

Integrating COTS Software: Lessons from a Large Healthcare Organization

C. Todd Coutts and Patrick F. Gerdes

Noblis

An organization that decides to acquire and implement a COTS product instead of developing a customized solution must fully understand the characteristics of COTS integration. In addition, the project team must manage and balance both business and IT needs throughout the effort.

Both government agencies and private sector organizations are feeling pressure to maximize their return on IT investment, which often means using custom designs more sparingly. Indeed, the US Office of Management and Budget (OMB) states that capital investments should reduce risk by “avoiding or isolating custom designed components.”¹ For commercial companies, an ingredient in competitiveness is software that shortens their products’ time to market—which often goes against using custom applications.

In the face of this pressure, many IT managers are turning to commercial off-the-shelf (COTS) software applications. OMB, for example, directs agencies to acquire COTS software unless the cost-effectiveness of developing custom software to meet mission needs is clear and documented.² But COTS products aren’t a panacea, particularly if the aim is to integrate them into complex operational and technical environments. IT managers must understand the unique characteristics of COTS integration projects to reap a COTS application’s full benefit.

From 2003 to 2006, Mitretek Systems (now Noblis) worked in a project management and systems engineering advisory role with a large healthcare organization as it introduced several large COTS applications into an environment that traditionally emphasized custom development. The COTS applications included customer relationship management, enterprise resource planning, and other applications that provide prepackaged business functions. The integration projects had varying degrees of success and consequently offer useful lessons for other organizations.

Separate the COTS Integration Life Cycle

Figure 1 shows a COTS integration life cycle. The project life cycle when implementing a COTS product differs from that of a custom development project.³ For example, COTS package configuration replaces systems development as the primary technical activity after the team completes design. If the organization has a well-developed custom development life cycle, it might need to tailor it for a COTS integration

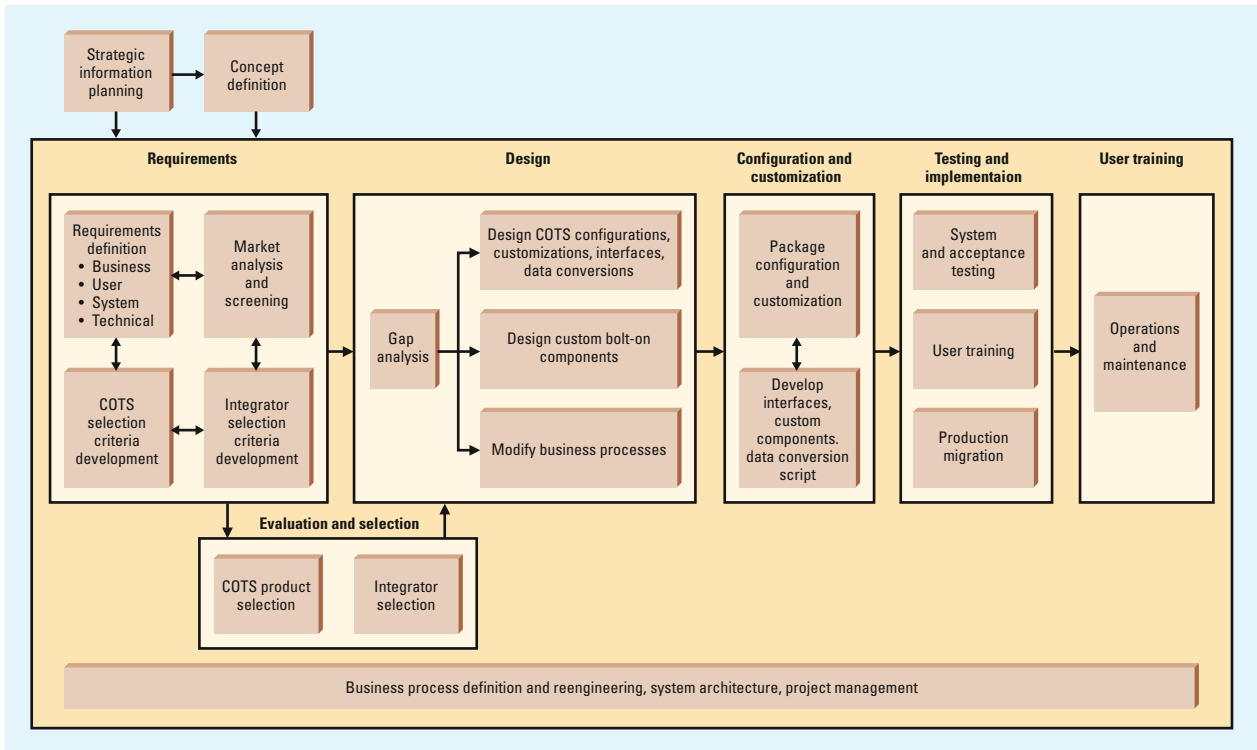


Figure 1. Integration life cycle of commercial-off-the-shelf software. When integrating a COTS product, configuration and customization become the primary technical activity after the design phase. In addition, during requirements definition, the project team should consider the products currently on the market.

project. If it doesn't, the organization's project management office and other oversight functions might try to impose irrelevant lifecycle reviews or technical standards, as occurred in some of the healthcare organization's COTS integration projects.

As Figure 1 shows, the COTS integration life cycle requires the same degree of systems engineering rigor and discipline, but it differs from the system development life cycle in several important ways. Awareness of these differences should be at the core of any strategy for managing a COTS integration project.³

Perhaps the greatest difference is that the COTS integration life cycle includes a phase for evaluating and selecting a COTS product. If no suitable COTS solution exists, the project might need to transition to custom development. The remaining differences pertain to the projects themselves. COTS integration projects should

- *minimize custom coding.* Significant customization of COTS products introduces risk and complexity into the project and negates the benefits of purchasing a COTS software application.

- *within reason, adapt requirements to the market's capabilities.* Requirements for COTS integration projects should be flexible enough to exploit the functions vendors can provide without significant customization.
- *be strongly linked to business process definition and reengineering.* The organization might have to change business processes to exploit the COTS product's functionality and the business practices it embodies.

Requirements definitions should document user requirements while considering available products. During design, the project team should compare the requirements with the out-of-the-box COTS product's capabilities and identify gaps. The main focus in the configuration and customization stage is to use the vendor's utilities to configure the product to user requirements as opposed to writing a large amount of custom code. Finally, the project team must test the system and migrate the final solution to production.

Determine the Organizational Fit

COTS integration might not be appropriate for all system needs. An organization must decide if the

vendor marketplace offers a COTS product that can satisfy both the operational and system requirements. What IT considers “essential need” might not match the business owner’s definition. Whereas IT might focus on the architectural fit and alignment with infrastructure standards, business owners might seek the value-add of a solution that streamlines current business processes using built-in functionality. If the business process associated with the system is unique and critical to the organization’s strategy, finding a COTS product that satisfies requirements without requiring significant customization might be difficult. In these cases, custom development might be more appropriate.

What IT considers “essential need” might not match the business owner’s definition.

One of the healthcare organization’s COTS integration projects sought to automate a well-known customer care and case-tracking process. The organization was willing to modify its existing business process to conform to the COTS functionality provided. This willingness was a major reason that the project was one of the most successful in meeting requirements on schedule and within budget. When conflicts arose between the native COTS product capability and a desired user requirement, the project team identified and evaluated the underlying requirement driver. If the requirement wasn’t based on a mandatory policy, they adapted the desired user requirements to the COTS product’s native capabilities.

Incorporate Multiple Views of Requirements

As in custom development, users must help define requirements. The main difference in COTS integration projects is the mindset.³ An organization achieves the optimum business value of a COTS purchase by exploiting as many out-of-the-box capabilities as possible. This can mean compromising on requirements that don’t affect

the core mission or revising current business processes to use the product’s functionality without customization.

In one of the healthcare organization’s projects, for example, the project team discouraged users from requesting extensive changes to the COTS product’s screens to meet personal design inclinations. The team achieved the business acceptance of the changes using several tactics. First, the team set the expectations upfront during the requirements definition phase that the COTS solution probably wouldn’t meet all the requirements and that compromises would be necessary. Second, the team used extensive walkthrough and review sessions with end users to preview the COTS functionality and workflow, and discussed some of the anticipated process changes.

Another difference between COTS integration and custom development is the selection process—perhaps the most critical phase of a COTS integration project. An important output of the requirements phase is selection criteria for a COTS product and systems integrator. As the project team elicits user requirements, it should document selection criteria for the COTS package and system integrator.³

Market analysis and screening are also critical. As team members gather user requirements and gain an understanding of the system, they should survey the market to identify candidate COTS packages. During this screening process, the team can feed results back to requirements definition, thereby adapting requirements to the target COTS product.³

In one of the more successful of the healthcare organization’s COTS projects, the team defined initial requirements while developing the system’s business case. During requirements definition, the team identified candidate COTS packages and then adapted the requirements based on the candidate products’ capabilities. While developing the initial requirements, the team also documented selection criteria and requirements for the COTS package and system integrator. At the end of the requirements phase, the team had a documented business case, requirements, a target set of COTS products, and selection criteria for the acquisition. At this point, the team should share the adjusted requirements set with the business users to continually align their expectations of the COTS product.

Make Requirements Definition and Design Interactive

COTS integration projects lend themselves to interactive requirements and design processes. While configuring and customizing the prepackaged system, the project team can, in many cases, quickly generate screens and reports for user review with a reasonable effort. These review sessions—or *conference room pilots*, as some vendors and integrators call them—let users provide requirements and design feedback on the basis of early user interface versions. Conference room pilots are also valuable in validating product selection decisions.^{3,4}

In one of the healthcare organization's projects, the project team didn't let users see early versions of the screens provided with the COTS package. When the team implemented the final solution, the users were less than enthusiastic because the team hadn't identified other COTS configurations, which would have made navigation easier. The most successful projects, on the other hand, conducted regular reviews that included a user interface preview.

Consider Selecting a System Integrator

During the selection phase, IT managers should consider whether contracting with a systems integrator to configure and implement the COTS product is necessary. Many larger COTS packages require experienced and trained personnel to configure them properly. Some vendors require certification in their products before they grant access to documentation and technical support. If IT managers don't have these skills in house, they might need to engage an experienced integrator.

One of the healthcare organization's projects didn't take this approach. Instead, it relied on the vendors' sales engineers (see the next lesson) and in-house staff to configure and implement the system. Consequently, the team didn't configure the COTS products correctly for the target requirements set, and the project manager had difficulty getting support after buying the software license. The implemented system experienced many performance problems and excessive service outages. In the successful projects, in contrast, the teams selected an integrator with extensive experience in the respective COTS product and relied on certified individuals.

Don't Rely on the Vendor for Everything

Most vendors have a professional services staff that can assist with designing, configuring, and implementing their products. However, IT managers should ensure that the vendor can support the full life cycle before deciding to rely on that vendor alone for configuration and implementation support. Few vendors have a business model that can support integration services for a full project life cycle. Furthermore, the vendor's primary goal is often to sell the COTS product; selling its professional services is secondary. Although these goals don't necessarily conflict with the business's goals, they don't ensure success

Few vendors have a business model that can support integration services for a full project life cycle.

either. From a risk-mitigation perspective, augmenting the vendor's offering or capabilities with a well-chosen systems integrator could provide the objectivity needed to guide the integration effort given the business goals and needs.

Have a Complete Set of Selection Criteria

Selecting and implementing a COTS product represents a significant investment and partnership with a software vendor. To realize return on this investment, IT managers must consider at least four criteria in selecting a product:

- *Functionality.* How well do the COTS products meet the business and user requirements as defined in the requirements phase?
- *Architecture.* How closely do the COTS products meet the technical requirements and align with the organization's technical architecture and policy?
- *Lifecycle costs.* What are the complete lifecycle costs of implementing and operating the COTS product?
- *Vendor.* How stable are the vendor and product line?

Table 1. Sample technical architecture criteria.

Criterion	What IT managers should ask (at the very least)
Overall architecture	What is the basic system architecture (for example, two or three tier)? What type of user interface does the product support (that is, does it require a Web browser or desktop software installation)?
Operating system, database management system, Web servers, application servers	Does the product support the organization's standard operating system, DBMS, Web servers, application servers, or any other supporting technical infrastructure standards?
Port and protocol requirements	What communications protocols does the COTS package use (Java Database Connectivity for databases, HTTPs from the desktop)? What communications ports must be open for the COTS package to function properly?
User management	How does the COTS package manage user accounts (that is, does it integrate to Lightweight Directory Access Protocol or manage them internally)? How does the COTS package manage authentication and authorization?

An organization that ignores any of these criteria risks selecting a COTS product that can't meet its requirements. Understandably, an organization should consider many more COTS product selection criteria to help zero in on the best choice. Other researchers have identified some of these criteria,^{5,6} which include

- *performance suitability*—that is, throughput and disk and memory requirements;
- *transparency*—that is, documentation of specifications or interfaces;
- *functional match*—that is, using function-point analysis or the percentage of requirements delivered; and
- *interoperability*—that is, proprietary interfaces that force the organization to commit to a single vendor for many products.

The selection team must also consider the product's security, safety, maintainability, update cycle, maturity, upward compatibility of revisions, quality, and reliability.

Architecture

Table 1 lists a sampling of technical criteria for a COTS product. Selecting a product that doesn't meet a minimum set of technical requirements could result in increased costs to purchase supporting hardware or in security compromises. In this case, IT managers must decide if they want to grant exceptions to the architecture policy, modify the COTS product to fit the architecture, or choose another solution, such as a different product or even custom development. Each of these options increases risk and likely cost as well. To arrive at the best solution, IT managers should ask,

- Will the benefits of using the COTS product justify the architectural shortcomings?
- Will the vendor enhance the product to meet our technical requirements? Would modifying it introduce unacceptable risk and cost to the project?
- Will another viable COTS product meet the requirements?
- If all else fails, do we have the resources for custom development?

All the healthcare organization's COTS integration projects had significant problems because the COTS products didn't align with specific provisions of the organization's technical architecture. One project selected a COTS package that relied heavily on Microsoft Windows although the organizational standard was Sun Solaris. The organization spent significant resources to modify the product for Sun Solaris.

Another project had to implement a COTS product on an operating system version that the vendor didn't support. The organization wanted all applications implemented on Sun Solaris 9, but the vendor hadn't yet upgraded its product from Solaris 8. The configuration was unsuccessful and caused several months delay.

Lifecycle Costs

IT managers should evaluate the complete cost of acquiring, implementing, and supporting a COTS product. Such an evaluation means accounting for all relevant costs so management can make appropriate investment and budgeting decisions. At a minimum, IT managers should consider the licensing model, vendor assumptions, infrastructure and training requirements, and any additional services the vendor

Table 2. How two projects approached COTS lifecycle costing.

Cost driver	Project A		Project B	
	Action	Result	Action	Result
Understand the vendor's assumptions	Identified the vendor assumption that the organization would provide: the Sun One Web server and Oracle database server for the COTS package.	Project managers planned for this supporting infrastructure.	Overlooked vendor's assumptions that the organization would provide a required SQL server database.	The project struggled to find funding for this extra expense.
Plan for infrastructure improvements	Identified hardware and networking improvements required to support the COTS package, including load balancing and storage requirements.	Project managers included these requirements in the budget request.	Failed to identify a comprehensive set of hardware and networking improvements required to support the COTS package.	The project team had to compromise the system architecture to use available hardware.
Identify training needs	Ensured that the system integrator staffed the project with personnel trained in the selected COTS products.	The team had the expertise to configure the product to user requirements and integrated the product into the organization's data center on schedule.	Didn't budget for training staff on the COTS product.	With no certified staff or access to the vendor's product documentation and help desk support, there were substantial delays and quality problems with configuring and implementing the product.

provides. Table 2 contrasts two approaches to lifecycle costing and their consequences for the healthcare organization's COTS integration projects.

Licensing model. The organization must purchase some form of COTS software license. The cost for the license typically consists of an initial fee and a yearly maintenance fee. The maintenance fee, often 18 percent of the initial fee, gives the organization the legal right to use the product as well as access to product upgrades and help desk support.

Licensing approaches. Vendors have several licensing approaches. An organization can buy one license for each user. (These individual licenses are referred to as seats.) It can also enter into an enterprise agreement, in which the organization buys the right for all its users to use the product. Volume-based licensing requires the organization to pay the vendor on the basis of how much information passes through the product. Finally, in a computer-processor-based licensing agreement, the organization pays the vendor on the basis of processor type and capacity.

Vendor assumptions. COTS software vendors often assume that the acquiring organization will provide supporting software that the product needs to function properly. This software can include Web servers, application servers, and database management systems. If the organization fails to identify explicitly these supporting software products and account for their cost, they risk cost overrun.

Likewise, many large COTS products offer optional modules that could be more mandatory than optional, depending on the requirements. These, too, have additional costs.

Infrastructure and training. To ensure proper system performance, the organization might need to upgrade its hardware and networking environment. The organization must consider the additional cost of these upgrades during the selection process. Training in the COTS product to ensure its smooth implementation and operation is another possible cost.

Additional vendor services. Many COTS vendors sell technical services packages that provide access to experts or upgraded help desk support. These services can be extremely valuable, but

Is Your Vendor Financially Healthy?

If a financial analyst is unavailable, you can still get some idea of a software vendor's financial health. The following checklist should prove useful.

How Is the Stock Performing?

For publicly held firms, research the stock performance and investment community coverage. A poorly performing stock could indicate a company that is a candidate for takeover or bankruptcy.

What's the Bond Rating?

Bond rating services, such as Standard & Poors, Moody's Investors Service, and Fitch Investors Service, rate a company's probability of defaulting on bonds based on an analysis of its financial condition and profit potential (www.investopedia.com). The bond rating can indicate financial health.

Is There Anything of Concern in Financial Statements?

Review the vendor's income statement, balance sheet, and cash flow statement and compare them to those for similar companies. Most public companies post financial reports on their Web sites. Additionally, Internet resources such as Morningstar (www.morningstar.com) and MSN Money (www.moneycentral.msn.com) offer easy access to public companies' financial information and identify industry peers.

Reviewing key financial ratios and financial statement notes is also valuable.

Privately held companies might share financial information if you sign a nondisclosure agreement. Be sure to determine the vendor's history of total profitability and profit margins, as well as if there's a history of positive cash flow and enough cash to service debt. Note anything of concern, such as pending litigation or restatements of earnings.

Where Is the Financing Coming From?

Is the source of operating capital stock issuance, debt, venture capital, or reinvested earnings? Vendors operating on venture capital funding can quickly run into trouble if the venture capitalist withdraws support.

What's the Installed Base and References?

Obtaining the vendor's total installed base can help you validate the quality of the vendor's product and support services and current customers' satisfaction level.

How Is the Vendor Organized?

An organization chart will show that the company is organized sufficiently to support the installation and any required maintenance. Identifying the number of employees will help you approximate operating expenses.

require funding. The organization must consider the costs of acquiring professional services to configure and integrate the COTS product in the evaluation and selection process.

Vendor and Product Line Stability

Selecting a COTS product necessitates some degree of dependency on its vendor. If the vendor goes out of business, discontinues the product line, or is acquired, the organization could have an unsupported COTS product and no access to the source code. If the organization has a software escrow agreement with the vendor, the organization needs to understand and maintain the vendor's source code if the vendor discontinues the product line. This further increases the importance of knowing the product's underlying coding language and technology in the event that the organization must maintain the code. These risks make it imperative to evaluate a vendor's financial performance and the product line's stability.³

Vendor stability. It's impossible to know a vendor's future with any certainty. However, IT managers can and should research candidate vendors and be reasonably confident that the selected vendor can support the product for the system's life. A vendor analysis can help determine a vendor's financial health, as the "Is Your Vendor Financially Healthy?" sidebar describes, but it also helps the IT manager form a picture of the vendor's overall stability.

The analysis should help the acquiring organization determine—at least to some degree—if the vendor will be able to stay in business or if it appears to be an acquisition target. Merger and acquisition activity is common in some vendor markets. When a software vendor is acquired or merges with another company, the vendor's direction and support for the COTS product can deviate significantly.

One of the healthcare organization's COTS integration projects didn't systematically evaluate

the selected vendor's stability. The vendor was a startup company with an innovative product that had the potential to transform one of the organization's core business processes. The vendor was operating through a combination of venture capital funding and a partnership with a prominent systems integrator. Unfortunately, the vendor lost its venture capital funding, declared bankruptcy, and ceased operations. Once this occurred, the project lost its momentum, and the organization had to start a significant replanning effort.

Product line stability. If a vendor determines that a specific product line is unprofitable or no longer supports its business strategy, it might sell that line to another company or discontinue the product. To avoid this risk, IT managers should systematically assess the product line's stability. One action is to determine how much the product contributes to the vendor's bottom line: the smaller the contribution, the more likely the product line is unstable. The vendor's financial statements, which typically reveal the company's revenue sources, should reveal a particular product's contribution.

Another approach that helps characterize product line stability is assessing the growth of the product line's market and the vendor's market share. The more the product line grows and receives customer interest, the greater the chance that the vendor will continue supporting it. Another indicator of the vendor's ability to survive market turmoil and continue to support the organization's use of the product is the scope and breadth of other customers currently using the product and the length of time they've been using it.

Consider a Test Installation

IT managers and business leaders should consider installing the COTS product in a test environment before making the final selection.³ Implementing the product in the organization's environment can expose potential problems integrating the product into the technical infrastructure. It also lets users see a pilot version and validate the selection.

One of the healthcare organization's COTS integration projects didn't take this approach. When the project team began installing the product, it discovered that the application required a

DB2 database that the vendor hadn't mentioned.

Identify Gaps

The critical activity during the design phase is to compare requirements to the out-of-the-box product capabilities and identify any gaps—that is, where the organization's requirements diverge from the COTS product. To close the gaps, an organization can

- modify the business process to align with the COTS product,
- configure the COTS product within the product's intended scope,
- build custom bolt-on functionality to supplement the COTS product, or
- perform more invasive customizations that are beyond the COTS product's scope but deemed critical to success.

The vendor's financial statements should reveal a product's contribution to the company's bottom line.

One of the healthcare organization's COTS integration projects successfully created custom bolt-on functionality that addressed a requirements gap and avoided COTS customization. The solution involved integrating two COTS products, each with a separate user interface. The team built a custom Web portal that gave users one home page to access each portion of the system.

Be Cautious with Data Schema Changes

The project team must carefully consider any changes to the data schema that ships with the COTS product because altering the column and table names can cause the product to behave erratically. Without source-code access, such malfunctions can be difficult or even impossible to fix.

As part of requirements definition, the project team should create a logical data model and data dictionary that reflect user requirements. However, instead of forward engineering a logical model into a physical data model, they should

map the logical model to the COTS product's existing data model. This mapping exercise identifies how the project will use the COTS product's existing data model.

One of the healthcare organization's COTS integration projects had difficulty completing life-cycle reviews with the enterprise data architects. The data architecture group was oriented toward custom development and attempted to impose its data modeling and naming standards on the project. The project management team had to request exceptions to these standards.

Clearly, COTS integration can accelerate system rollouts while minimizing risk and overall cost. However, to realize this potential, IT managers must identify opportunities for COTS integration, recognize the unique characteristics of such integration projects, and partner with business owners to reengineer business processes as appropriate. The organization must be prepared to adjust its typical system development life cycle to ensure an effective COTS integration effort. Expecting to take a COTS integration project through the same systems development lifecycle steps as a custom development project can jeopardize a COTS solution's potential savings and benefits.

The most successful of the reference projects identified a business process that was appropriate for COTS integration, modified the life-cycle methodology for COTS integration, and reengineered the business process to align with the COTS product. The projects that struggled didn't recognize the unique characteristics of COTS integration and adapt their approaches appropriately. ■

References

1. US Office of Management and Budget, *OMB Circular No. A-11*, June 2005; www.whitehouse.gov/omb/circulars/a11/current_year/a11_toc.html, 2008.
2. US Office of Management and Budget, *OMB Circular No. A-130*, Feb. 8, 1996; www.whitehouse.gov/omb/circulars/a130/a130.html#8.
3. Software Eng. Inst., "COTS-Based Systems Lessons Learned Web Pages," www.sei.cmu.edu/cbs/lessons/index.htm.
4. G.A. Thomas and S. Jajodia, "Commercial-off-the-Shelf Enterprise Resource Planning Software Imple-

mentations in the Public Sector: Practical Approaches for Improving Project Success." *J. Government Financial Management*, vol. 53, no. 2, 2004, pp. 12-19.

5. L.C. Briand, "COTS Evaluation and Selection," *Proc. Int'l Conf. Software Maintenance*, IEEE CS Press, 1998, p. 222.
6. Systems Eng. Process Office, "Commercial-Off-the-Shelf (COTS) Evaluation, Selection, and Qualification Process," *PR-SPE-05 V1.0*, Code 212, Space and Naval Warfare Systems Center, 17 Oct. 2002; http://sepo.spawar.navy.mil/COTS_Process.doc.

C. Todd Coutts is a manager at Noblis. His technical interests include the effective management of information technology to support business operations and management decision-making. Coutts has an MS in IT management from the University of Virginia and a Project Management Professional (PMP) certification from the Project Management Institute. Contact him at todd.coutts@noblis.org.

Patrick F. Gerdes is a senior manager at Noblis. His technical interests include systems engineering, enterprise architecture, and assisting clients transform their enterprises through effective decision support systems and initiatives. Gerdes has an MS in aeronautical and astronautical engineering from Stanford University. Contact him at patrick.gerdes@noblis.org.