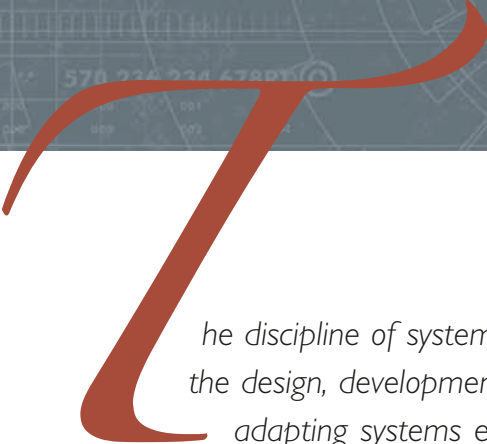


# Systems Engineering for Complex Information Systems in a Federated, Rapid Development Environment

Charles R. Spaulding, W. Scott Gibson, Stephen F. Schreurs,  
Duane L. Linsenhardt, and Antonio DeSimone



The discipline of systems engineering has tremendous potential to improve the design, development, and fielding of complex information systems, but adapting systems engineering methods to the real-world challenges of large-scale, rapidly adaptable information systems is a daunting task. The scale can challenge traditional systems engineering approaches, but the problem of scale alone does not drive the need for a new look at how to apply systems engineering methods. Three challenges drive the systems engineering approach described here: fielding capabilities quickly; evolving capabilities to address rapidly changing mission needs; and, finally, building the system from components provided by a heterogeneous mix of solution providers who cooperate in a loose federation. APL, with a DoD sponsor, has implemented a phased, iterative value-added systems engineering strategy that provides the sponsor a tailored systems engineering approach to support critical capability development and deployment.

## INTRODUCTION

Information management and communications in today's world have been shaped by the technological, economic, social, and political changes of the last quarter of the 20th century. All aspects of modern life have been touched by the continuing decentralization and diversity of information technologies. Communications and information technologies that were once scarce, expensive, and carefully managed are now common, cheap, and proliferating uncontrollably. Those trends

have driven a reassessment<sup>1</sup> of DoD's processes for acquisition of such systems; the changes also drive changes to the practice of systems engineering (SE) for complex information systems.

APL is engaged with a DoD sponsor as the SE trusted agent carrying out transformational capability development and fielding efforts for a system referred to hereafter as BT, for BIGTERRITORY. Major elements of BT have the characteristics of a complex information

system, similar to other systems relying on modern communications and information technologies. Inherently, these systems are characterized by problems of scale, but are further characterized by rich interconnections that generate behaviors that are difficult to characterize. Furthermore, BT, like many other systems, operates in an environment of changing demands and evolving technologies. Program constraints preclude discussion of detailed technical characteristics of the system; instead, this article focuses on the engineering challenges and the approach taken to address those challenges. APL worked with the sponsor to develop and implement a value-added SE (VASE) approach that improved the design, development, and fielding of the system.

Three challenges drive the SE approach described in this article: fielding capabilities quickly; evolving capabilities to address rapidly changing mission needs; and, finally, building the system from components provided by a heterogeneous mix of solution providers who cooperate in a loose federation. APL, with a DoD sponsor, has implemented an approach to SE that addresses these three challenges. The first two challenges, speed and change, reflect the nature of the sponsor's environment and the environment surrounding many complex information systems. The third challenge, operating in a federated solution provider environment, reflects the need for SE to accommodate parallel technology explorations and deliver a nimble response in the face of rapidly changing needs. The environment and the federated structure violate many of the underlying tenets of traditional SE approaches: stable requirements, *a priori* decomposition of requirements, centralized requirements management, and fixed development milestones. In addition, the variety of software development models used by solution providers improves the effectiveness of the development teams but also adds to integration complexity. APL, with the DoD sponsor, has identified, appropriately adapted, and applied VASE processes and practices to meet sponsor needs in such an environment. This article discusses the underlying technical and cultural challenges inherent in the customer's federated solution provider environment and characterizes constraints that inhibit some engineering approaches. It further describes the phased, iterative VASE strategy that APL and the sponsor are pursuing to refine and implement a tailored SE approach to support critical capability development and deployment.

Before discussing this specific case, as context for the discussion, the next sections describe some SE successes and the challenges to SE disciplines presented by complex information systems.

## SE Successes

SE as a discipline evolved as large systems comprising diverse technologies came to dominate the 20th

century. A prime example of successful application of SE is the national telephone network built in the United States during the middle of the last century. The network was built in the heyday of the Bell System, roughly from the Telecommunications Act of 1934, which established a regulated monopoly for telephone services in the United States, to the breakup of the Bell System by the divestiture agreement of 1984. The Bell System was assembled and operated with a clear mission of providing universal telephone service. The Bell System built or acquired most of the technologies that made up the national telephone network; therefore, it was able to implement SE in a disciplined way under the control of a central authority. Before the effects of divestiture affected its operation, the Bell System documented its engineering practices across the system in a volume called *Engineering and Operations in the Bell System*.<sup>2</sup> According to this document, systems engineers “planned the nationwide telephone network and its operations. They considered the entire network rather than just one part.” The Bell System approach relied on careful central planning. Another document from that era, *Notes on the Network*,<sup>3</sup> describes the mindset: “Because the sums invested are large and the service life of most plants involves many years, it is important that fundamental plans be made well in advance of any actual change.” The Bell System is the triumph of a unified set of engineering processes implemented over many decades through central authorities that operated over long planning cycles to develop a well-characterized set of system capabilities.

The sprawling Bell System is not the only example of SE success. Space systems, of more limited scope but of no less technical sophistication, have demonstrated successful application of SE methods.<sup>4</sup> A space system is, by its very nature, clearly bounded with well-defined interfaces to the outside world. The system must implement a clearly defined set of functions that can be decomposed *a priori* by a central SE function and then allocated to subsystems. Within the system, various parameters—size, weight, power, and so on—can in turn be allocated to subsystems to allow designers to build the buses and packages that make up the space system. A launch provides a spectacularly specific milestone for the systems engineers, designers, and developers to use for planning.

These systems have a number of attributes in common: central authority to make technical decisions, either within a corporation or via a prime contractor for an acquisition; clear boundaries within which technical decisions are made, with system interfaces under the control of the authority; complete and clearly articulated objectives up front; and, finally, slow changes in the operational environment that permit long planning cycles, allowing the systems engineer “to first define the requirements and then design the system to meet those requirements” (see Chapter 4 in Ref. 4).

SE practices appropriate to these environments emphasize the “waterfall” approach, from user requirements, to architecture, to design and implementation, to integration and testing, and finally to operational deployment. That model for SE is captured in the SE “V” commonly used to represent SE practices.<sup>5</sup> The VASE effort has adapted the disciplines of the traditional “V” to the sponsor’s environment. This article is focused on a subset of SE disciplines most significant to the success of VASE:

- **Requirements Management:** Defining the scope of the requirements process, tailoring the requirements process, integrating the requirements process into the BT environment, and identifying requirements management tools
- **Interface Management:** Defining interfaces, directing interface design, providing an interface repository, and controlling changes to the interfaces
- **Technical Planning and Control:** Tailoring the management process, establishing a management structure, and establishing monitoring methods
- **Planning for System Integration, Testing, and Operations:** Defining test cases, identifying required test data, and defining validation methodologies
- **DoD Acquisition Management:** Tailoring the DoD methodology and providing acquisition support

### SE Challenges for Complex Information Systems

From the beginning of the information age—coincident with the articulation of Moore’s law for the purpose of this discussion—technology has grown exponentially in its ability to process, store, and retrieve information. Complex information systems pose a challenge to SE practices. The demands to respond rapidly to change require a new way of doing business. As noted in a recent Defense Science Board study, “the conventional DoD acquisition process is too long and too cumbersome to fit the needs of the many systems that require continuous changes and upgrades.”<sup>1</sup> Complex information systems must respond quickly to change: changes in the environment; changes driven by markets, regulators, competition, and evolving threats and opportunities; and changes required by the diversity in sources of technologies and by changing proprietary solutions, evolving open-source software, and the emergence of important new protocols and applications. All these changes make the long, stable planning cycles that are the premise of the traditional SE approach untenable in the sponsor’s environment. The value SE adds to improving the design, development, and fielding of systems is still critical, but SE needs to adapt to the environment.

Four attributes of complex information systems challenge the application of SE disciplines:

1. **Scale:** The systems are large, in the number of sub-components and sites and in the volume, variety, and velocity of information.
2. **Interconnections:** The behavior of systems now emerges from the interactions among many components, generating behavior that is difficult to characterize.
3. **Changing Demands:** The environment in which the systems operate requires adaptation on a small timescale compared with the time to develop and deploy systems by using the waterfall methodology.
4. **Evolving Technologies:** Both hardware and protocols change rapidly in the information systems arena.

The problems of scale can stress SE practice. Technologies applied to requirements management, configuration management, and testing automation can mitigate the challenges of scale. The complexity arising from rich interconnectivity and the resulting complexity is more difficult to manage.<sup>6</sup> Rapidly changing demands and rapidly evolving technologies introduce more fundamental challenges to the application of SE practices. Rapid change requires planning on many timescales, with a mix of infrastructure development and capability development in each iteration. The approach to project planning and scheduling that relies on major milestone reviews becomes either unwieldy or ineffective.

The monolithic capability acquisition model that has been used in the past to develop and deliver system capabilities would not meet the deployment pace needed to address the complex and rapidly evolving technical environment. APL’s SE “loop,” described by Seymour and O’Driscoll in the Introduction to this issue of the *Technical Digest*, captures the activities contributing to developing systems capabilities. The loop demonstrates the spiral nature of capability development. The approach described in this article is closely related but aligned with the sponsor’s need to field capabilities rapidly. In brief, operationally useful capabilities are delivered to the field every 90 days, in rapid spins. That is too rapid to run through all activities serially; thus, the work to deliver a capability will span multiple spins. Selecting capabilities for each spin requires aligning the mission priorities with what is possible in the spin timeline. The SE effort provides the planning and coordination for spins and the necessary interlocks across development teams.

### Overview

The remainder of this article describes how SE methods have been applied in the sponsor’s environment. The article is organized in three sections. *Background and the Sponsor’s Environment* describes the engineering environment under discussion, including the broad trends that

shape the SE approach and an overview of the development program and the sponsor's general approach in fielding mission capabilities. *SE Challenges* examines key technical and organizational challenges posed by the selected development approach. The final section describes the VASE approach that APL developed with the sponsor to tailor SE methods to the BT environment with the aim to improve product quality and timeliness and reduce program and technical risks. This includes a discussion of the SE roles and processes defined within the program, with a particular focus on mission thread engineering, which proved effective in managing the complexities of the development environment.

## BACKGROUND AND THE SPONSOR'S ENVIRONMENT

As discussed in the *Introduction* to this article, SE and acquisition for complex information systems is changing in response to many broad and fundamental trends. The dominant feature of today's world, as a consequence of these trends, is an explosion in the degree of interconnections that influence behaviors and decisions. Interconnections in turn change the way the DoD thinks about major systems.<sup>7</sup> Rather than focusing narrowly on the capabilities of a single system, acquisitions now must support net-centricity as described in DoD directives DoDD 8000.01<sup>8</sup> and DoDD 8320.02,<sup>9</sup> must plan for information support per DoDD 4630.05,<sup>10</sup> and must meet Net-Ready Key Performance Parameters (NR-KPP) per Chairman of the Joint Chiefs of Staff Instruction (CJCSI) 6212.01E.<sup>11</sup>

Interconnections also change the way the sponsor executes its mission. Specialized systems tailored to narrow, well-defined missions no longer fit the environment in which the sponsor operates. Net-centric principles of interoperability and data sharing provide the agility necessary to execute in today's environment.

In practice, the move to net-centricity must overcome "stovepipe" thinking: the approach to acquisition, development, deployment, and operations that is predicated on what have now become unrealistic preconditions—clear, well-defined user needs and complete control over all aspects of the system. Stovepipe thinking has its appeal: a program simplifies its risk profile by eliminating dependencies and interconnections but, as history has shown, at a significant cost in money, time, and lost capability. APL's sponsor can afford none of those costs. The system under transformation encompasses a widely distributed, high-volume, low-latency data processing and storage environment that must be adaptive to rapidly evolving mission needs. The sponsor is responsible for several mission areas (similar to corporate profit centers) with each mission area performing specific mission threads.

APL helped the sponsor develop an approach to fielding capabilities centered on applying SE discipline to development, deployment, and operations across diverse teams. The remainder of this section outlines the project structure and the environment put in place to enable this approach.

The sponsor's dynamic and complex mission environment required an iterative and evolutionary approach to system development that is substantially different from traditional program approaches. The sponsor developed guidance on program structure and on methodologies for development, testing, and deployment all aimed at achieving agile, incremental capability improvement. The new approach focused on four key tenets:

1. **Transform the Mission:** Combine existing capabilities from multiple projects with new technology and capabilities to form a completely new way to perform the mission.
2. **Rapid Fielding:** Break the paradigm of a long lead time before fielding capabilities.
3. **Clear Decisions:** Build a project team that is government led and managed at all levels.
4. **Dedication:** Ask for an exceptional commitment from the team.

The transformational tenet is applied at all levels of capabilities. It is applied to specific capabilities within a mission area, to the mission area as a whole, and to the synergies between cooperative mission areas.

## A Federation of Solution Providers

The first tenet recognizes the need to build on existing capabilities and multiple solution providers: there is no "clean piece of paper." Functional components that implement mission capabilities exist in all states of maturity, from research prototypes to legacy systems in maintenance mode. The need to draw on all these sources of technologies to achieve the mission objectives implies a need to integrate many different software development approaches. This led to the formation of a loose federation of development teams, or solution providers, as the project structure for BT. The federation is led by an upper-level government management team that provides overarching technical direction. Each solution provider is represented by one or more government technical leads who are responsible for the development of specific capabilities within their development teams. The effort includes solution providers from more than two dozen cross-organizational software development teams.

## Mission Capability Threads

Components in themselves do not deliver a complete mission capability. A mission capability thread,

or mission thread, is a set of functions carried out by the system's functional components to deliver a complete mission capability. Thread engineering is the set of SE activities carried out to effectively design, develop, integrate, test, and deploy components from two or more solution providers to achieve end-to-end mission capabilities.

### Iterated Rapid Development and Deployment

The second tenet required a model for development, testing, and deployment that avoided the long delay in fielding new capabilities. To help reduce the time from capability concept to fielding, the sponsor chose to employ 90-day development spins, or iterations, with the goal of fielding operationally useful capabilities with each spin. Rapid, regular fielding provides quick feedback that could be factored in to planning for subsequent spins. Each spin includes development work to support multiple mission capability threads, and each thread will pass through one or more of the following activities in a spin: planning, design, development, integration, testing, deployment, or upgrade.

### Leadership Commitment and Effective Project Management

The team required a leadership structure with clear authorities and processes for decision-making. Because the BT project had very high visibility within the sponsor organization, the BT project team was government led and managed at all levels, including some of the solution provider development teams. The project team was composed of approximately 40% government personnel, with the remaining 60% contractor support. This high commitment of government resources is indicative of the importance of BT to the customer. In turn, this high degree of government involvement and visibility added to the intensity of the BT project environment.

In keeping with the fourth tenet, the BT management team asked for and received an exceptional commitment to the success of the BT project.

## SE CHALLENGES

The sponsor's approach of using a federated, rapid development model to deliver capabilities seemed, on face value, to be a logical and reasonable approach given the technical constraints. This approach, however, called for managing multiple facets simultaneously; there was no single priority that subsumed all others. That introduced greater demands on SE efforts to orchestrate multiple efforts and synthesize the results of those efforts to deliver capabilities. APL's engagement with this sponsor and broad experience in SE allowed the team to identify specific challenge areas and

potential friction points related to this technical approach. The challenges are discussed in this section. In essence, each represented a problem to be solved; APL's contribution was to determine how SE methods could be applied to each problem area to help the sponsor succeed in its mission by using available resources efficiently to improve timeliness and product quality.

The key systems engineering challenges were as follows:

- Management of cultural differences
- Orchestration of multiple solution providers
- Synthesis of integrated end-to-end capabilities
- Coordination of multiple mission scenarios
- Integration of new and existing capabilities
- Transformation of capabilities across incremental deliveries
- Management of diverse and evolving interfaces
- Tailoring acquisition documentation

The list does not represent a priority order, but demonstrates the breadth of challenges the SE team needed to meet. Each challenge is discussed more fully in the following sections.

### Management of Cultural Differences

Introducing SE methods within the development environment met varying degrees of cultural resistance, primarily due to the developers' varied experience with SE on past projects. Many of the developers were initially concerned that SE would impose an unwelcome level of burden or overhead. These misgivings prompted the BT SE team to adopt the approach of incrementally demonstrating that SE practices added value to product development and fielding with the goal of earning support from the solution provider teams. The SE team took a stepwise approach to implementing SE practices that could show early returns for adopting the practices. In particular, efforts to gain support for SE methods were focused on three key challenge areas:

- Expanding on the team's success in rapidly fielding key point solutions to emphasize building and deploying large-scale, sustainable systems
- Increasing the focus on component integration and testing required by the federated nature of the development community and the distributed system architecture
- Increasing the team's understanding of how the development and fielding activities fit within the formal DoD acquisition process

### Orchestration of Multiple Solution Providers

The reliance on federated solution providers to supply products that are integrated into a complex system presented several challenges. These challenges include:

- Relying on multiple solution providers to develop a mission capability
- Defining and managing data exchange interfaces
- Defining, negotiating, and managing requirements
- Coordinating the development and delivery of products to allow a stepwise construction of a mission capability
- Defining and implementing common processes
- Integrating and testing end-to-end mission capabilities
- Building an effective governance structure

Achieving effective system outcomes when each solution provider is focused on providing their products in a way that best meets their needs is difficult at best. The processes put in place to address these challenges had to be flexible enough to accommodate each solution provider's development model, show value to the participants and program management, be able to evolve as program needs mature or change, and keep overhead work to a minimum.

### Synthesis of Integrated End-to-End Capabilities

Synchronizing the planning, development, integration, and testing of an end-to-end mission capability is a challenge that is further complicated by the approach of using federated development teams to deliver integral components of the mission capability. Each solution provider brought knowledge that was limited to the team's domain, but the requirements for development and testing needed to be defined for the end-to-end mission capability; therefore, solution providers and systems engineers needed to collaborate to define an appropriate solution. Carrying the solution forward to implementation required a disciplined approach to interface definition and management, as well as a requirements development and management process appropriate to the federated solution provider environment.

These issues are hard enough when they are addressed for a single end-to-end mission capability, but the BT environment required that the team address these issues for multiple end-to-end mission capabilities, several of which involved common solution providers. Coordination of deliverables and resource management was a significant challenge in this environment.

### Coordination of Components, Mission Threads, and Mission Scenarios

Another key challenge for the BT environment was managing complexity and building for flexibility. The initial development for BT was aimed at producing key components for an infrastructure that initially fulfilled a few end-to-end mission threads. The infrastructure needed to mature into a system that integrated specific

capabilities spanning the whole mission area. Furthermore, the infrastructure needed to enable the sponsor to construct threads that exploited the synergies among mission areas. The sponsor has already seen a 3-fold growth in specific end-to-end mission threads. Continual growth in the number of solution providers added to the coordination challenge. A primary concern of this challenge was scalability of both technology and process.

### Integration of New Capabilities with Existing Capabilities

The reuse of existing capabilities was necessary to help maintain schedules and control costs. Several unique challenges arose when existing capabilities were integrated into the BT environment:

- Knowledge about the old capabilities may have been lost; time-consuming reverse engineering must be kept to a minimum.
- The existing capabilities are not designed to interact with one another.
- The existing capabilities may use old, obsolete, and sometimes incompatible data formats.
- The technologies used by the existing capabilities may be obsolete and incompatible with the new technologies.
- Existing capabilities used a large variety of programming languages, development environments, and hardware platforms.

A variety of techniques can be applied to address these challenges. For example, some existing capabilities can be encapsulated by a new wrapper to allow new interactions and to integrate with the new data flow; alternatively, solution providers can provide data translation and transformation services for interactions with the existing capabilities. The SE team and the solution providers needed to work together to assess the trade-offs among possible solutions.

### Transformation of Capabilities Across Incremental Deliveries

Concurrency of activities during any one development spin presented both technical and resource issues. Activities to plan for future spins, execute current spin development, perform integration and testing on previous spin deliverables, field previously developed capabilities, and monitor and report progress in these areas overlapped throughout the spin and often relied on the same set of resources. This challenge affected both the BT-level staff and the solution provider staff. Developers needed to work on at least three different versions of their capabilities (i.e., current development version, integration and testing version, and fielded version).

Systems engineers needed to track and coordinate requirements for end-to-end mission capabilities through multiple states (backlog, in development, verified by testing, and deployed to production).

The commitment to transformation and the management request for an exceptional commitment to BT, two of the key tenets of the program, at times resulted in surges in the use of resources for extended periods, which is unsustainable over the longer term. The SE effort needed to help control surges in the use of resources.

A significant challenge was designing for change. The early thinking, again based on a research or technology-focused perspective, presumed block upgrades without consideration for backward or forward compatibility. When deploying at scale, the block upgrade approach was no longer sustainable; incremental delivery of system components became essential. Both forward and backward compatibility had to be managed explicitly, with new components designed to accommodate existing capabilities—to operate without causing older components to fail—and designed with a view to the future. For example, a component should be designed to permit extensions to the data exchange formats without failing. If these criteria cannot be met, incremental deployments are difficult and time consuming and will eventually become impossible to perform as the system scales up.

### Management of Diverse and Evolutionary Interfaces

Many of the challenges described in other sections of this article relate to interface management. The BT program interfaces numbered in the hundreds and involved groups of solution providers that may have had limited interaction with the other teams before the BT effort. In addition, the new systems interfaces required the solution providers to define new or enhanced data exchanges to achieve the desired capability. Schemas and data formats that were once internal to a solution provider or exclusively controlled by a solution provider were now shared among component developers, and exclusive control was no longer possible. The increased documentation needed for adequate interface management and change control and the documentation to meet the needs of the acquisition process presented further challenges.

### Tailoring Acquisition Documentation

The DoD acquisition process is well suited for large programs with long lead times for delivering products and capabilities. The process needs to be tailored to provide meaningful documentation in the sponsor's environment of rapid capability deliveries in a rapidly changing environment.

The DoD acquisition process requires many Department of Defense Architectural Framework (DoDAF) diagrams (e.g., operational views OV-1 and OV-5, systems and services view SV-4, etc.) and documents (e.g., the SE plan, or SEP; and the testing and evaluation master plan, or TEMP; etc.) to communicate and document objectives related to acquisition decisions.<sup>12</sup>

The framework provides two approaches to delivering capability: a traditional waterfall model (single step to full capability) or evolutionary acquisition. BT followed the evolutionary acquisition model, as shown in Fig. 1, to deliver mission capability as quickly as possible. Historically two or three BT increments, each a distinct acquisition program, are in various stages (e.g., pre-milestone B, C, etc.) of the acquisition process. Because multiple BT increments are in play at any given time, the BT SE team must deal with overlapping acquisition milestones.

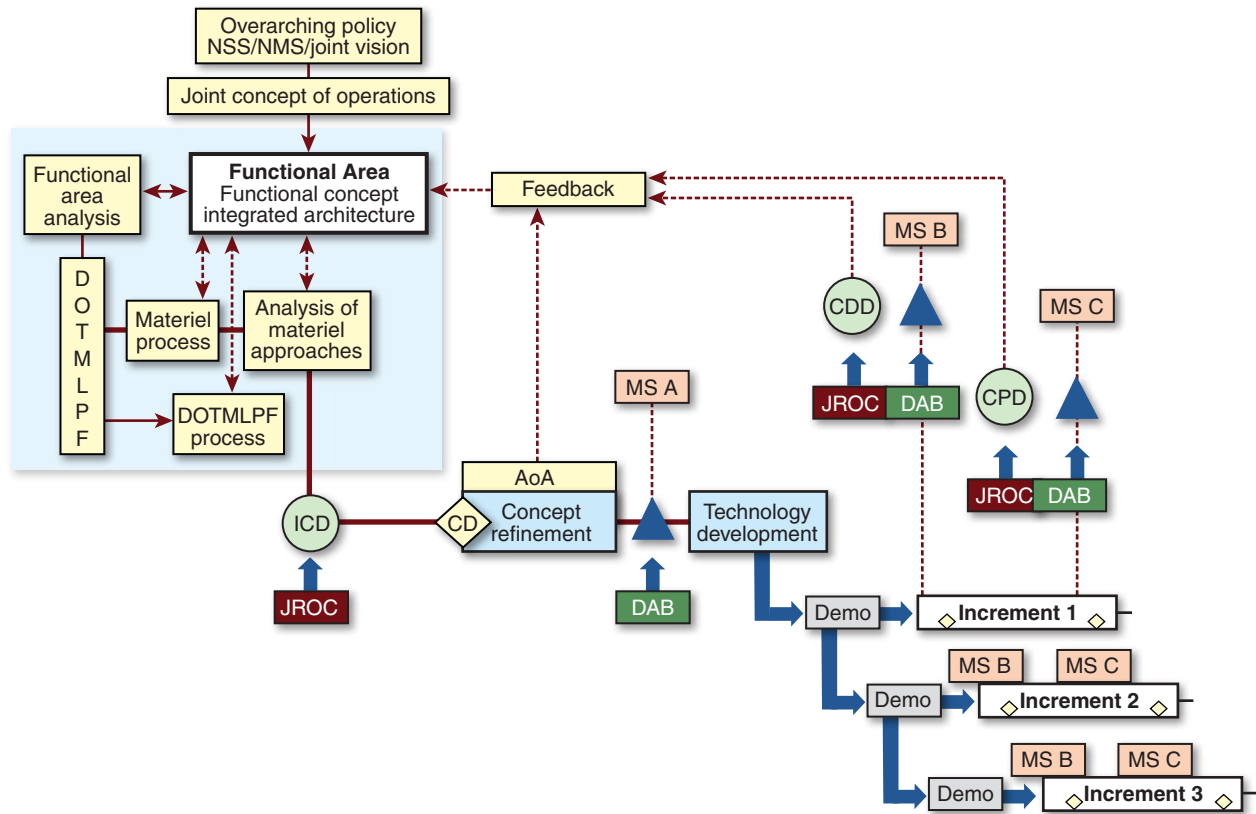
The challenge for SE is to maintain agile processes<sup>13</sup> and to be able to integrate at the Enterprise level, reach across to dependent projects and organizations outside of BT, and function within the technology development and formal increment phases of the acquisition to deliver the required capability to the user.

## VASE APPROACH

In light of the challenges previously described, the sponsor established a government-led BT VASE team, supported by APL, to provide SE guidance and support to development and fielding activities. The VASE team was positioned to provide SE services at the BT program level rather than at the solution provider level and focused on integration among the component developers. In some cases, the VASE team provided direct support to solution provider teams as needed to address particularly complex engineering problems. This section describes the approach taken to incrementally develop and refine the set of SE processes sufficient to address the challenges discussed in the previous section.

VASE was established incrementally by developing and refining the set of SE processes sufficient to address the challenges discussed in the previous section. Philosophically, the multiple solution provider efforts aligned best with agile development principles that recognize the value of rapid, iterative component delivery and embrace changing requirements.<sup>14</sup> The team adopted a mindset of judiciously applying resources to efforts that provide the most leverage against technical challenges. Thus, procedural changes or new processes needed to quickly demonstrate value or they were discarded. The VASE team adopted many agile development principles.

One of the key constructs in VASE is the thread engineering approach, by which a thread systems engineer is



**Figure 1.** The evolutionary acquisition process as defined in the DoD architectural framework.<sup>12</sup> The process defines multiple increments, each with milestone decisions. The development for the increments overlaps, as do the acquisition milestones. The SE team needed to maintain agile and rapid development processes for development while functioning within the longer formal increment phases of the acquisition. AoA, analysis of alternatives; CD, capability development; CDD, capability development document; CPD, capability production document; DAB, Defense Acquisition Board; Demo, demonstration; DOTMLPF, doctrine, organization, training, materiel, leadership and education, personnel and facilities; ICD, initial capabilities document; JROC, Joint Requirements Oversight Council; MS, milestone; NSS/NMS, national security system/national military system.

aligned with the mission capability threads (discussed in the *Mission Capability Threads* section of this article) to ensure that the products delivered by solution providers can be integrated into a complete mission thread to deliver an operational capability.

The following sections describe the processes that the VASE team has developed over time, the VASE team relationship with the BT solution providers, the rapid development strategy used with BT, and the SE construct for the mission threads used in VASE.

### Iterative Processes

As an outgrowth from multiple research efforts, the initial BT development process lacked a coherent SE approach. SE support for this transformational development effort was initially provided by a small cadre of government and APL systems engineers who had been forward-deployed from the enterprise SE organization. To establish credibility with technology-focused solution provider development teams, the SE staff sought

to quickly demonstrate that SE methods could improve solution provider deliverable products. The SE team focused initially on addressing deficiencies that were well recognized by the developers (e.g., cross-component integration and testing). As SE was shown to improve product quality and timeliness in these areas, the team focused on other SE functional areas in which improved processes would achieve the greatest gain in development effectiveness.

### Early Emphasis on Integration and Testing

One of the first areas to which VASE was applied was the integration and testing of the solution provider deliverables. Solution provider resources did not have the perspective to plan and execute integration and testing at the BT level. Furthermore, those resources properly had priorities tied to developing the specific technologies that the solution provider was responsible to deliver. In many cases, before establishing the SE processes, solution providers had not tested their interfaces



with their nearest neighbors, let alone with logical interfaces more than one step removed.

The first step was to define an integration and testing strategy that minimized the use of solution provider resources and put in place a testing staff that would become knowledgeable about the solution provider products and how they should function as a whole. Each solution provider had one representative participating with the testing team while the remainder of their resources concentrated on development activities. At this stage, the integration and testing environment was a set of interconnected equipment borrowed from solution providers and repurposed from a laboratory. The testing activities were performed by finding empty desks throughout the facility and by using runners and telephones to coordinate the tests.

Initial attempts at integration and testing clearly showed the value in the activity. Problems that had previously prevented integration of software components were identified by using the processes and tools in the testing environment. As a result, the solution providers volunteered to add resources to the effort and to perform the initial integration activity before the testing team received the deliverables.

As the integration and testing activities matured, the activity matured from very basic data flow and functionality testing to detailed test coverage analysis and implementation, complete regression testing, and a combination of laboratory and live testing. The testing environment gradually evolved into a formal testing room that held 30 people (developers and testers) and a testing laboratory with dedicated end-to-end testing systems.

### **Interface Definition and Management**

Another problem-prone area in which the SE team knew that an appropriate amount of attention would result in tangible, near-term benefits was interface management. Early on in the project, development teams defined and managed their interfaces with cooperating solution providers as needed to exchange data among their components. As the number, types, and versions of these interfaces expanded, integration and testing became increasingly difficult and the percentage of system deficiencies attributed to data exchanges across those interfaces grew. The SE team established methods to define, negotiate, and centrally manage the array of interfaces among the federated solution providers, helping to alleviate problems that plagued earlier component integration efforts. SE maintains an interface description wherever one solution provider interfaces with another solution provider. These interfaces were both direct and indirect (e.g., a solution provider two steps down from another solution provider needs the upstream solution provider to supply specific information or perform a specific transformation).

### **Requirements Definition and Management**

The requirements management process is another activity that matured through small, incremental, value-added steps. In the first few spins of development, only the testing team used documented requirements. The solution providers saw little value in formal requirements because they felt they had enough clarity on system needs and that a high-level activity statement from each solution provider was all that was needed to stay coordinated. The testing team was responsible for deriving their testing requirements from the spin plans. As the complexity of the program increased and mission thread teams were established, agreements on formal requirements became necessary. A complete end-to-end solution required allocating requirements across solution providers, which then defined the workload on solution provider resources. Once the requirements were distributed, the ordering of the component deliveries was specified so the end-to-end mission capability could be integrated and tested in the proper sequence. Once the solution provider deliverables were ready, they needed an environment that was properly equipped and configured to perform integration and testing for the capability.

At this stage, the testing team no longer had the resources to derive requirements and perform all of the testing required by the increased complexity of the systems. A requirements team was formed. Now, the requirements engineer needed both to derive the requirements and to publish requirements reports by using Microsoft Word and Microsoft Excel. Because the detailed requirements were developed too late in the process, this approach proved to be too slow and inaccurate to meet the needs of either the thread teams or the solution providers. An additional need for well-understood, coordinated, accurate, and timely requirements came from the acquisition process, which had advanced to the point at which formal developmental testing and evaluation results were required.

At the request of BT management, the thread leads, and the solution providers, a new requirements process and a requirements team was created to meet the following objectives:

- Early definition of requirements by the thread teams and the solution providers
- Coordination and negotiation of requirements between the threads and the solution providers and among the solution providers
- Involvement of the testing team in determining the testability of the requirements
- Provision of traceability of the requirements during development and testing
- Provision of a timetable of requirements delivery during a spin to the testing team, the thread teams, and management
- Provision of metrics about the requirements

To help meet these goals, the requirements team established a common requirements repository to provide visibility to the requirements and to aid communications among all participants, enabling all participants to access the repository with appropriate rights to accomplish their tasks.

### Acquisition Support

BT grew out of a prototyping effort. After its demonstration phase, parts of BT were included in a formal DoD major system acquisition. BT development and fielding is currently executed under a tailored version of the Defense Acquisition Management Framework established in 2003 (DODD 5000.2).<sup>15</sup> The tailoring reduces some documentation and review requirements but retains the overall structure and intent of the framework.

BT simultaneously executed activities in both an informal technology development phase leading to demonstrations (using a spiral development approach) to define future increments and in formal increments with appropriate acquisition milestones (B and C). Demonstrations were performed both in development and in live operational environments for technology maturation and user feedback.

One instance of documentation tailoring was the DoDAF artifacts. To keep pace with overlapping acquisition increments, the 90-day development cycles, and mixed program funding, the SE team chose to document the entire “to-be,” multi-increment architecture and create “cut-outs” of selected products to highlight the capabilities of the current BT increment. Highlighted products included the OV-1 (concept graphic) and OV-5 (activity node tree).<sup>12</sup> Operational activities pertinent to and funded by the active program increments were color-coded and appropriately annotated in the architecture diagrams. Architecture artifacts were updated before each milestone. The acquisition documents such as the SEP were much less dynamic and received minor updates before each milestone decision.

### Technical Planning and Control

Overall system complexity, including system interdependencies, mandated the need for in-depth technical plan-

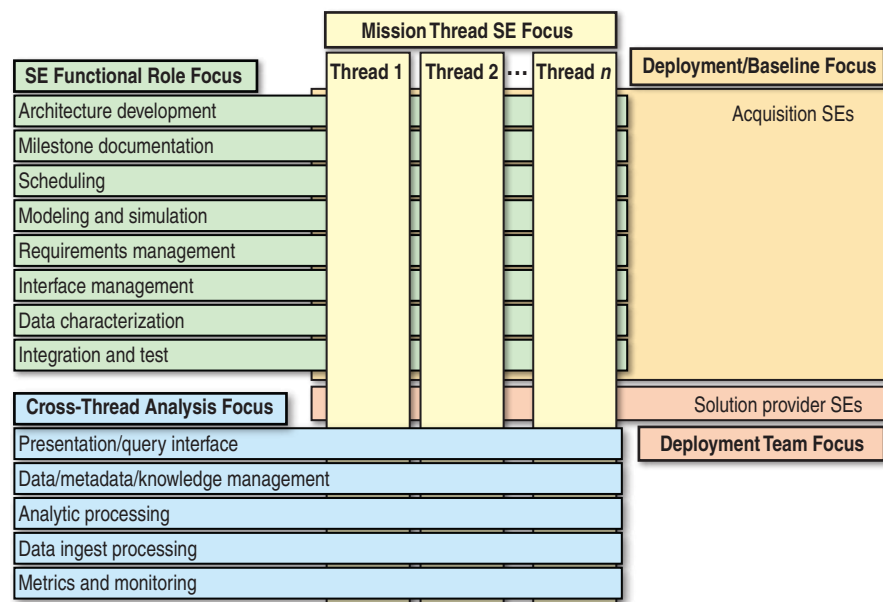
ning and monitoring within the spin construct. The technical planning and monitoring process was created and implemented along two thrusts, thread- and non-thread-related activities.

The thread systems engineers coordinated with the various projects to map out all capabilities and delivery dates needed in the spin time frame to meet the mission objective. Capabilities were tied to technical requirements decomposed from the mission objectives. Impacts to interfaces and testing were coordinated appropriately. As each capability was delivered during the spin, the integration and testing team verified that the requirements were met. Progress was tracked to understand overall schedule progress across all aspects of the thread.

Nonthread work is not tied to a specific capability thread. Those work items typically included capability or systems infrastructure enhancements and were also tracked within the requirements and product delivery process.

### SE Team Approach

APL also supported the sponsor in establishing an appropriately sized, multiple-perspective SE team structure tailored to the development and acquisition environment. Figure 2 depicts the relationship of these mutually supporting SE focus roles. The driver mechanism for mission capability development was the mission thread. Systems engineers assigned to the mission



**Figure 2.** The BT team structure includes multiple overlapping SE functions, with mission thread systems engineers providing the organizing construct for capability development. The thread systems engineers rely on functional-area systems engineers, whose activities center around traditional SE functions, but within the context of the mission threads that coordinate and synchronize the capability development.

threads executed a variety of SE tasks in support of the development, integration, and fielding of the thread capabilities. The thread systems engineer's role is examined in more depth in the *Thread Engineering Approach* section of this article.

A set of SE functional teams lent the thread systems engineers support in specific areas as needed. These teams, oriented along SE functional roles such as requirements management, interface management, integration and testing, etc., provided direct support to the thread systems engineers and ensured that common procedures and methods were used across the various mission threads.

Some systems engineers were assigned the role of coordinating capability development as it related to a particular acquisition program. These acquisition-focused SEs also received support from the SE functional teams in key areas such as architecture development and programmatic milestone documentation. They also coordinated closely with the mission thread systems engineers to ensure that capabilities under development met specific program objectives.

Some BT components were sufficiently complex that they warranted dedicated, cross-subsystem SE support. These solution-provider-focused systems engineers performed analysis, design, and technical coordination activities among the component's subsystems and performed key interface engineering activities at the component system boundaries with other BT systems.

The SE team later established a cross-thread analysis focus role aimed at identifying and coordinating common approaches across all the mission threads in capability functional areas such as metadata and knowledge management, presentation and query interfaces, and capability metrics and monitoring. This cross-thread SE focus helped reduce the risk of any particular mission thread becoming a stovepipe in itself.

### Federated Solution Provider Teams

BT development was supported by more than 25 cooperating solution provider teams, ranging in size from three to more than 80 team members. Solution provider organizational affiliations ranged from tightly coupled (funding and management structure was provided by BT and all development was focused on BT) to decoupled (funding and/or management structure was not provided by BT and BT development was one of many corporate needs provided by that solution provider). A decoupled solution provider might be, for example, a research organization or a central information technology organization.

Each solution provider was responsible for supplying specific functional capabilities, some narrowly scoped and others more broadly applied throughout the infrastructure. The capabilities are represented by the puzzle

pieces in Fig. 3. The scope of the capabilities ranged from infrastructure services (e.g., hosting environment or messaging service) to highly specialized mission processing. The complexity of the capabilities ranged from moderately complex, such as developing a family of data transformation, extraction, or enhancement utilities, to highly complex endeavors such as monitoring system health and status to support automated command and control functions. The relatively autonomous nature of the individual teams allowed each to use technologies, tools, and processes tailored to their development needs.

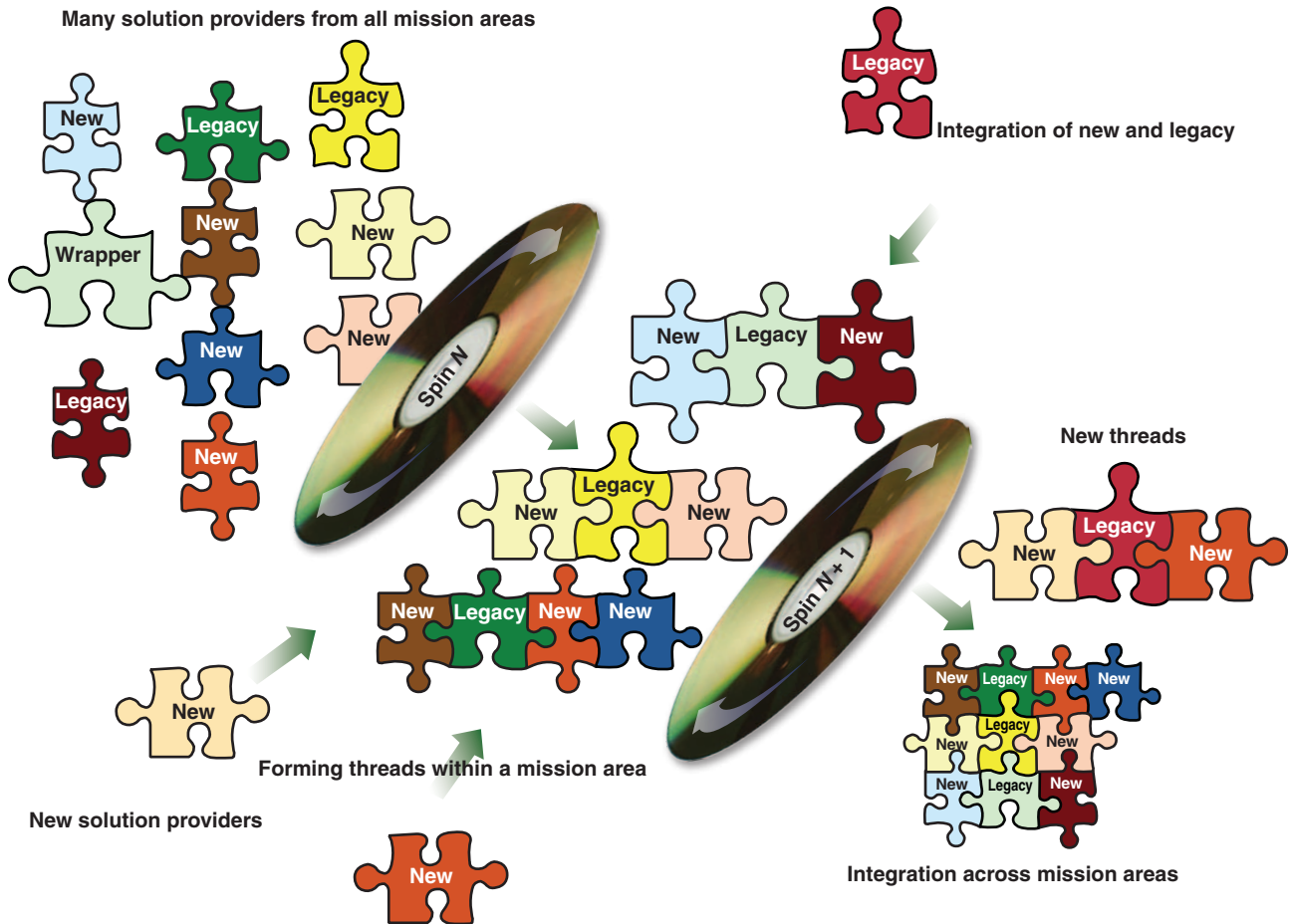
The various solution provider teams may have been focused on different mission areas, and early in the evolution of BT, the solution provider's puzzle pieces were assembled into mission capability threads within a single mission area. The real power of the VASE approach and mission thread engineering is in the ability to tie together capabilities within a mission area and across mission areas as needed to meet mission needs.

SE was critical to bringing together the components from different solution providers, but the heterogeneous nature of the teams meant that teams engaged with SE at varying levels; the result is that SE did not have a uniform view into the solution providers. Figure 4 is a graphical representation of the SE visibility concept. In some cases, solution providers operated entirely autonomously without SE support (virtually no visibility, black edge); in others the solution providers drew on systems engineers as a pool of resources with specific skill sets (limited visibility, light gray edge); in others, particularly those most central to BT and most complex, SE was integral to the solution provider team in the design and development activities (dark gray edge). The result of this development team specialization is that BT is an assemblage of "black boxes" with gray edges. The gray edges are necessary to allow capability interaction and cooperative development of an end-to-end mission. Capability interaction is achieved through interface definitions, and cooperative development is achieved through information sharing (e.g., a hosting environment and a mission capability need to agree on services provided by the hosting environment).

Technical guidance and governance for the solution provider federation was provided by an upper-level BT management team that reported directly to the most senior level at the sponsor. Interaction among BT technical management and solution provider teams was accomplished through weekly meetings of the technical leads, management forums, and one-on-one meetings as needed.

### Rapid Development

The iterative, spin-oriented development and fielding approach used in the BT project set forth certain assumptions and expectations for each development spin, including the following:



**Figure 3.** Federated solution provider teams are the puzzle pieces that make up a mission-area capability. The solution providers are aligned on each spin to deliver end-to-end capabilities based on mission threads. In early spins, a solution provider was likely focused on a single mission area; the VASE approach and mission thread engineering allowed the sponsor to evolve to an environment in which puzzle pieces within a mission area and across mission areas can be tied together to meet mission needs.

- Each spin will produce usable mission capabilities.
- Capabilities may mature over the course of several spins.
- Not all capabilities will mature at the same rate.
- New capabilities can be added at any time.
- Development that does not meet operational needs as anticipated (e.g., prototypes, concept explorations) will be discontinued.
- A capability may require more than one spin for initial planning, prototyping, and design.

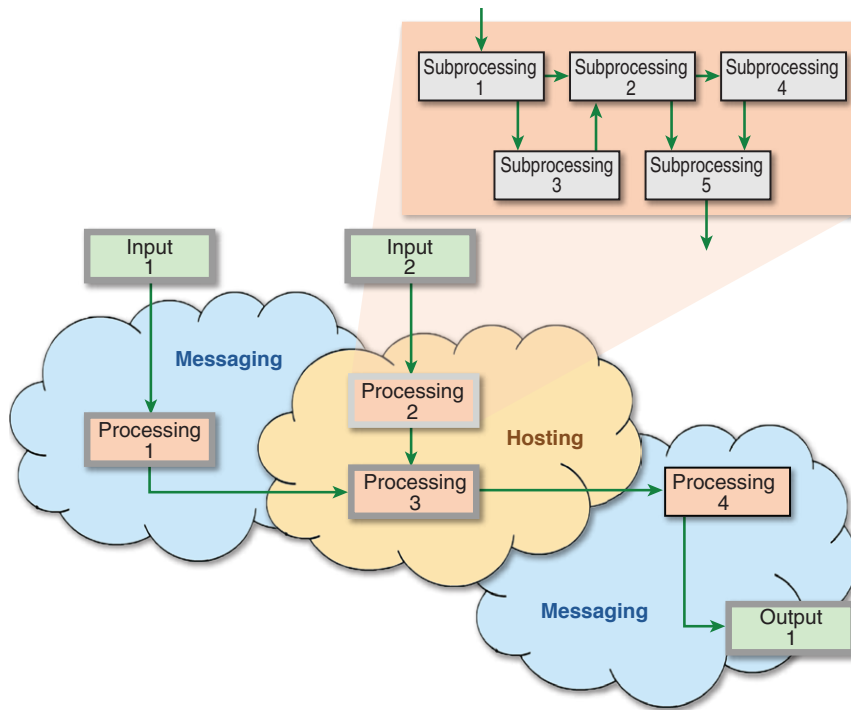
Several timing strategies for the iterative life cycle were tried during the initial spins. The timing strategy has matured to the following for a representative spin  $N$ :

- Planning and requirements definition occur in spin  $N-1$ .
- Development, incremental integration, incremental testing, and status reporting occur in spin  $N$ .
- Final integration, final testing, formal developmental testing and evaluation, and field testing occur in spin  $N+1$ .
- Deployment occurs in spin  $N+2$ .

Although the representative cycle shows a time to market (planning to deployment) of three to four spins (9 months to 1 year), the process accommodated emergent needs that must traverse the development and fielding cycle more quickly. A major impact of the iterative life cycle for the BT solution providers was the need to manage, monitor, and report status for activities that occur over four spins: planning spin, development spin, test spin, and deployment spin. During any one spin, solution providers were addressing capabilities that were in different phases of the life cycle. Figure 5 is a graphical representation of the spin life cycle.

### Thread Engineering Approach

The SE team worked with the sponsor to develop the concept of end-to-end mission threads, each supported by a thread systems engineer, and to use them as organizing constructs to focus capability development. A thread is an end-to-end functional capability that satisfies an operational need and accounts for all data acquisition, processing, storage, and user presentation functions



**Figure 4.** The solutions provided by development teams come together as black boxes that expose interfaces. By their nature, federated teams engage with SE at varying levels, meaning that SE does not have a uniform view into the solution providers. Some of the teams, especially those building capabilities central to BT, had SE integral to the team. SE had better visibility and could connect the black box at “dark gray edges” with a fine-grained understanding of the internal implementation. For others, SE had a less detailed view, akin to a light gray edge. The federated model accommodates solution providers that have little involvement with SE and define sharp black edges.

needed to achieve that capability. The thread engineer was central to the operation of the BT development effort and, more broadly, to the engagement of the BT development effort with the user and acquisition communities. Using these mission threads to organize the application of SE methods and resources provided a context for defining needs, conducting analysis and design activities, and integrating the various system components provided by the solution providers. Over time, the SE team defined a set of processes and methods that constituted the thread engineering approach that is discussed in this section. Using the mission thread context, in conjunction with cross-thread analysis activities, also helped identify and prioritize common infrastructure components and services throughout the architecture.

Thread-focused engineering was carried out by a team composed of a government lead, one or more thread-focused systems engineers, and the technical leads of the development teams slated to deliver components that enable the thread capability. The strategy of carrying out engineering activities focused on mission threads evolved over the course of several development spins. Over time, the SE team adopted and refined techniques and methods that proved valuable

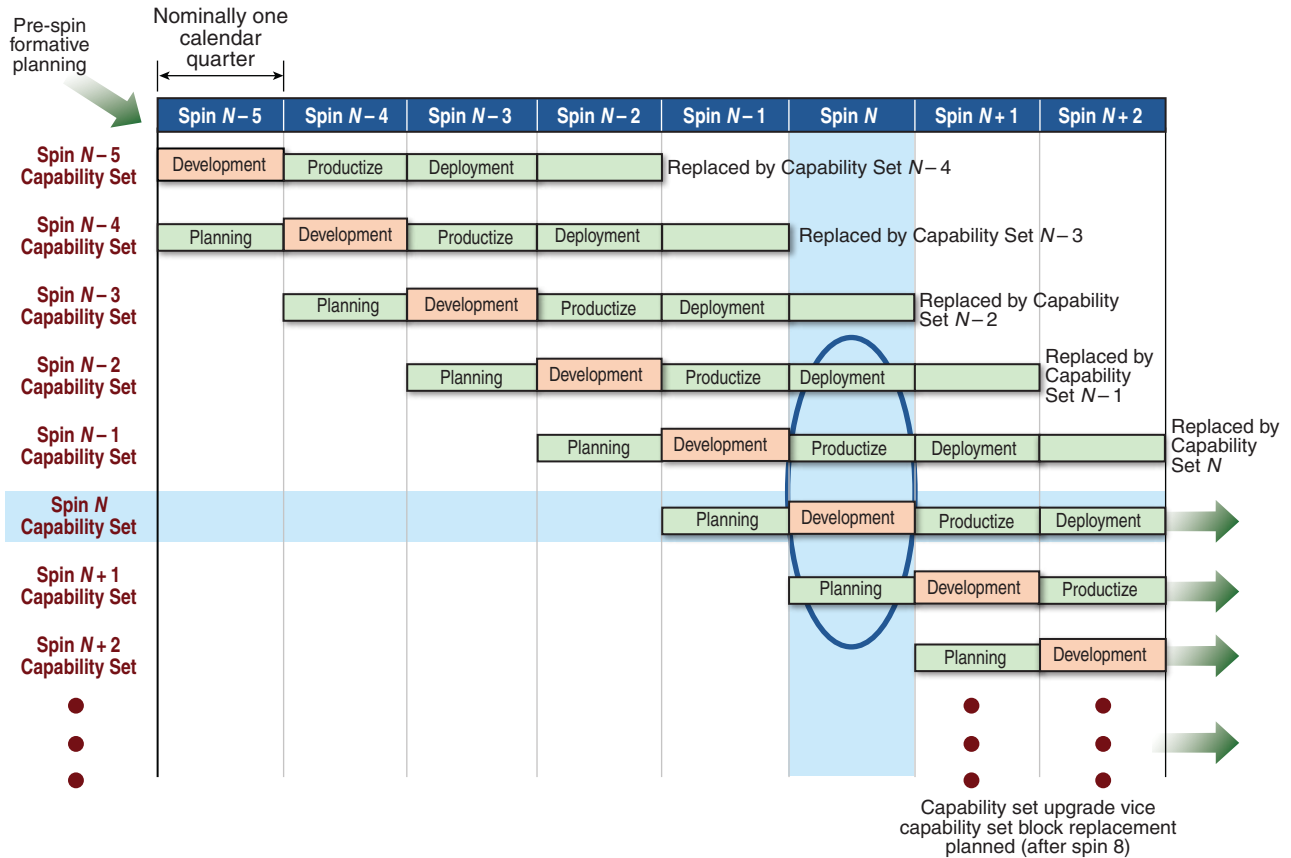
and dropped candidate methods as soon as they proved unsuitable in the federated, rapid development environment.

The remaining sections discuss specific thread SE functions and tasks. Those functions and tasks interact with and support the overlapping SE functions illustrated in Fig. 2. The threads tie together the SE functions to deliver end-to-end capabilities.

### **Requirements Identification and Management**

Thread systems engineers worked closely with the acquisition systems engineers to ensure understanding of how their thread capability satisfied, either in whole or in part, requirements contained in acquisition program documents. They also cultivated and maintained working relationships with the technical staff of end-user organizations to adequately assess the underlying technical requirements and constraints associated with a particular mission capability.

The thread systems engineers were responsible for identifying and tracking system-level technical requirements resulting from mission thread analysis and design activities. As discussed in the *Iterative Processes* section of this article, the early requirements definition and management process had requirements developed and documented by a systems engineer focused on integration and testing. With the more mature approach, the thread systems engineer had an end-to-end focus with primary responsibility to develop the requirements to meet mission needs; the requirements management team provided an engineer to assist in documenting the requirements and maintaining them in the requirements repository. During development spin planning, the thread team reviewed the requirements backlog, consisting of previously cataloged system requirements not yet satisfied, along with any newly identified requirements, to prioritize the requirements and to allocate them to a development spin. After this initial thread planning phase, requirements were forwarded to the appropriate solution provider teams for analysis, allowing the developers to examine all of the development or support tasks assigned to them by the thread teams. The next planning phase involved iterative negotiation between the solution provider teams and the pertinent thread teams to ensure that the thread



**Figure 5.** Concurrent development spins are central to the ability to bring capabilities to the field rapidly. Each spin produces usable mission capabilities, but some capabilities mature over multiple spins. The circle around spin N development illustrates how SE manages across four spins, each in different phases of the cycle. The rapid cycle for any one spin allows quick execution of decisions on reprioritizing capabilities and abandoning development that does not meet operational needs.

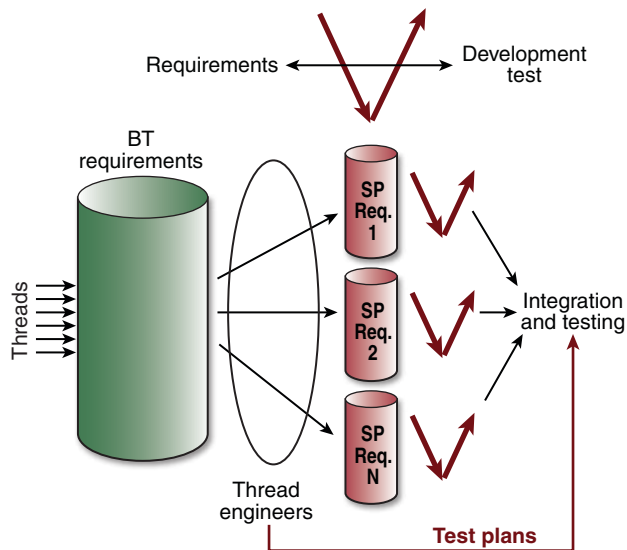
requirements were understood and to develop estimates of what was achievable given resource constraints. Any cross-thread or cross-component conflicts that could not be resolved during this process were referred to technical management for adjudication and final resolution. This requirement-planning phase resulted in the set of requirements that were queued to be addressed by the solution providers during the upcoming development spin. An abbreviated form of this process was used to address emergent or *ad hoc* requirements that arose outside of the normal spin-planning cycle. The requirements management team also produced the requirements verification and traceability matrix used during integration and testing.

Figure 6 depicts a two-level (BT and solution provider) requirements strategy with each level performing its own requirements management. The diagram also highlights the role of thread engineers within this strategy. The requirements for the entire system were maintained at the BT level. Thread engineers allocated requirements to solution providers and developed (or contributed to) testing plans for integration testing based on requirements maintained at the BT level. Solution providers had the

freedom to manage requirements within their projects as they saw fit, but solution providers had to pull the BT requirements relevant to their support for threads in the current spin into the solution-provider-specific requirements management process. (Solution providers had the option to use the requirements management tools at the BT level for their requirements management process, but that decision was made by the solution provider team.)

**System Analysis, Design, and Architecture Technical Documentation**

The thread engineering construct used the thread team approach to support analysis and design activities needed to define the implementation solution for any given mission capability. The thread team carried out a variety of analysis tasks needed to support architecture and performance trade decisions that influenced development. The analysis tasks were organized as needed to address the specific design questions at hand and were supported as needed by systems engineers from the component development teams, analysts from the modeling and simulation team, and subject-matter experts on a



**Figure 6.** Requirements management is a central SE function and is essential to the VASE approach. BT system-level requirements are developed by thread systems engineers to meet mission needs, with the support of a systems engineer from the BT requirements management team, which allocates the requirement to a development spin. The requirements are allocated to the solution provider team for analysis, and iterative negotiations eventually lead to a development plan for the spin, which then drives integration and testing planning for the spin. SP Req., solution provider requirement.

particular mission domain. By using analysis results as input, the thread team synthesized the implementation option and produced the design documents and architectural artifacts needed to support follow-on development activities. Although these design artifacts adhered to standard thread engineering templates, thread teams had the option of extending this set of design documents as needed to address thread-specific needs.

### Thread Technical Planning and Control

The government leads, supported by the thread systems engineer and the rest of the thread team as well as the analysts from the scheduling team, were responsible for developing and coordinating the capability development and deployment schedule for their assigned mission capabilities. Working closely with technical management, they identified key dependencies between solution providers, defined milestones, and developed resource allocation approaches to achieve the scheduled work. They also developed risk management strategies. The thread team leads established the working model for the team. Much of the work was done virtually, and most teams conducted weekly meetings to ascertain status, perform technical reviews, discuss risks, and reorient the plan as necessary. The thread teams also participated in quarterly technical reviews with technical management.

### Interface Management

From the analysis and design stage and continuing through the deployment phase, the thread team coordinated with the interface management team to identify and manage system-level interfaces between components implementing thread capabilities. The interface management teams maintained an extensive library of systemwide interface diagrams that depicted several different interface perspectives covering both physical and logical representations. These diagrams were generated from an interface database that catalogs the various protocol layers and schemas that document the interfaces. The thread team also coordinated with the interface management team, the requirements management team, and cross-thread systems engineers to identify areas of potential conflict with or opportunities for convergence with other mission threads.

### Integration and Testing

The thread team also collaborated on activities required to integrate the various components developed by the solution providers and to shepherd the capability through product verification. The thread systems engineer worked several integration and testing tasks during the development spin. Foremost of these was identifying an appropriate test data set that would exercise the system functionality under testing. The data characterization team helped the thread systems engineer to articulate test data needs and to either identify an existing test data set that satisfied the need or to work to procure or generate an appropriate data set. By using the requirements verification traceability matrix developed in coordination with the requirements management team, the thread systems engineer collaborated with the integration and testing team to develop the test plan and test cases for the integrated capability. During component integration and testing, the thread systems engineer provided technical support to the integration team and maintained liaison with the solution provider in order to obtain integration support as needed to address deficiency reports and other issues.

### Capability Deployment

The thread team supported capability deployment by participating in deployment planning activities and providing assistance to deployment engineers during system installation and configuration. During initial system operation, the thread systems engineer monitored performance of the system to ensure that metrics fell within expected bounds and collaborated with system maintenance staff to identify and troubleshoot abnormal system behavior. Those system observations were documented and provided feedback to the iterative capability planning and development process.

## Cross-Thread SE

The SE team established a cross-thread analysis focus aimed at identifying and coordinating common approaches across all the mission threads in capability functional areas such as metadata and knowledge management, presentation and query interfaces, and capability metrics and monitoring. This cross-thread SE focus helped reduce the risk of any particular mission thread becoming a stovepipe in itself.

## CONCLUSION

VASE has been developed to tailor SE activities to the federated and iterative (spin-based) development process. Thread engineering is the centerpiece of the VASE approach. The thread engineer provided a common interface point among the end customers, acquisition programs, and developers, ensuring end-to-end mission functionality for applications. If VASE could not demonstrate to all stakeholders that it added value to the development process, VASE would be ignored. Without a common interface point, provided through VASE by the thread engineer, development would be less effective in meeting mission needs, thereby impairing the sponsor's ability to field mission capabilities.

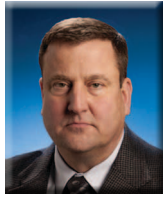
Thread engineering led to several successes: system development and testing improved significantly, allowing the sponsor to grow from five active threads to more than 15; the thread engineering concept spread from its original program to other programs within the sponsor's organization; and solution providers initially resistant to SE involvement have requested SE support. Next steps include tailoring thread engineering to other VASE activities such as process integration and solution provider activities.

## REFERENCES

- <sup>1</sup>Report of the Defense Science Board Task Force on Department of Defense Policies and Procedures for the Acquisition of Information Technology, Defense Science Board, Washington, DC, <http://www.acq.osd.mil/dsb/reports/ADA498375.pdf> (March 2009).
- <sup>2</sup>Rey, R. F. (ed.), *Engineering and Operations in the Bell System*, 2nd Ed., AT&T Bell Laboratories, Murray Hill, NJ (1983).
- <sup>3</sup>AT&T Network Planning Division, *Notes on the Network*, AT&T Bell Laboratories, Murray Hill, NJ (1980).
- <sup>4</sup>Wertz, J. R., and Larson, W. J. (eds.), *Space Mission Design and Analysis*, 3rd Ed., Microcosm Press, El Segundo, CA (1999).
- <sup>5</sup>Turner, R., "Towards Agile Systems Engineering Processes," *CrossTalk* 4, 11–15 (2007).
- <sup>6</sup>Bar-Yam, Y., "When Systems Engineering Fails—Toward Complex Systems Engineering," in *Proc. International Conf. on Systems, Man & Cybernetics*, Piscataway, NJ, Vol. 2, pp. 2021–2028 (2003).
- <sup>7</sup>Department of Defense, *Global Information Grid Architectural Vision: Vision for a Net-Centric, Service-Oriented DoD Enterprise*, <http://cio-nii.defense.gov/docs/GIGArchVision.pdf> (June 2007).
- <sup>8</sup>Department of Defense, Directive DODD 8000.01, *Management of the Department of Defense Information Enterprise*, <http://www.dtic.mil/whs/directives/corres/pdf/800001p.pdf> (10 Feb 2009).
- <sup>9</sup>Department of Defense, Directive DODD 8320.02, *Data Sharing in a Net-Centric Department of Defense*, <http://www.dtic.mil/whs/directives/corres/pdf/832002p.pdf> (2 Dec 2004).
- <sup>10</sup>Department of Defense, Directive DODD 4630.05, *Interoperability and Supportability of Information Technology (IT) and National Security Systems (NSS)*, <http://www.dtic.mil/whs/directives/corres/pdf/463005p.pdf> (5 May 2004).
- <sup>11</sup>Chairman of the Joint Chiefs of Staff, Instruction CJCSI 6212.0IE, *Interoperability and Supportability of Information Technology and National Security Systems*, [http://www.dtic.mil/cjcs\\_directives/cdata/unlimit/6212\\_01.pdf](http://www.dtic.mil/cjcs_directives/cdata/unlimit/6212_01.pdf) (15 Dec 2008).
- <sup>12</sup>Department of Defense, *Department of Defense Architecture Framework, Version 1.5*, [http://cio-nii.defense.gov/docs/dodaf\\_volume\\_i.pdf](http://cio-nii.defense.gov/docs/dodaf_volume_i.pdf) (23 April 2007).
- <sup>13</sup>Dove, R., "Fundamental Principles for Agile Systems Engineering," in *Proc. 2005 Conf. on Systems Engineering Research (CSER)*, Hoboken, NJ, <http://www.parshift.com/Files/PsiDocs/Rkd050324CserPaper.pdf> (2005).
- <sup>14</sup>Haberfellner, R., and de Weck, O., "Agile SYSTEMS ENGINEERING Versus AGILE SYSTEMS Engineering," in *Proc. 15th Annual International Symp. of the International Council on Systems Engineering (INCOSE)*, Rochester, NY, [http://strategic.mit.edu/docs/3\\_59\\_INCOSE-2005-AGSEvsEAGS.pdf](http://strategic.mit.edu/docs/3_59_INCOSE-2005-AGSEvsEAGS.pdf) (2005).
- <sup>15</sup>Department of Defense, Directive DODD 5000.02, *Operation of the Defense Acquisition System*, <http://www.dtic.mil/whs/directives/corres/pdf/500002p.pdf> (8 Dec 2008).



# The Authors



W. Scott Gibson



Stephen F.  
Schreurs



Duane L.  
Linsenbardt



Antonio  
DeSimone

**Charles R. Spaulding** (not pictured) is a senior systems engineer for the DoD sponsor and the government's lead systems engineer for this DoD project. He manages the engineering efforts from planning to integration and testing and through deployment of capabilities to the field. **W. Scott Gibson**, a Senior Professional Staff member in APL's Applied Information Sciences Department, has been in a variety of systems engineering roles for DoD sponsors engaged in transformational systems design and development. He has led multiple threads for the project described in this article and has participated at the systems engineering management level as well. **Stephen F. Schreurs** is a member of the Principal Professional Staff in APL's Global Engagement Department and led the APL team of systems engineers directly supporting the systems engineering effort described in this article. He has more than 20 years of experience as a senior systems engineer and has developed information management systems for private industry and the federal government. **Duane L. Linsenbardt** is a Senior Professional Staff member in APL's Global Engagement Department and has more than 20 years of experience in systems programming and network engineering for online systems. He has participated in both thread and solution provider systems engineering work and DoD acquisition support for the sponsor of the effort described in this article. He had specific responsibility for information technology systems architecture. **Antonio DeSimone** led APL's systems engineering team supporting this DoD sponsor from 2007 to 2009. He spent 18 years in the private sector, where he led product development and research teams in Internet and telecommunications technologies. Dr. DeSimone is now with the National Security Analysis Department at APL, engaged in systems engineering work for senior leader communications at the national level. For further information on the work reported here, contact Scott Gibson. His e-mail address is [scott.gibson@jhuapl.edu](mailto:scott.gibson@jhuapl.edu).