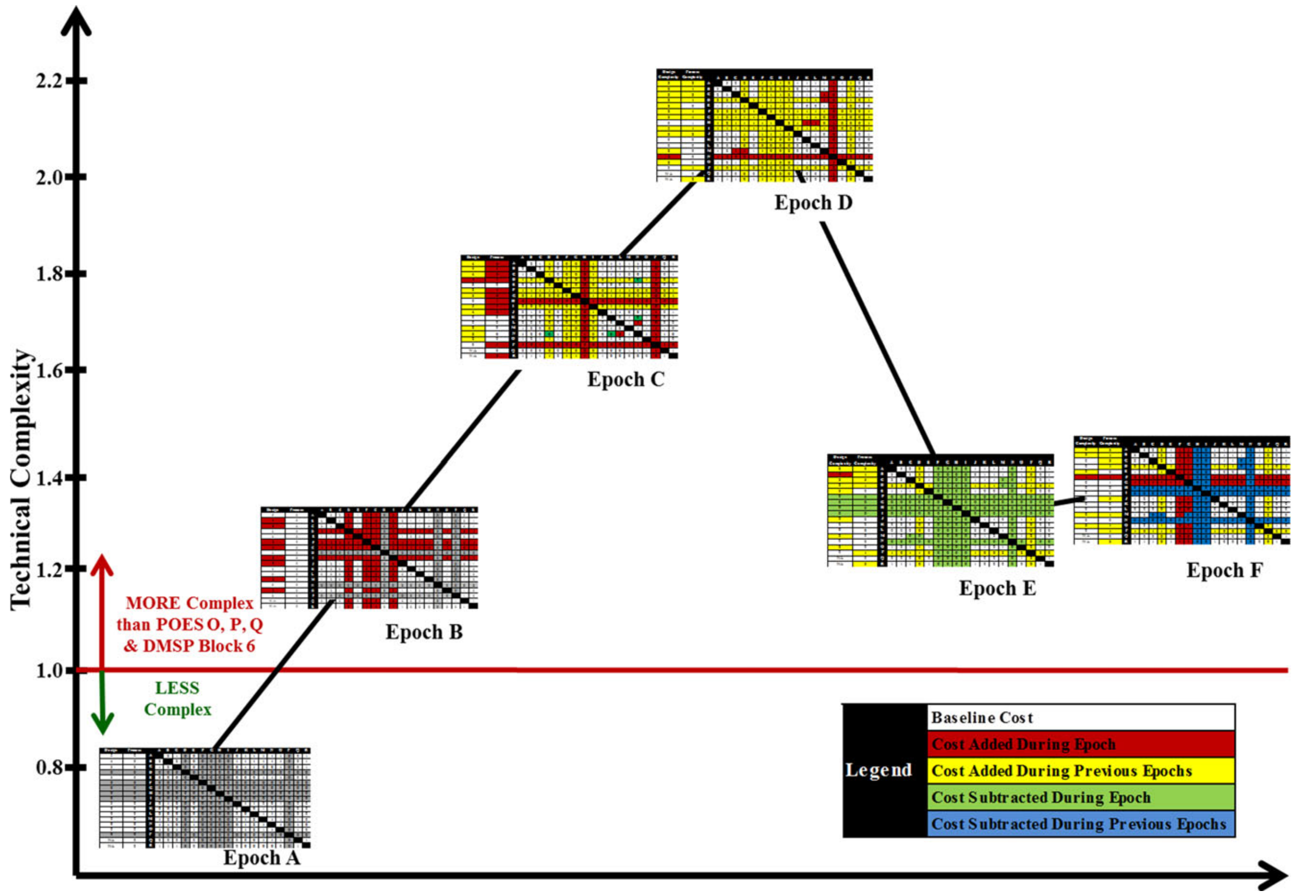


Figure 5. Evolution of NPOESS's technical architecture.

in Figure 7, L_{cc} increased significantly after Epoch A, while the program's cost estimates continued to remain low. From this comparison, we can draw several conclusions:

- Complexity was injected into the NPOESS technical architecture during the program's earliest epochs.
- Despite this increase in complexity, the program's costs estimates continued to remain low until after its prime contractor was selected in Epoch D.
- After the prime was selected, the program's cost estimates increased substantially while its technical complexity continued increasing on the trajectory that it had followed during previous epochs.
- The Nunn-McCurdy certification in Epoch E effectively reduced technical complexity; however, the program's cost estimates continued to grow.

The relationship between the technical complexity metric and the program's cost estimates (shown in Figure 7) motivates several conclusions about cost growth on the NPOESS program:

- During the early epochs, the NPOESS program underestimated and undermanaged the complexity contained within its technical architecture.
- After the program selected a prime contractor to serve as its system integrator, it was able to more actively manage its technical complexity. Once the prime contractor began managing the technical system and realized its inherent complexity, its cost estimates increased.
- During the program's final epochs, its cost had both a technical and a nontechnical component, since the program's cost estimates exceeded the costs predicted by our technical complexity metric.

These two cost components suggest that the NPOESS organization contributed to cost growth in two distinct ways: by enabling technical cost growth and by inducing nontechnical cost growth. Qualitative within-epoch data indicate that organizational complexity enabled cost growth by hindering the program's ability to recognize and to actively manage the complexity contained in its system. Organizational complexity also induced cost growth by reducing the program's efficiency and effectiveness.

Like technical complexity, we observe that organizational complexity also evolved throughout NPOESS. Figure 6 illustrates this evolution and using its global perspective on the program, we are able to draw several conclusions:

- The joint NPOESS organization during Epoch A was less complex than two separate POES and DMSP organizations.
- Despite this initial decrease in complexity, the decision to delay selecting a prime contractor increased orga-

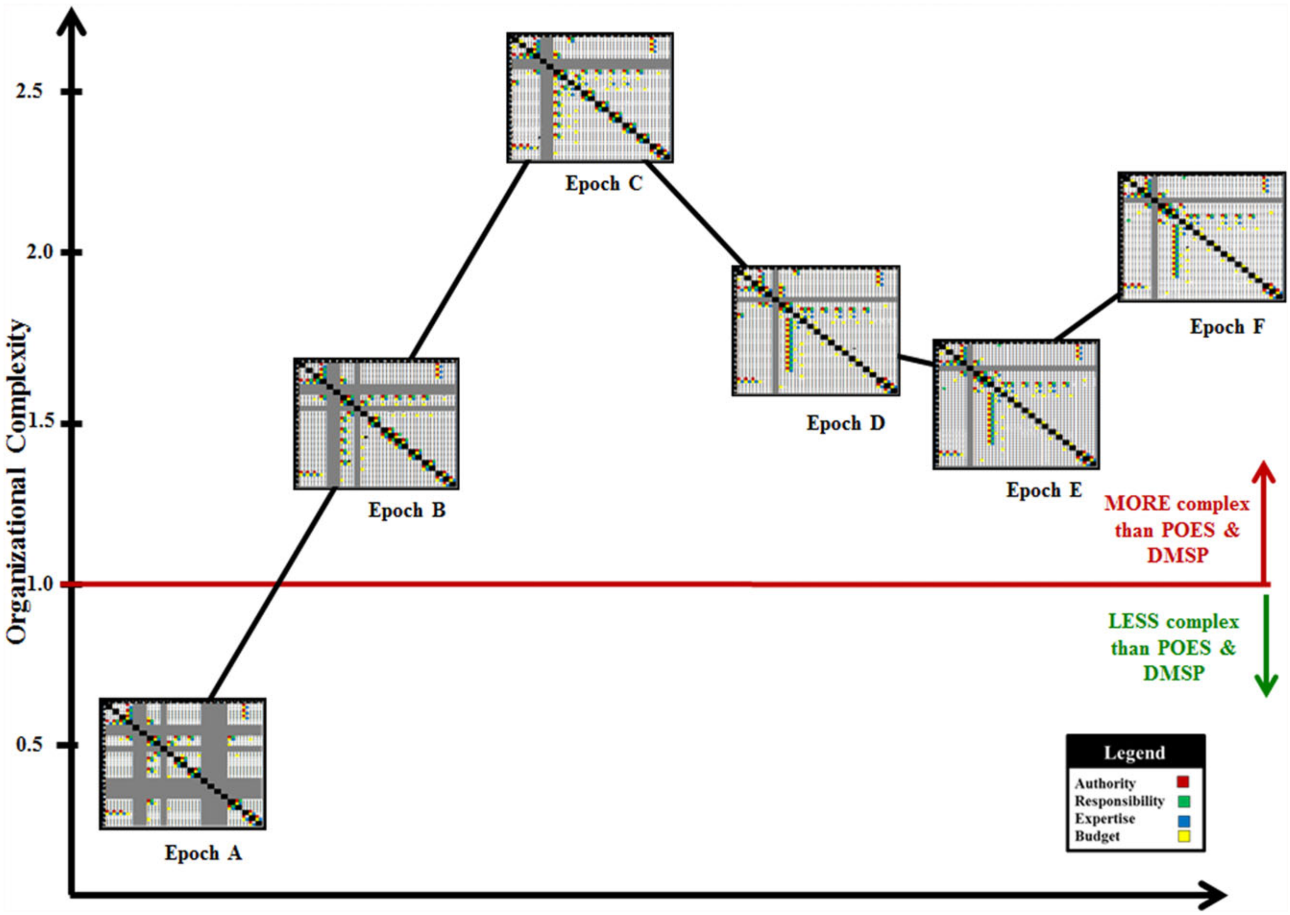


Figure 6. Evolution of NPOESS's organizational architecture.

nizational complexity because it assigned total system responsibility to a prime contractor but did not award that contractor authority over the system until Epoch D.

- Once the prime contractor was selected, authority and responsibility were better aligned and organizational complexity decreased.
- Despite the decrease in complexity after the prime was selected, complexity remained high for the rest of the program because the addition of NPP misaligned NASA's responsibility for NPP's climate science mission with its authority over the instruments that executed that mission.
- The addition of NPP misaligned authority and responsibility throughout the entire organizational architecture. As a result, when the organizational components shared responsibility but not authority for a decision, that decision had to be elevated up the organizational hierarchy to management that had authority over both of the responsible components. The only managing component that had authority over both the NPOESS IPO and NPP was the EXCOM; as a result, the EXCOM was often tasked to make decisions that affected both programs.
- Reforms during the Nunn-McCurdy certification process in Epoch E reduced organizational complexity by aligning responsibility and authority under a single PEO; however, because these reforms did not align authority and expertise and budget and responsibility between the NPOESS and NPP programs, much of the organization's complexity remained.
- Finally, just prior to the program's cancellation, the PEO's authority was eroded and organizational complexity increased once again.

Each of the above conclusions is consistent with the qualitative data that were used to support each DSM's construction and each complexity metric's calculation.

4.7. Case Study Conclusions

Applying the framework to study the NPOESS program enabled us to revise the acquisition community's understanding of the program's history. First, we observed that after Epoch B, the NPOESS system was more complex and costly than both the POES and DMSP systems. However, contrary to blame that has been levied on the program's prime contractor in the past [DoC 2006, IRT, 2009], we observed that complexity was inherent in the NPOESS system well before the program selected its prime and that this complexity had a more significant impact on cost than did any mismanagement

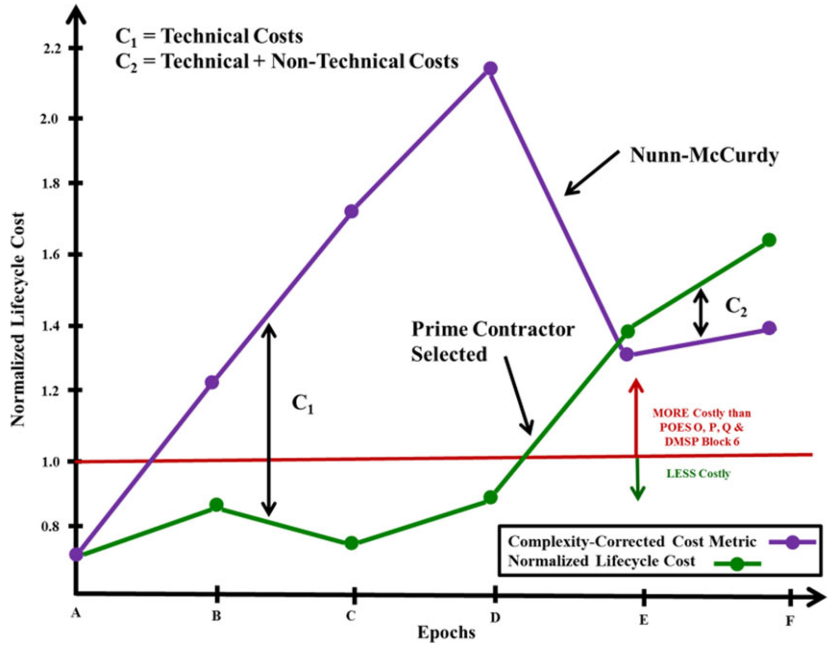


Figure 7. Comparison of technical complexity metric and normalized life cycle cost estimate.

by the prime contractor. Furthermore, the organizational complexity induced by delaying the prime contractor's selection hindered the program's ability to accurately estimate and to aggressively manage its costs. We observed this relationship between organizational and technical complexity during our within-epoch qualitative data analysis and then observed it using our quantitative framework; as shown in Figure 6, without a prime contractor, organizational complexity increased because authority and responsibility were misaligned. However, after the prime contractor was selected in Epoch D, organizational complexity decreased.

Also, contrary to past conclusions [GAO 2005, 2007; IRT 2009], we assessed the EXCOM to have had a less significant cost impact than did the complexity of the organizational architecture *beneath* it. Specifically, we observed that when authority and responsibility were misaligned between two components in the organization, those components were forced to elevate decisions to another component that had authority to make decisions which affected both of them. Because the EXCOM was the single component with authority over an organization where authority and responsibility were misaligned, the triagency board was forced to play a larger role in program execution than was ever intended. Thus, we conclude that it was the complexity of the organizational architecture beneath the EXCOM, rather than the EXCOM itself, which most significantly affected the program's costs.

Finally, the framework allowed us to observe how significantly the program's technical and organizational architectures evolved throughout its life cycle. This perspective motivates the conclusion that any analysis of program cost growth that does not consider the program's architectural evolution is incomplete. Furthermore, without this perspective, we suggest that it was impossible for past studies to uncover the underlying mechanisms that drove cost growth on NPOESS and to use that understanding to architect less costly programs in the future.

5. CONCLUSIONS AND FUTURE WORK

This paper presented a framework for studying cost growth on complex acquisition programs and demonstrated the utility of that framework by applying it to a case study of the NPOESS program. In studying NPOESS, we demonstrated that a program's architectures, its technical and organizational complexity, and its cost are all dynamic properties. We also illustrated how, by studying programs using the simultaneously in-depth and longitudinal perspective enabled by the framework, one can gain new insights into the specific mechanisms that affect a program's cost as well as those mechanisms' impacts relative to one another and to other events on the program. Although our case study focused on only one program, by defining our DSMs and complexity metrics generally, we hope that the framework can be applied to study other complex acquisition programs in future.

We also hope that future research will expand upon our framework by refining our definitions of complexity and the metrics that we use to assess it. Currently, our metrics are tools that aid *observation* rather than *prediction*: the metrics are useful because they enable the researcher to observe the dynamics of complexity throughout a program's lifetime. Since cost estimates are uncertain at different points a system's life cycle and often grow throughout, at this point in time, it is impossible to calibrate our metrics; however, if future research uses our framework to track past and present acquisition programs, it may be possible to calibrate the metrics and use them as predictive tools in the future. Despite this opportunity for future refinement, the most important strength of our proposed framework is that it combines detailed information about a program's activities during each epoch with a global view of the program's evolution across epochs: these dual perspectives can be used to illuminate the relationship between microscopic within-epoch behavior and macroscopic trends that span across epochs. Using this new perspective, future researchers will be better equipped to identify strategies to manage complexity mechanisms within epochs and to more effectively prevent cost growth across them.

REFERENCES

F. Alibay and N.J. Strange, Trade space evaluation of multi-mission architectures for the exploration of Europa, Paper presented IEEE Aerospace Conference, Big Sky, MT, 2013, Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6497199, accessed August 18, 2014.

American National Standards Institute/American Institute of Aeronautics and Astronautics (AIAA), Mass properties control for space vehicles, missiles, and launch vehicles (ANSI/AIAA R-020A-1999), AIAA, Washington, DC, 1999.

N.R. Augustine, *Augustine's laws*, American Institute of Aeronautics and Astronautics, Reston, VA, 1997.

N. Azizian, et al., A framework for evaluating technology readiness, system quality, and program performance of U.S. DoD acquisitions, Syst Eng 14(4) (2011), 410–426.

C. Baldwin and K.B. Clark, *Design rules: Volume 1, the power of modularity*, The MIT Press, Cambridge, MA, 2000.

D.A. Bearden, A complexity-based risk assessment of low-cost planetary missions: When is a mission too fast and too cheap? Acta Astronaut 52(2–6) (2003), 271–279.

M. Bennett, *Options for modernizing military weather satellites*, Congressional Budget Office, Washington, DC, 2012, Retrieved from http://www.cbo.gov/sites/default/files/cbofiles/attachments/09-20-WeatherSatellites.pdf.

I. Blickstein, *Root cause analyses of Nunn-McCurdy breaches, Vol. 1*, RAND Corporation, Santa Monica, CA, 2011.

T. Brady and D. Nightingale, "NASA Mars Pathfinder technology readiness," in S. Eppinger and T. Browning (Editors), *Design structure matrix methods and applications*, The MIT Press, Cambridge, MA, 2012, pp. 49–53.

M.M. Brown, Acquisition risks in a world of joint capabilities: A study of interdependency complexity, Paper presented at the 11th Annual Acquisition Research Symposium, Monterrey, CA, 2014, Retrieved from http://www.acquisitionresearch.net/files/FY2014/NPS-AM-14-C11P13R03-053.pdf, accessed August 18, 2014.

M.M. Brown, R.M. Flowe, and S.P. Hamel, The acquisition of joint programs: The implications of interdependencies, CROSSTALK: J Defense Softw Eng (2007), Retrieved from http://www.dtic.mil/dtic/tr/fulltext/u2/a487683.pdf, accessed August 18, 2014.

R.L. Coleman, J.R. Summerville, and M.E. Dameron, The relationship between cost and schedule growth, *Acquis Rev Quart* (2003), Retrieved from http://www.dau.mil/pubscats/pubscats/AR%20Journal/arq2003/ColemanSP3.pdf, accessed August 18, 2014.

Committee on Cost Growth in NASA Earth and Space Science Missions, Controlling cost growth of NASA earth and space science missions, The National Research Council, Washington, DC, 2010, Retrieved from http://www.nap.edu/openbook.php?record_id=12946, accessed August 18, 2014.

E.F. Crawley, B.G. Cameron, and D. Selva, *System architecture: Technology strategy and product development for complex systems*, Pearson, Upper Saddle River, NJ, 2015.

Defense Science Board (DSB), Air Force Science Advisory Board Joint Task Force, *Acquisition of national security space programs*, Office of the Under Secretary of Defense for Acquisition, Technology, and Logistics, Washington, DC, 2003, Retrieved from http://www.acq.osd.mil/dsb/reports/ADA429180.pdf, accessed August 18, 2014.

A. Downs, *Inside bureaucracy*, Little, Brown, and Company, Boston, 1972.

G.F. Dubos and J.H. Saleh, Spacecraft technology portfolio: Probabilistic modeling and implications for responsiveness and schedule slippage, Acta Astronaut 68(7–8) (2011), 1126–1146.

M. Dwyer, The cost of jointness: Insights from environmental monitoring systems in low-earth orbit, Doctoral dissertation, 2015, Retrieved from http://systemarchitect.mit.edu/docs/dwyer14e.pdf, accessed February 8, 2015.

M. Dwyer and Z. Szajnfarber, A framework to assess the impacts of jointness, Paper presented at CESUN, Hoboken, NJ, 2014, Retrieved from http://systemarchitect.mit.edu/docs/dwyer14b.pdf, accessed February 8, 2015.

M. Dwyer, Z. Szajnfarber, B. Cameron, M. Bradford, and E. Crawley, The cost impacts of jointness: Insights from the NPOESS program, Paper presented at the AFCEA Acquisition Research Symposium, Monterey, CA, 2014, Retrieved from http://systemarchitect.mit.edu/docs/dwyer14a.pdf, accessed February 8, 2015.

K.M. Eisenhardt and M.A. Graebner, Theory building from cases: Opportunities and challenges, Acad Manage J 50(1) (2007), 25–32.

S. Eppinger and T. Browning, *Design structure matrix methods and applications (Engineering Systems)*, The MIT Press, Cambridge, MA, 2012.

Final Phase 0 Cost and Operational Benefits Requirements Analysis Report (COBRA), January 31, 2000.

J. Galbraith, *Designing complex organizations*, Addison-Wesley Publishing Company, Reading, MA, 1974.

A.L. George and A. Bennet, *Case studies and theory development in the social sciences*, The MIT Press, Cambridge, MA, 2005.

A. Gore, From red tape to results: Creating a government that works better and costs less, A Report of the National Performance Review, 1993, Retrieved from http://govinfo.library.unt.edu/npr/library/nprrpt/annrpt/pdf/com01.pdf, accessed August 9, 2014.

R.M. Henderson and K.B. Clark, Architectural innovation: The reconfiguration of existing product technologies and the failure of established firms, Admin Sci Quart 35(1) (1990), 9–30.

J. Hofbauer, et al., *Cost and time overruns for major defense acquisition programs: An annotated brief*, Center for Strategic and International Studies, Washington, DC, 2011, Retrieved from http://csis.org/files/publication/110517_DIIG_MDAP_overruns.pdf.

The White House, Restructuring the national polar-orbiting operational environmental satellite system, 2010, Retrieved from http://www.whitehouse.gov/sites/default/files/npoess_decision_fact_sheet_2-1-10.pdf, accessed August 7, 2014.

D. Katz, et al. The relationship of technology and design maturity to DoD weapon system cost change and schedule change during engineering and manufacturing development, Syst Eng 18(1) (2015).

D. Katz and K. Robert, *Organizations and the systems concept: The social psychology of organizations*, J. Shafritz and P. Whitbeck (Editors), Moore Publishing, Oak Park, IL, 1978.

A. Langley, Strategies for theorizing from process data, Acad Manage Rev 24(4) (1999), 691–710.

W. Larson and J.R. Wertz (Editors), *Space mission engineering: The new SMAD*, 1st edition, Microcosm, Hawthorne, CA, 2011.

C.J. Leising, B. Sherwood, M. Adler, R. Wessen, and F.M. Naderi, Improvements in JPL's mission formulation process, Paper presented at IEEE Aerospace Conference, Big Sky, MT, 2010, Retrieved from http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber = 5446881, accessed August 18, 2014.

C.J. Leising, R. Wessen, R. Ellyin, L. Rosenberg, and A. Leising, Spacecraft complexity subfactors and implications for future cost growth, Paper presented at IEEE Aerospace Conference, Big Sky, MT, 2013, Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber = 6497159&tag = 1, accessed August 18, 2014.

M.W. Maier and E. Rechtin, *The art of systems architecting*, 2nd edition, CLC Press LLC, Boca Raton, FL, 2000.

P. Malone and L. Wolfarth, Measuring system complexity to support development cost estimates, Paper presented at the IEEE Aerospace Conference, Big Sky, MT, 2013, Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber = 6496853&tag = 1, accessed August 18, 2014.

M. Meyer and A. Lehnerd, *The power of product platforms*, The Free Press, New York, 1997.

D.A. Nadler and M.L. Tushman, *Competing by design: The power of organizational architecture*, Oxford University Press, New York, 1997.

NASA Office of the Inspector General, NASA's challenges to meeting cost, schedule, and performance goals, Report No. IF-12-021, 2012, Retrieved from http://oig.nasa.gov/audits/reports/FY12/IG-12-021.pdf, accessed August 18, 2014.

D. Nigg, R. O'Brian, and R. Abbott, FireSat revisted: Investigations in tradespace exploration, Paper presented at AIAA Space, Long Beach, CA, 2011, Retrieved from http://arc.aiaa.org/doi/abs/10.2514/6.2011-7272, accessed August 22, 2014.

E.H. Nowinski and R.J. Kohler, A view of how we manage: The lost art of program management in the intelligence community, *The Central Intelligence Agency*, 2007, Retrieved from https://www.cia.gov/library/center-for-the-study-of-intelligence/csi-publications/csi-studies/studies/vol50no2/html_files/Program_Management_4.htm, accessed August 18, 2014.

NPOESS Independent Review Team (IRT), Final Report, 2009, Retrieved from http://www.spacepolicyonline.com/pages/images/stories/tom_young_npoess_report.pdf, accessed August 9, 2014.

Office of the Under Secretary of Defense (OSD), Acquisition, Technology, and Logistics, Performance of the defense acquisition system, 2013 annual report, 2013, Retrieved from http://www.defense.gov/pubs/Performance-of-Defense-Acquisition-System-2014.pdf, accessed August 18, 2014.

A. Rasmussen and R. Tsugawa, Cost effective applications of constellation architectures of large, medium, and small satellites, AIAA Paper 97-3950, American Institute of Aeronautics & Astronautics, 1997, retrieved August 18, 2014.

D. Selva, *Rule-based system architecting of earth observation satellite systems*, Doctoral dissertation, 2012, Retrieved from http://systemarchitect.mit.edu/docs/selva12c.pdf.

S.A. Sheard and A. Mostashari, A complexity typology for systems engineering, INCOSE, 2009, Retrieved from https://www.researchgate.net/publication/228741550_A_Complexity_Typology_for_Systems_Engineering, accessed August 18, 2014.

H.A. Simon, The architecture of complexity, Proc Amer Philos Soc 106(6) (1962), 467–482.

K. Sinha, Structural complexity quantification for engineered complex systems and implications on system architecture and design, Doctoral dissertation, 2013, Retrieved from http://web.mit.edu/deweck/Public/AVM/Final-Report-21807_S8-Thesis.pdf, accessed February 8, 2015.

K. Sinha and O. de Weck, Structural complexity quantification for engineered complex systems and implications on system architecture and design, Proceedings of the AMSE 2013 International Design Engineering Technical Conference. Paper presented at 39th Design Automation Conference, Portland, OR, 2013, Retrieved from http://proceedings.asmedigitalcollection.asme.org/proceeding.aspx?articleid=1830329, accessed August 18, 2014.

R. Smaling and O. de Weck, Assessing risks and opportunities of technology infusion in system design. Syst Eng 10(1) (2007), 1–25.

M.E. Sosa, T. Browning, and J. Mihm, Studying the dynamics of the architecture of software products, Proceedings of the AMSE 2007 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference. Paper presented at IDTE/CIE, Las Vegas, NV, 2007, Retrieved from http://www.insead.edu/facultyresearch/faculty/personal/msosa/research/documents/sosa_detc2007-34761_v2.pdf, accessed August 18, 2014.

M. Sosa, S. Eppinger, and C. Rowles, Are your engineers talking to one another when they should? *Harvard Bus Rev* (2007), Retrieved from https://hbr.org/2007/11/are-your-engineers-talking-to-one-another-when-they-should, accessed February 8, 2015.

E.S. Suh, M.R. Furst, K.J. Mihalyov, and O. de Weck, Technology infusion for complex systems: A framework and case study, Syst Eng 13(2) (2009), 186–203.

J.M. Sussman, R.S. Dodder, J.B. McConnell, A. Mostashari, and S. Sgouridis, *The "CLIOS process": A user's guide*, MIT, Cambridge, MA, 2009, Retrieved from http://esd.mit.edu/Faculty_Pages/sussman/CLIOS-PROCESS.pdf, accessed August. 18, 2014.

The Aerospace Corporation, NPOESS Lessons evaluation: Executive summary, 2010 (Publication No. ATR-2011(5558)-1), Retrieved from http://www.osd.noaa.gov/download/NPOESSAerospaceReport.pdf, accessed August 9, 2014.

J.D. Thompson, *Organizations in action: Social science bases of administrative theory*, Transaction Publishers, New Brunswick, NJ, 2003.

U.S. Department of Commerce (DoC), Office of Inspector General, Poor Management Oversight and Ineffective Incentives Leave NPOESS Program Well Over Budget and Behind Schedule, Audit Report No. OIG-17794-6-0001, 2006, Retrieved from http://www.oig.doc.gov/OIGPublications/OIG-17794.pdf, accessed August 9, 2014.

U.S. Government Accountability Office (GAO), *NASA program costs: Space missions require substantially more funding than* *initially estimated* (Publication No.: GAO/NSIAD-93-97), 1992, Retrieved from http://www.gao.gov/products/GAO/NSIAD-93-97, accessed August 18, 2014.

U.S. Government Accountability Office (GAO), *Polar-orbiting environmental satellites: Status, plans, and future data management challenges*, Statement of Linda D. Koontz, Director, Information Management Issues (Publication No. GAO-02-684T), 2002, Retrieved from http://www.gao.gov/assets/110/109529.html.

U.S. Government Accountability Office, *Military space operations: Common problems and their effects on satellite and related acquisitions* (Publication No.: GAO-03-825R), 2003, Retrieved from http://www.gao.gov/new.items/d03825r.pdf, accessed August 18, 2014.

U.S. Government Accountability Office (GAO), *Polar-orbiting environmental satellites*: Information on program cost and schedule changes (Publication No. GAO-04-1054), 2004, Retrieved from http://www.gpo.gov/fdsys/pkg/GAOREPORTS-GAO-04-1054/html/GAOREPORTS-GAO-04-1054.htm.

U.S. Government Accountability Office (GAO), Testimony before the committee on science of the house of representatives. Technical problems, cost increases, and schedule delays trigger need for difficult trade-off decisions, Statement of David A. Powner (Publication No. GAO-06-249T), 2005, Retrieved from http://www.gao.gov/assets/120/112634.pdf.

U.S. Government Accountability Office (GAO), Testimony before the subcommittee on disaster prevention and prediction, committee on commerce, science, and transportation of the U.S. Senate. Cost increases trigger review and place program's direction on hold (Publication No. GAO-06-573T), 2006a, Retrieved from http://www.gao.gov/assets/120/113254.pdf.

U.S. Government Accountability Office (GAO), *Space acquisitions: Improvements needed in space systems acquisitions and keys to achieving them* (Publication No.: GAO-06-626T), 2006b, Retrieved from http://www.gao.gov/products/GAO-06-626T, accessed August 18, 2014.

U.S. Government Accountability Office (GAO), *Polar-orbiting operational environmental satellites: Restructuring is under way, but technical challenges and risks remain* (Publication No. GAO-07-498), 2007a, Retrieved from http://www.gao.gov/assets/260/259758.html, accessed February 8, 2015.

U.S. Government Accountability Office (GAO), *Space acquisitions: Actions needed to expand and sustain use of best practices* (Publication No.: GAO-07-730T), 2007b, Retrieved from http://www.gao.gov/products/GAO-07-730T, accessed August 18, 2014.

U.S. Government Accountability Office (GAO), *Defense acquisitions: Assessments of selected weapon programs* (Publication No. GAO-09-326SP), 2009a, Retrieved from http://www.ndia.org/Advocacy/PolicyPublicationsResources/Documents/GAO%20Assessment%20of%20Selected%20Weapons%203.091.pdf, accessed January 25, 2015.

U.S. Government Accountability Office (GAO), *Polar-orbiting environmental satellites: With costs increasing and data continuity at risk, improvements needed in tri-agency decision making* (Publication No. GAO-09-564), 2009b, Retrieved from http://www.gao.gov/new.items/d09772t.pdf.

U.S. Government Accountability Office (GAO), *Agencies must act quickly to address risks that jeopardize the continuity of weather and climate data* (Publication No. GAO-10-558), 2010, Retrieved from http://www.gao.gov/assets/310/304904.pdf.

U.S. Government Accountability Office (GAO), *Space acquisitions: DoD delivering new generations of satellite but space system acquisition challenges remain* (Publication No.: GAO-11-590T), 2011, Retrieved from http://www.gao.gov/products/GAO-11-590T, accessed August 18, 2014.

U.S. Government Accountability Office (GAO), *DoD faces challenges in fully realizing benefits of space acquisition improvements* (Publication No.: GAO-12-563T), 2012a, Retrieved from http://www.gao.gov/products/GAO-12-563T, accessed August 18, 2014.

U.S. Government Accountability Office (GAO), *NASA: Assessment of selected large-scale projects* (Publication No.: GAO-12-207SP), 2012b, Retrieved from http://www.gao.gov/products/GAO-12-207SP, accessed August 18, 2014.

U.S. Government Accountability Office (GAO), *Changing requirements, technical issues, and looming data gaps require focused attention* (Publication No. GAO-12-604), 2012c, Retrieved from http://www.gao.gov/assets/600/591643.pdf, accessed August 8, 2014.

J.Q. Wilson, *Bureaucracy: What government agencies do and why they do it*, Basic Books, New York, 1989.

M. Weber, "The essentials of bureaucratic organization: An ideal-type construction," in R.K. Merton, A.P. Gray, B. Hockey, and H.C. Selvin (Editors), *Reader in bureaucracy*, The Free Press, Glencoe, IL, 1952.

R.K. Yin, *Case study research: Design and methods,* 4th edition, SAGE, Los Angeles, 2009.

O. Younossi, et al., *Is weapon system cost growth increasing?*, The RAND Corporation, Santa Monica, CA, 2007.

O. Younossi, *Improving the cost estimation of space systems*, The RAND Corporation, Santa Monica, CA, 2008.

Morgan Dwyer completed this work as a Ph.D. candidate in Technology, Management, and Policy at MIT. Prior to MIT, Morgan worked as a systems engineer at Boeing Space and Intelligence Systems and held internships at Fermi-lab, NASA Marshall Spaceflight Center, and the Space Telescope Science Institute. Morgan received a B.S. in Astronomy and Physics from Yale, an M.S. in Aeronautics and Astronautics from Stanford, and her Ph.D. from MIT focused on understanding the costs induced by acquiring systems jointly. She currently works as a systems engineer in the Washington DC metro area.

Bruce Cameron is a Lecturer in Engineering Systems at MIT and a consultant on platform strategies. At MIT, Bruce ran the MIT Commonality Study, a 16-firm investigation of platforming returns. His current clients include Fortune 500 firms in high tech, aerospace, transportation, and consumer goods. He holds an undergraduate degree from the University of Toronto and graduate degrees from MIT.

Zoe Szajnfarber is an Assistant Professor of Engineering Management and Systems Engineering at the George Washington University. Her research seeks to understand the fundamental dynamics of innovation in technology-intensive organizations. She received her bachelor's degree in Engineering Science from the University of Toronto, a dual master's degree in Aeronautics and Astronautics and Technology Policy, and a Ph.D. in Engineering Systems, all from MIT.