

Think Outside the COTS

By: Philip Newcomb
September 2007

Abstract: As the need to modernize legacy computer systems accelerates, some Information Technology (IT) managers believe that the approach offering the lowest risk and cost is replacement of the aging system with a Commercial-Off-The-Shelf (COTS) solution. That belief however, needs close examination in light of the substantial information to the contrary.

1.0 Introduction

Much has been made lately of the applicability of Commercial-Off-The-Shelf (COTS) products to answer the needs of an IT community facing the daunting task of modernizing generations of legacy computer systems. Even though most legacy systems contain unique business rules and logic that require functionality beyond the generic scope of most COTS products, the COTS providers are marketing their products as being capable of replacing major portions of a legacy system. The difference between this COTS myth and reality however, is quite different.

The National Science Foundation's Center for Empirically Based Software Engineering (CeBASE) defines COTS as:

“a software product, developed by a third party (who controls its ongoing support and evolution), bought, licensed, or acquired for the purposes of integration into a larger system, which might or might not allow modification at the source code level, but may include mechanisms for customization, and is bought and used by a significant number of systems developers.”

The COTS system development/modernization process typically leverages the inherent functionality of each component, extends some part of that component's functionality to meet specific local needs, and then integrates those disparate components to create a new system.

COTS providers claim that their products have been proven in the competitive marketplace, thus they are highly capable, reliable and contain functionality superior to custom-built components [1]. In addition, COTS components are marketed as being easily integrated, functional in a multitude of computing environments, and easily extended and tailored to meet any requirement. Available data however, when either using these products and processes to create new systems or replace legacy systems does not appear to support such claims.

Three separate studies by The Aerospace Corporation, the National Research Council of Canada, and the Canadian Department of National Defence have shown the inherent weaknesses and risks of using COTS products to develop new systems. The “lessons learned” from those studies were quite similar and best summarized by the Mitre Corporation effort that resulted in the matrix shown in Figure 1 [2].

COTS Myth	COTS Reality
Less expensive development costs	<ul style="list-style-type: none"> • Higher integration costs & risks; more up-front testing • COTS acquisition costs – licenses & fees
Lower life-cycle costs	<ul style="list-style-type: none"> • Maintenance of COTS products required • Software modification can be difficult & costly • Product upgrades may include needed functionality, but may be incompatible with prior releases & other software
Shorter development schedules	<ul style="list-style-type: none"> • Inexperienced integrator will require learning curve • Schedule must include product upgrade integration time
Functionality exists in COTS	<ul style="list-style-type: none"> • Software may only partially satisfy requirements • Software may satisfy unnecessary requirements • Limited requirements influence, risks vendor failure
Application portability/interoperability	<ul style="list-style-type: none"> • Not guaranteed, even among similar operating systems • Ease of interfacing proportional to system complexity
Higher reliability	<ul style="list-style-type: none"> • COTS products still contain bugs, possibly malicious code • Developer must rely on COTS vendor to fix problems • Difficult to determine source errors • Vendor not liable for failure
Better user interface	<ul style="list-style-type: none"> • Interfaces unmodifiable for generic user communities
Complete/true documentation, manuals	<ul style="list-style-type: none"> • Variable documentation, undocumented features
Vendor service available	<ul style="list-style-type: none"> • Variable support, training, integration

Figure 1: COTS – Myth vs. Reality

Even under the most successful COTS system development scenarios, The Aerospace Corporation’s Computer Systems Division came to the conclusion that any benefits achieved using a COTS solution to develop a new system can only be attained after careful preparation and execution, taking into account the complexities of preparing:

- To make adjustments to the acquisition and development processes;
- For inherent cost, schedule and performance risks beyond government or contractor control; and
- For a complex development and sustainment effort [3].

In many cases COTS products have proven to be a risky solution to new system development. It is much more difficult to successfully implement a COTS solution than the vendors would have you think.

2.0 Legacy System Modernization Uncertainties

Based on available empirical data about the risks and problems associated with the use of COTS products and processes to develop new systems, how is it then possible that these same products and processes are now being touted as the answer to the legacy system modernization problem? How is it that executive staffs and their supporting IT managers are allowing themselves to be sold on the concept that a generic product, with limited applicability to a program office’s specific mission, can replace their existing legacy system?

The answer to those questions is marketing. COTS providers tell a very compelling story that is easily understood by an inexperienced buyer. However IT managers must better understand the total COTS alternative before selecting a COTS solution as their preferred legacy system modernization pathway. If they don't understand that picture, they will potentially be introducing significant uncertainties into their modernization project. Seven key areas of uncertainty are cost, design, performance/functionality, schedule, control, security and risk. The issues surrounding these uncertainty areas are:

Cost - The initial comparison of the cost of maintaining and modernizing a legacy system to the cost of that system's replacement with a COTS solution is typically an apples-to-oranges process. The total life cycle cost of the legacy system is well known and documented. The cost of the COTS solution, while well defined on the producer's price sheet, has many hidden costs that are not easily quantified. These costs include:

- Specific hardware upgrades required to enable the COTS product to run;
- Customization of the COTS product to address missing functionality, including hardware upgrades resulting from needed software upgrades; and
- COTS product integration, installation, support, training and documentation.

Given the difficulty in calculating these costs at the outset of a project, cost overruns can impact if not totally derail a project. A dramatic example of such an overrun has occurred in our nation's air traffic control system. According to the U.S. House Committee on Transportation and Infrastructure, the COTS-based system modernization effort was originally estimated to cost \$2.5 billion. At the time of the reports publication however, the Federal Aviation Administration (FAA) had already spent \$35 billion with a decade more spending anticipated before the system is fully operational [4].

Design - If an IT manager opts for a COTS solution, then the design of the future system becomes the next major issue. The design activity is driven by the need to define the performance and functionality requirements to be replaced and enhanced, and by the selection of a COTS product or products that best meet those requirements. Of primary concern is defining the integration process between the COTS product and any new software to be developed. That integration process is particularly demanding when several COTS products are involved, each with possibly different architectural styles and constraints [5]. Duplication of functionality between multiple COTS products can further exacerbate the integration process.

The Navy suffered the cancellation of the entire "Smart Ship" program when the guided-missile cruiser USS Yorktown went dead in the water for two and a half hours after a data entry error set off a cascade of system failures in the array of COTS products that had been designed to operate as an integrated system. This mishap was so serious that top Navy leaders changed course in the program and substantially reduced the Smart Ship COTS content in the final system design [6].

Performance/Functionality - COTS products' performance and functionality have often been overstated in the provider's literature and that performance and functionality can be either a mismatch with the project's requirements or overkill for those requirements. Then, when a performance or functionality shortfall occurs, the IT manager has limited access to the products source code, if they have access at all, to modify the product to meet the project's needs. And if it is possible to obtain the source code of the COTS product, the IT manager will have to pay a large sum of money to the vendor for that access [2]. Thus, an IT manager may be forced to live within the limitations of the COTS product as designed and not achieve the system performance and functionality that is truly required.

The Navy experienced such performance and functionality problems with the operator workstations aboard the E-2C aircraft. When many of the units were installed in the system, they were not functional. It wasn't because they were broken – they had passed acceptance testing. The problem occurred because changes to specifications occurred in the on-board systems that made the COTS product incompatible, requiring a full year for the vendor to perform a sweeping analysis of their sub-vendors before the problem was rectified. Database issues were also an E-2C problem due to commercial databases as the transaction speed needed in the commercial world don't remotely meet the transaction speed needed to support a tactical situation.

The E-2C project manager's final perspective was that the military's needs are not only more demanding than commercial needs, but also exceed the commercial industry's understanding of the concept of speed and performance [7].

Schedule - As found during the Mitre study, COTS providers often sell vaporware through premature product announcements because of the need to pre-position their product in the marketplace [3]. The actual release date of a functioning product can be quite different than the announced date. In addition, the larger market and not the specific needs of a project are what drive COTS providers to upgrade and fix their product. Should technical issues arise with regard to inherent product weaknesses in the course of a project, the IT manager becomes captive of the COTS provider's long-term product development plans as opposed to their project's near-term needs.

Project schedules are therefore impacted at the outset of a project because the COTS product isn't ready for prime time. Later, the project can suffer equally deleterious schedule impacts when the provider does not address problems in a timely manner. The United Kingdom ran into serious COTS schedule slippage when modernizing their en-route air traffic management system called NERC. The COTS implementation, scheduled for completion in 1996, didn't go operational until 2002 because of COTS-related delays [8].

Control - To have a successful modernization project and to ensure that the modernized system can be easily and inexpensively sustained in the future, an IT manager must have control of the entire systems' components. COTS solutions however, are outside the control of the system user. With COTS providers being driven by the far larger commercial marketplace, IT system developers will take a back seat when it comes to getting the providers to address the product's deficiencies and the support requirements that will always be unique to their system.

As has happened in the past, the COTS provider may go out of business leaving the IT manager entirely without support. Even a thriving provider may drop or de-emphasize a product, change the type of support it provides to that product or change or drop promised features, performance and upgrades, depending on prevailing market conditions, without warning [3]. The possibility of any one of these events occurring leaves the IT manager completely at the mercy of the COTS provider.

Lack of control of COTS software has seriously hampered the efforts of the Navy to have two vital air defense systems interoperate. AEGIS systems are used to help defend vessels against air attacks, and the Cooperative Engagement Capability system gathers and shares radar data with multiple ships. Interoperability has been so poor however, that the fleet commander didn't believe he was seeing the right battle picture.

As pointed out by the Navy, the problem is not the individual systems, but their interoperability compounded by the COTS display systems. The Navy stated that the COTS is the more challenging problem because while they license the product, they don't have access to the source code, so they can't "get under the hood" to fix the problem [9].

Security - System security, which is always of paramount concern, is a major issue to be addressed by the IT manager. All one need do is read the newspaper each day to find the latest security incursion being perpetrated on a COTS software product. The very openness of COTS products makes those products easy targets to any number of threats. In today's world where Homeland Security is the watchword, COTS products are targets for hackers, commercial thieves, disgruntled employees, political dissidents, and even organized crime. For the military user this threat list can be expanded to include cyber-terrorism and espionage [3]. Thus, the threat to system security is more than a potential loss of business revenue; it is a danger to national security.

Ron Lewis, a system security specialist for CDO Technologies, Inc. of Dayton, Ohio, states that while performing bandwidth assessments at a U.S air base in Italy in March 2000, network engineers watched bandwidth utilization on a Fast Ethernet LAN segment of the enterprise system suddenly spike and peak at 90%, while throughput dropped from 30 Mbps to 7 bps.

To determine why there were large scale network outages being reported across the base, CDO Technologies used their protocol analyzers to scan all the network traffic and watched as the majority of the network began sending large amounts of data. They quickly determined that they were witnessing firsthand, the impacts of the newly released LoveBug virus.

Fortunately, the virus was generally non-malicious in nature. It consumed bandwidth, but it did not re-format hard drives, scramble BIOS, or otherwise destroy hardware. But what if it had? What if cyber-terrorists had exploited known vulnerabilities and targeted the base's critical infrastructure? Lewis speculated that the entire base enterprise system would have been taken hostage given the vulnerabilities of the COTS product and its widespread deployment throughout the base [10].

Risk - The issue of project risk is really a compilation of the difficulties encountered in the first six areas of uncertainty. Yet risk is usually foremost in an IT manager's mind when trying to select the optimum system modernization solution. To avoid risk, which can make a shambles of the most carefully conceived project cost and schedule estimates, an IT manager will often accept a solution, be it more expensive and requiring a longer duration to execute, if they can be assured of a project's success within the postulated plan.

COTS solutions, because of the inherent risks they introduce to a project, significantly increase the possibility that a modernization project will exceed planned estimates or fail completely. As an example of the risks associated with using COTS solutions to address legacy system replacement, Accenture's experience has shown that 96% of COTS users did not achieve all the benefits they had expected [11]

With so many uncertainties and risks associated with using COTS products to modernize or replace legacy systems, IT managers must better understand the available alternatives to the COTS solution, with an eye toward using the latest technological breakthroughs to best address their modernization needs.

3.0 Legacy System Modernization Alternatives

The modernization of a legacy system is a complex undertaking. How a program office chooses to address that process has a significant impact on the cost and schedule of the project, as well as an array of ancillary issues such as system functionality and performance, user training, maintenance, and future system enhancement.

The need to modernize legacy systems is primarily driven by three factors:

- Expansion of the system's functionality;
- Improved maintainability of the system using modern tools and techniques; and
- Reduction of operational costs and improved reliability by replacing obsolete hardware suites with high-speed, open-architecture systems.

Alternative solutions to COTS for modernization of the system include:

- Developing a new system;
- Manual rewriting of the legacy applications software and databases to operate within a modern computing environment; or
- Automatic transformation of legacy applications software and databases to operate within a modern computing environment.

Developing a totally new system or replacing legacy systems by manually rewriting the system's software is costly and time consuming. According to Gartner Group findings, legacy system modernization using manual methods can cost between \$6 and \$26 per line of code and thus far, such projects have a success rate of approximately 7%; a success rate that has not bred confidence within the IT community [12].

Using semi-automated system transformation tool-based solutions, while relatively promising, have not provided a sufficient level of automation to overcome the drawbacks associated with manual intervention required to address untransformed code. Once the untransformed code has to be developed, all the shortcomings of the manual approach come into play. And once developed the IT manager has to address code non-uniformity during the integration process.

A very strong case can now be made for the use of fully automated system transformation of code as the best solution to legacy system modernization. Through the application of a suite of newly available tools, it is now possible to achieve nearly perfect levels of automation, thus dramatically improving the value of the automated approach to legacy system modernization through transformation.

Automated transformation focuses on fully automated transformation and migration of the legacy system’s application code and database structure to achieve a functional equivalent of the original system and re-hosting in a modern computing environment. From this baseline, the IT manager can continue operations without the disruption of user and support personnel retraining, and can more efficiently begin to expand the system’s functionality as required.

As shown in Figure 2, automated transformation has been successfully used to modernize a wide variety of system applications, across an array of source languages at levels of automation exceeding 99%.

Legacy Source Code	System Applications
COBOL	<ul style="list-style-type: none"> ● Administration ● Logistics ● Finance ● Health Maintenance
Ada	<ul style="list-style-type: none"> ● Aircraft Display ● Satellite Terminal ● Mission Planning
Assembler	<ul style="list-style-type: none"> ● Strategic Missile Defense
Jovial	<ul style="list-style-type: none"> ● Satellite Tracking ● Aircraft Control ● Command & Control
CMS-2	<ul style="list-style-type: none"> ● Tactical Weapons Display
FORTTRAN	<ul style="list-style-type: none"> ● Weather ● Strategic Warfare Planning ● Strategic Missile Defense
C	<ul style="list-style-type: none"> ● Homeland Defense
VAX Basic	<ul style="list-style-type: none"> ● Shipboard Engineering
MUMPS	<ul style="list-style-type: none"> ● Hospital Operations
P/L 1	<ul style="list-style-type: none"> ● Commercial Airline Operations

Figure 2: Successful Automated Transformation Applications

Specific examples of performance metrics for projects using this technology include:

- A 40 to 1 reduction in functional testing time for a Jovial transformation task of 250,000 lines that enjoyed an automation level of 99.98% and
- A functional test of 560,000 lines of COBOL transformed at 99.99% that only identified 250 “bugs” during functional testing, many of which were resident in the original COBOL system [13].

Being that an automated tool created the newly generated software, it also has the benefit of consistent quality and uniformity. Systems comprised of large quantities of code, if addressed manually, will require many programmers. Programmers, though writing code in the same language, have different styles. Those stylistic differences can create major difficulties during the system integration and testing phase of a transformation project. A highly automated approach requires negligible manual intervention, offers a solution that facilitates the uniformity of the code and thus, compresses the integration and testing schedule for the project.

4.0 Conclusion

The reality of legacy system modernization projects is that often the legacy system contains business rules and functionality so unique, that COTS products cannot meet the needs of the IT manager and the system’s user community. The functionality of the COTS product will provide only a percentage of what is needed, requiring that the IT manager redevelop the missing functionality at great cost, over an extended period of time, and at significant risk. When the missing functionality is developed, the IT manager has to successfully integrate that new functionality with the COTS provider’s proprietary code, to which that manager has limited access or control.

Many legacy systems are too unique to be modernized using a generic COTS solution. It is far better to modernize such systems using proven automated tools capable of preserving the business rules and functionality of the legacy system that historically has served the specific needs of the user community. By employing available low-risk, low-cost transformation tools to modernize a legacy system as opposed to trying to replace it with a COTS product, the new system’s functionality will be identical to the original system, but operate in a modern computing language and on a modern platform-independent computing environment.

Automated transformation solutions also ensure that no training is needed for the system operators or maintenance staff, since the system operates the same way it did in its legacy state. And once in the modern computing language and environment, the IT manager can far more easily modify and/or enhance the system’s functionality by using modern code development tools and by finding COTS products that can be easily added to the system at low risk.

COTS solutions are not the legacy system modernization panacea that many executive staffs and their supporting IT managers are being led to believe. COTS solutions have proven to be high-risk, costly and prone to failure. As the need to modernize the vast array of aging computing systems accelerates, it is imperative that the managers of modernization projects and the executives they report to, not be drawn towards what

appears to be, the quick and easy answer of using COTS products to replace their legacy systems. They need to perform a far more in-depth investigation into all the risk-related issues surrounding the use of COTS products to replace legacy systems, and into the more successful alternative solution of automated transformation, before leaping to a quick and easy answer that they will pay for dearly down the road.

5.0 References

1. Canadian Department of National Defence' Defence Research and Development Branch, "Using COTS Software in Systems Development", 1999.
2. MITRE Corporation, Dr. Mark Maybury, "What Rots about COTS: Hidden Risks of Commercial Software", 1999.
3. The Aerospace Corporation, Richard Adams and Suellen Eslinger, "COTS-Based Systems: COTS Software Lessons Learned, Recommendations and Conclusions", 2000.
4. Young D. (Chairman), "Modernization Of Air Traffic Control Programs: Cost Overruns & Delays Have Plagued FAA's Efforts". U.S. House Committee on Transportation and Infrastructure Hearings October 2003.
5. Morisio, M. and Sunderhaft, N. Commercial-Off-The-Shelf (COTS): A Survey. University of Maryland 2000.
6. Walsh, E.J., Navy Postmortem Tries to Pinpoint What Went Wrong With the 'Smart Ship'. Military & Aerospace Electronics March 2001.
7. Campbell, L., COTS: Is It Just a Check Mark for Your Program?. Defense Acquisition University March-April 2000.
8. Ladkin, P.B., "COTS versus Bespoke ATC Systems. Safety Critical Mailing List Archive 2002.
9. Stutz, M., Navy Software Dead in the Water. Wired News July 1998.
10. Lewis, R.A., CDO Technologies, Inc. Interviewed February 5, 2003.
11. Walker, D. Look Carefully into COTS. Next 2003
12. The Gartner Group Forecasting The Worldwide IT Services Industry: 1999. The Red Herring 2000.
13. Newcomb, P. and Doblar, R. Automated Transformation of Legacy Systems. The Journal of Defense Software Engineering (CrossTalk) December 2001.

6.0 About the Authors



Philip Newcomb, President and Chief Executive Officer

Internationally recognized as an expert in the application of artificial intelligence and formal methods of software engineering, he has published numerous papers in his field, and has degrees from Carnegie Mellon, Ball State, and Indiana Universities, and the University of Washington. He formulated the conceptual product framework and developed the transformation technology now residing in the *JANUS Studio*TM toolset described herein.

Philip Newcomb - philip.newcomb@tsri.com
The Software Revolution, Inc.
11410 NE 122nd Way, Suite 304
Kirkland, WA 98034
(425) 284-2790
(425) 284-2785 (Fax)
www.tsri.com