# Rules of Thumb for the Use of COTS Products

Michele Motsko
Patricia Oberndorf
Ellen-Jane Pairo
James Smith

*December 2002*

TECHNICAL REPORT
CMU/SEI-2002-TR-032
ESC-TR-2002-032

Carnegie Mellon
Software Engineering Institute
Pittsburgh, PA 15213-3890

# Rules of Thumb for the Use of COTS Products

CMU/SEI-2002-TR-032
ESC-TR-2002-032

Michele Motsko
Patricia Oberndorf
Ellen-Jane Pairo
James Smith

*December 2002*

**COTS-Based Systems Initiative**

This report was prepared for the

SEI Joint Program Office
HQ ESC/DIB
5 Eglin Street
Hanscom AFB, MA 01731-2116

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

FOR THE COMMANDER

Jay Minsky

Jay Minsky
Contracting Officer's Representative

This work is sponsored by the U.S. Department of Defense. The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense.

For information about purchasing paper copies of SEI reports, please visit the publications portion of our Web site (http://www.sei.cmu.edu/publications/pubweb.html).

# Table of Contents

# List of Tables

# Abstract

More and more organizations are realizing the benefits—and sometimes the necessity—of incorporating commercial off-the-shelf (COTS) products in the systems they acquire and use. But COTS products are not necessarily appropriate for every system. When is it wise to pursue a COTS-based systems approach, and when is it best to hold back? How can sound COTS-based system practices be reconciled with an organization’s regulatory and policy constraints? This report provides some information to help guide these decisions.

# Executive Summary

More and more organizations are realizing the benefits—and sometimes the necessity—of incorporating commercial off-the-shelf (COTS) products in the systems they acquire and use. But COTS products are not necessarily appropriate for every system. This report provides some information to help guide decisions about when COTS products are an appropriate solution—and when they are not.

There are many myths and misplaced expectations involving COTS-based systems. This makes it difficult to make the necessary decisions. Here we provide seven "rules of thumb"—general guidance for making an early assessment about whether or not it's smart to attempt the development of a system through the use of COTS products. With each "rule" we provide a discussion to help you in applying the rule. These seven rules are

Rule 1: Find the ways in which COTS products can help you.

Rule 2: Determine the regulatory, statutory, and policy constraints on your system to decide whether they make the use of COTS products infeasible.

Rule 3: Establish whether your system is subject to extreme performance requirements (e.g., security, safety, real-time) and whether they exceed the capabilities of the COTS products.

Rule 4: Decide whether the requirements and users' business processes are flexible enough to accommodate COTS products.

Rule 5: Determine your ability to leverage the marketplace.

Rule 6: If you are developing your COTS-based system by evolving an existing architecture, examine that architecture for its resiliency and its ability to keep on evolving.

Rule 7: Generate a gross cost/benefit analysis.

While these seven "rules" will help you think about some basic technical reasons for using—or deciding not to use—COTS products in your system, they do not cover all the elements that are necessary for success with the endeavor. There are several factors that provide addi-

tional insights concerning such things as your ability to take a spiral approach, to form partnerships with vendors, to incorporate industry standards, to balance your need to keep up with the marketplace while maintaining system stability, and to understand (at all levels in your organization) and respond to the wide-ranging implications of a COTS-based systems approach.

# 1 Introduction

More and more organizations are realizing the benefits—and sometimes the necessity—of incorporating commercial off-the-shelf (COTS) products in the systems they acquire and use. While this is not unique to the federal government (including the DoD), it certainly is prevalent there. Managers who find themselves in this situation are often poorly equipped to make even what seem to be basic decisions about moving to a COTS-based systems approach. This is in part because of a lack of good information and a wealth of misinformation.

## 1.1 Myths and Misplaced Expectations

Expectations for the benefits of using COTS products tend to run high. After all, we have gotten far away from handcrafting items for our use in almost every other aspect of our lives. Why can't we do the same for our software-intensive system needs and have it work just as well?

One reason we can't is that the field of software engineering with COTS products is not nearly as mature as the rest of software engineering—and the rest of software engineering is not nearly as mature as most other engineering disciplines. Another is that there is no universal software architecture that is suitable for all systems or to which all COTS products subscribe. Perhaps most significantly, there are very few fields of endeavor supported by software systems that are so time-tested that all agree on what they should do and how they should do it.

Even accounting, a field that has certainly been around as long as there has been money, incorporates multiple approaches and practices.  Thus for the mature accounting field, we find that the marketplace offers several different financial management packages, and none of them will suit everyone's needs.  Commercial industry, defense contractors, federal, state, and local governments, school districts and universities each have distinct requirements for their financial management.

But at least financial management is well enough defined that vendors can successfully find a large base of users who are willing to buy the vendors' solutions. In less mature fields, such as satellite communications and missile guidance, fewer broadly established practices and characteristics exist. In addition, since these are more specialized fields, they yield a smaller user base. These conditions combine to diffuse the market niche, making it harder to find products that will match your needs.

All this does not stop the leaders in industry and government from taking it on faith that the use of COTS products is always a good idea. With this assumption usually come several misguided notions, such as

- COTS products will save money.
- Because it’s COTS, we don’t have to test it.
- Because it’s COTS, we don’t need to worry about any architecture or engineering.
- Because it’s COTS, there will be nothing to maintain.

While we could put together a much longer list of myths and misplaced expectations, suffice it to say that there is evidence of a great deal of naïveté. Those who want to be successful using COTS products in their systems will need to overcome that naivete—in themselves and in those with whom they work.

## 1.2 Further Complications

To make matters even worse, these misconceptions also mean that leaders have been slow to realize that acquisition and the use of COTS products affect each other. The rules for system acquisition are affected by the use of COTS products, and the incorporation of COTS products is affected by the rules for system acquisition.

# 2 Rules of Thumb

So what's a manager to do? How can you tell whether using COTS products is right for your system? What's needed are some rules of thumb—general guidance for making an early assessment about whether or not it's smart to attempt the development of a system through the use of COTS products.

With each "rule" we provide a discussion to help you in applying the rule.

## 2.1 Seven Rules of Thumb

### Rule 1: Find the ways in which COTS products can help you.

***Stick to your knitting—let COTS do the heavy lifting.***The first step is to find all the reasons that a COTS-based systems approach is *right* for your system. There are many contributions that COTS products can make to your endeavor, and they must be taken into account.

COTS products can often perform some rather mundane but necessary functions. There is little reason to recreate these products if the existing ones suit your needs.  Few organizations would consider developing their own operating system, database management system, or word processing software. Allowing COTS products to perform the functions they do best enables you to concentrate on what you know best—the features and functionality that make your system unique and cannot be bought in the commercial market.

Another reason to use COTS products might be that the technology needed is too advanced and specialized for the organization to undertake development. The organization may not have the expertise to reinvent, much less improve upon, an existing product that has the advantage of being user-tested.

Even if you had the expertise to reinvent existing technologies, you probably don't have the time. Many years have been spent developing a technology and the products that express it. One of the most compelling reasons for using COTS products is that they are available and ready to use; certainly they will have to be integrated into your system to achieve your purposes, but the delays involved in that are, by and large, insignificant compared to an attempt to recreate the technology or to create your own components.

So, will COTS products fulfill at least the main requirements of the system? Of course, for highly specialized applications, this may not be the case. But for most, there will be some or many products to evaluate. Determine which ones best suit your needs and where they fall short. A gap analysis can help you in this task. One strategy is to use a matrix to show the requirements versus the COTS products. Note the degree to which each product meets the requirement. It is important to analyze several product options because compromises may have to be made in regards to requirements and cost. Also, consider supplemental products and combinations of products to meet the system needs.

By now you're no doubt wondering why the first argument offered here was not cost. There is little compelling evidence that using COTS products is guaranteed to save you money. However, in some cases there may be substantial savings. For example, one program's members had tried several times (unsuccessfully) to create their own technology in a given area of interest to them. In the end, they did not have a sufficiently large research budget to pull it off. They only succeeded in fielding a system when they partnered with another organization with a similar interest to base its system on COTS products. In an environment of limited budgets, you may not be able to afford *not* to use COTS products.

Finally, people usually view things through the lens of what they already know. Project personnel frequently see custom development as desirable because it is the known alternative, with familiar risks and hurdles. Sometimes this viewpoint considers custom development through a fuzzy memory of the problems that borders on nostalgia. Forgotten are all the schedule slips, the software bugs, and the requirements reductions that were necessary to be able to field anything close to what was originally envisioned. When suffering from this state of diminished memory, a new, unfamiliar approach using someone else's products can seem very risky and undesirable. This bias needs to be consciously overcome.

### Rule 2: Determine the regulatory, statutory, and policy constraints on your system to decide whether they make the use of COTS products infeasible.

***It's hard to dance when they're holding your feet to the fire.***

There are two ways in which regulations, statutory limitations, and corporate policies affect your ability to develop a successful COTS-based system.

By law or policy, some systems must adhere to certain practices, data formats, outputs, security procedures, interoperability capabilities, and so forth. If some of these requirements are not implemented in available COTS products, they cannot be used. For example, one federal agency is required to adhere to the Joint Financial Management Improvement Program (JFMIP) practices. Before determining that a COTS solution was right for them, agency members first had to assure themselves that the products they looked at could fulfill this requirement. If such limitations or policies affect your situation, you must be certain that COTS products exist that embody the capabilities that are legislated.

The other kind of effect is found in regulations and statutes that haven’t caught up to the new COTS-based system development and maintenance paradigm yet. These affect your ability to approach a COTS-based system with practices that will be successful. For example, a spiral approach is suggested for COTS-based systems, but following one can be hard if your regulations or policies demand that requirements be finalized or a long-term budget be submitted at a point that is too early in the system development.

Another manifestation of this effect is review cycles for securing funding that demand knowledge and predictions so far out in the future that they easily span not only multiple product upgrades, but also significant technology advancement cycles. Another way in which funding cycles can be a problem is that vendors may unexpectedly offer a discount on products you need (usually in response to the need to enhance their bottom line at the end of a quarter or fiscal year). But stodgy funding cycles can mean that you do not have the funding resources necessary to take advantage of the offer. In today’s climate, creative approaches may be required.

Another manifestation is acquisition processes that require the premature preparation of justifications, long-range estimates, or other decision-making items. For example, one organization required the preparation of a “Sole Source Justification” based on market research and evaluation. Without realizing it, the organization was also requiring premature decisions about the high-level system architecture and design, which in turn constrain which COTS products are even eligible for consideration.

Some specific examples of these manifestations are discussed in the annex.

NOTE: Sometimes the regulations or policies don’t *really* demand a particular capability or procedure. The problem is that we are accustomed to interpreting them *as though* they demand it. Old habits and thought patterns die hard. Some regulations are obsolete, and no one has bothered to revoke them. In other cases, regulations or policies are misinterpreted. Question and investigate regulations that don't make sense or seem unusually prohibitive. Finding a solution often takes creativity … and maybe a bit of luck.

### Rule 3: Establish whether your system is subject to extreme performance requirements (e.g., security, safety, real-time) and whether they exceed the capabilities of the COTS products.

***You can't build a space shuttle from things you find lying around the house.***

Some non-negotiable system requirements are too extreme to be readily addressed by COTS products. In particular these are the kinds of performance requirements that are not easily isolated or "added on"—they are generally integral to the system and must be fundamentally addressed in the architecture and design.

But don't use this as an excuse not to give the COTS marketplace a chance. To investigate this aspect, look at the kinds of systems in other walks of life that bear a resemblance to your system in performance, and then explore the commercial products that may be in use for these analogous systems. For example, there may be some common safety concerns—and therefore some ability to find and use common COTS products—between a nuclear submarine and a nuclear power plant. But if you are really out there all alone in some performance parameter, a thorough survey of what is available in the marketplace could lead to the conclusion that a COTS solution is not appropriate for that portion of your system.

If you have extreme performance requirements for which you cannot find COTS solutions, do not automatically abandon the COTS marketplace. Start a dialogue with appropriate vendors; find out if there is any interest or—better still—initial work in a direction that might eventually satisfy your needs. You may not be able to use COTS products today, but you might be able to establish a foundation for your system that gives it the potential for evolving to the use of COTS products in the future.

### Rule 4: Decide whether the requirements and users' business processes are flexible enough to accommodate COTS products.

***Frozen requirements don't suit COTS-based systems: sometimes trying to sculpt ice leaves your creation shattered.***

System requirements may be driven by regulation, statute, or organizational policies. But usually, requirements for a system are driven by the capabilities of the existing system or the expectations of the end users and customers. Users have been known to say, when asked what they wanted the new system to do, "I want it to be exactly like the old system," leaving the poor requirements elicitor to wonder why they wanted a new system at all! If no COTS product fulfills the requirements, users and developers need to reevaluate what the real necessities are.

In general, every system has some truly immutable requirements—capabilities it cannot do without, whether they are driven by statute or physics or the characteristics of a mission or objective. These are non-negotiable and must be satisfied.[1] However, it is usually the case that the majority of the "requirements" placed on a system are more flexible—you could call them "desirements." They are negotiable; some may be only slightly negotiable, while others may be purely "nice to haves" that do not have to be satisfied at any level. If there are too many non-negotiable requirements, then it will be nearly impossible to find COTS products that can meet the demand. You will specify yourself right out of the COTS marketplace. The requirements have to be categorized according to whether they are negotiable or non-negotiable; negotiable ones have to be prioritized, preferably by the user.  If all non-negotiable requirements are met, a COTS solution could still be the right one.

Keep in mind that every COTS product is created based on the developer's concept of how it will be used. It may be explicit or implicit, but it is always there. If the product's process or mode of use does not agree with the end users' processes, then the system will fail.  If altering a business process to conform to a COTS product does not make sense or is extremely disruptive, a COTS approach may not be the best choice.

Another source of difficulty is that many organizations have a very unresponsive process for changing requirements once they've been established. In the COTS marketplace, change happens quickly.  Some changes might be improvements over the established requirements, but without a nimble process for agreeing on such requirements, evolution of a COTS-based system may not be an effective solution in the long run.

One key to success with COTS-based systems is to reconsider the end users' processes in light of what is available in the marketplace. If everyone understands—and accepts—from the start that the users' processes and the products' processes will not match perfectly and that compromise and change may be necessary, then the resulting system has a much greater chance of succeeding. It might be worth your while to invest in technical change management and business process re-engineering (BPR). Another success key is the use of a true spiral process in which the requirements evolve along with the system, rather than being immutably established at the beginning.

[1] Just because certain demands must be satisfied, however, does not mean that a COTS product needs to be modified. It may be that an additional product can be found that makes up for what would otherwise be a system deficiency. Or it may be that the missing functionality can be provided in an entirely custom-built component. In general, it is a bad idea to modify COTS products (i.e., change their source code) or perform other customizations that were not anticipated or provided for by the vendor.

### Rule 5: Determine your ability to leverage the marketplace.

***COTS-based systems are like investing in the stock market: if you don't make the effort to do your homework, you may lose very badly.***

Your ability to leverage the marketplace depends on how closely your system needs match the COTS products that are available.  Most COTS-based systems are in fact hybrids of many different kinds of components—COTS products, legacy components, nondevelopmental items obtained from sister organizations or previous projects, custom-developed components, and so forth. The use of COTS products is not an all-or-nothing proposition for any system.

One key to your success will be your ability to determine which parts of your system lend themselves to exploitation of the marketplace. This is only possible through knowledge of the marketplace and participation in it to make vendors aware of the needs you have that may not currently be addressed. Together, knowledge and participation spell leverage.

To truly leverage the marketplace takes considerable knowledge about it—what market niches are relevant to your system, how the marketplace works, what's hot (and what's not), what motivates the particular vendors, and so forth. To find out more, particularly about the market niches that apply to you, you will need to get familiar with what is out there. A market watch activity will help to clarify what the marketplace provides that you should be exploring. Your ability to leverage the marketplace is directly related to your ability to acquire the needed marketplace knowledge; this in turn depends on how early you are in the system life cycle and how much time and money you have to gain the marketplace knowledge and participation needed.

### Rule 6: If you are developing your COTS-based system by evolving an existing architecture, examine that architecture for its resiliency and its ability to keep on evolving.

***If you're going to modernize the house, you'd better be sure it has a sound foundation.***

The constant activity in the marketplace puts a lot of pressure on a system and its architecture. If the plan is to change out parts of the existing system and replace them with comparable COTS products, then it is necessary for the system architecture to be able to withstand this kind of change and evolution.

The architectures of many custom systems have not been given much attention; there is a tendency to focus more on the implementation. However, with a COTS-based system, your only tangible asset and means of dealing with change is the architecture. It must be evolvable. If the plan is to evolve an existing custom system into a COTS one, then the existing architecture must be up to not only the immediate task but also to the long-term evolution of the system after all of the COTS components have been introduced. If you just make the assumption that the old architecture will support a COTS-based approach, you are likely to end up with only incidental use of COTS products and derive none of the touted benefits. COTS-based systems must evolve. Their architectures are key to that evolution.

### Rule 7: Generate a gross cost/benefit analysis.

*If the use of COTS products doesn't make business sense over the life of the system, then don't do it.*

Even if all the other rules are satisfied, there is still a question of whether a COTS solution makes business sense in a given situation. Many people believe that using COTS products will automatically save them money, but that is not necessarily the case. Some projects using COTS products are finding a lifetime savings of only 5%–10% over their old solution.

Be sure when creating a COTS cost/benefit analysis that you truly compare the total ownership cost of all alternatives considered. It is necessary that you compare development as well as sustainment costs.  You must consider all elements that each will take, for example, all programming costs, whether for creating functionality or for  integrating the units. Also consider the success criteria and their relationship to the projected benefits for all alternatives. You cannot be selective and still create a valid analysis.

To approach a gross cost/benefit analysis, you must realize that the benefits may come in forms other than dollars, and so may the costs. You may conclude that there is no way you have the ability to satisfy the system need with a custom solution—perhaps you don't have the skilled personnel that are required, or it will take too long. You may find what one program's members did: that they could never get a budget that was large enough to fund the necessary research and technology development themselves. Another program focused primarily on the time frame: the only way to have the needed capability in time was to rely on components that already existed—that is, COTS components. A less tangible, frequently unanticipated benefit may be the ability to keep synchronized with the marketplace, which yields such benefits as improving interoperability, exceeding proposed requirements, or reducing cycle time throughout the life of the system.

The process of constructing a COTS-based systems cost/benefit analysis is no different from the one you already use.  However, some of the factors that you consider will be different. Some of these are discussed in Section 4.

# 3 Other Success Factors You Should Know About

But are these seven basic rules enough? Actually, no. While they will help you think about some basic technical reasons for using—or deciding not to use—COTS products in your system, they do not cover all the elements that are necessary for success with the endeavor. The following provide some additional insights. Another source of further guidance and food for thought along these lines is found in Carney [Carney 98].

### Factor 1: Do you have the ability to break the overall job into tractable pieces and focus on each one somewhat separately?

A “big bang” approach does not work in general and it has caused numerous programs a great deal of grief in approaching a COTS-based system. Keep in mind that some decisions are inseparable; for example, if two products need to work together, they may well have to be evaluated and chosen together. But usually, once certain fundamental architectural decisions have been made, it may be possible to focus on particular components that correspond to separable market niches and to expand the system one step at a time. That will increase your likelihood of success. A spiral approach will  help you with this.

### Factor 2: Do you have the ability to form partnerships with vendors?

This actually takes two forms. One is the ability to form such relationships as a matter of mindset. Vendors are not contractors. Do not expect to tell vendors what to do when. Each brings a wealth of expertise and perspective that is important to the success of your system, so you need to position yourself to take advantage of that. Do not try to force them to make changes that are unique to your system. Once you have chosen their product to include in your system, their success as a commercial enterprise is likely to be critical to your success as a system provider. Commercial enterprises succeed by doing a good job of satisfying a sufficiently large population. Learn to go with the flow and not to expect special favors or unique features that only you want. Learn to emphasize capabilities that have broader utility than just your system; this will give those capabilities commercial viability for the vendor. To do this, of course, you need to understand what other users want and need and join forces with them through such mechanisms as vendor user groups.

If the mindset is there, then the next form of the question addresses whether you have the resources to develop and nurture vendor relationships. You need people not only with the mindset but the skills. For example, you will need buyers and lawyers who understand how to negotiate license agreements that will ensure the success of both you and the vendor. You will also need to take the time and resources to talk with the vendor periodically, to share your needs, satisfactions, and dissatisfactions with their product and to understand their goals and the future of their product.

There are a lot of elements on which a positive vendor relationship can be based. What can you offer the vendor besides the money required for the number of licenses you need? Perhaps you have some expertise in the field that is unique in the community and that the vendor lacks; a good partnership would find a way to make that work to the advantage of the vendor—and yourself. Or perhaps you have the ability to do some special testing that a small vendor cannot handle: what accommodations might the vendor give to you in exchange for the test results on the product?

### Factor 3: Are industry standards available for elements of your system, and do you have the opportunity to use them?

Some market niches are very strong in their use of industry standards, other are not. A standards-based architecture can be a real advantage for a system when it comes to finding appropriate products and making them work together. It's not a guarantee—"plug 'n' play" has more appropriately been referred to as "plug 'n' pray—but if a market niche of interest to you uses standards, it's to your advantage to make use of them, too. This effort is just part of aligning your system and your organization with the market segments on which you will depend and with the business models that most affect you.

### Factor 4: Do all levels of your organization understand the wide-ranging implications of a COTS-based system approach, including the new skill sets required?

On the surface it would seem that changing to a COTS-based implementation approach is just a technical change. The reality is that it permeates *everything* you do, from your end users' processes, to your business strategies, to the very structure of your organization. An organization that is not prepared for this impact is likely to overlook some essential elements for COTS-based system success.

A COTS-based approach definitely requires different skills and thinking from the traditional approach. Although some people look to a COTS-based approach to relieve them of having to watch things, it seems that the exact opposite is true: you may have to be *smarter* as an acquirer of a COTS-based system than you ever had to be for custom systems.

But even if you're prepared with the right mindset, there are still new roles for your staff to fulfill and new skills that your staff are unlikely to have. For example, programmers are trained to take blank pieces of paper and create software designs and programs from scratch. It is a fundamentally different intellectual endeavor to take black boxes created by a variety of unknown people, discover the behaviors of each at the implementation level, and then integrate them to work seamlessly together. Such changes occur at *all* levels of personnel involved in your system—including managers as well as engineers, contractors as well as acquirers—and they must be identified with training and incentives provided to ensure the COTS-based system success.

### Factor 5: Do you have the flexibility in your system releases to accommodate the volatility of the marketplace?

COTS products are upgraded on the vendor's schedule, not yours. And each product's timing is independent of that of any other product. Say you have just 12 products in your system, each of which is upgraded on average once a year. That means that, on average, you will face the possibility of one upgrade to your system every month. But does that mean you have to do a release of your system every month? Not if you want to provide some level of stability for your users! And keep in mind that every release of your system requires updating of documentation and training, in addition to the testing that's necessary.

On the other hand, how often is often enough? If there are schedule restrictions that limit how often you can release a new version of the system, then you may find your ability to leverage the marketplace limited. For example, if there are constraints (whether real or by some executive's fiat) that do not allow you to upgrade your system more than once every three years, you are likely to fall behind the marketplace, or at least behind the vendors on which you are most dependent. In some cases, a whole generation of technology will occur before you can get on board with it. Very often, if you get too far behind a vendor's upgrades, you will find that your maintenance agreement is void and you have to start all over again, with a new (full-price) license just to catch up.

Every system must have the flexibility to balance the demands of keeping up with the marketplace and the need of end users for some level of stability.

# 4 Making a Business Case

The rules of thumb presented above should lead you to a preliminary decision based on solid experience. Making a business case requires commingling those rules of thumb and other described success factors with sound business judgment. The process for developing a COTS-based system business case is very similar to that used elsewhere in software engineering decision making. The difference in the business case for COTS-based systems is not in the process itself but rather in the factors necessary to properly evaluate a COTS-based approach. The COTS-based system shift away from the traditional process and the resultant learning curve must also be factored into the business case decision-making process. It could be that your decision based on the rules of thumb will not be substantiated in the rigors of making a business case.

The business case outlined in the following section may seem to repeat some of the material in the last two sections of this report. We felt this partial redundancy was preferable to forcing readers to assimilate all the information for themselves. The business case is a more detailed development of the thinking that is suggested by the rules of thumb.

## 4.1 The Business Case Process

The following items for the preparation of a business case for a COTS-based system are taken from a report by Oberndorf [Oberndorf 00]. They serve as a means to discuss the business case considerations that are particularly important, and in some cases, unique to COTS-based systems.

- **Determine critical success factors for the system**.

An essential part of any cost/benefit analysis is to articulate what “success” is and how it will be measured. What are the motivations and expectations for considering the use of COTS products? What does the system need to accomplish, within what parameters, that COTS products may be better able to do than a custom development could?

Some critical success factors for a COTS-based system may not be any different from considerations for other kinds of systems. These may include such goals as reducing overall operational support costs or personnel requirements or implications of system mission or vision.

Other considerations will be unique to a COTS-based system approach. These might include explicit statements of the degree to which the requirements are expected to be fulfilled (e.g., 80% may be adequate, or there may be specific high priority requirements that must be fulfilled while other less critical needs may be negotiated) or an ability to perform a technology refresh cycle every three years.

- **Conduct a preliminary study of the feasibility of a solution using COTS products.**

The intent of the preliminary study is to start to formulate acceptable alternatives using COTS products. It is necessary to determine whether the marketplace can help in the development of a solution for the system problem. It might even be possible at this stage to reduce the field of candidates or options, based on what is (or is not) available in the marketplace, to a number that can be reasonably analyzed.

- **Identify key COTS-based system assumptions.**

It is probably the case that your analysis will be based in part on some key assumptions. These will include some that are peculiar to your use of COTS products. Examples of these are which technology will win dominance of a particular market niche, when a particular technology will be overtaken by a newer one, or how long a vendor will maintain a commitment to a given product. These will greatly affect your analysis, and you will need to be sure you understand how sensitive your ultimate results are to these assumptions.

- **Formulate COTS-based system strategic plans.**

The strategic plans are as important as the alternatives for ultimately deciding which solution is to be preferred. These strategic plans should be applicable regardless of the solution being considered; the alternative solutions are more tactical, endeavoring to find the best way to implement the strategic plans. The strategic plans will cover such things as the bounds of the system problem to be solved and therefore the scope of the solutions, the marketing potential of the resulting system, the range of alternative approaches to be considered, funding sources, and the stability of funding. Different alternative solutions are likely to respond to these strategic plans differently, thus contributing to the analysis.

- **Articulate the alternatives to be analyzed.**

These ideas started to take shape conducting the preliminary feasibility study (see above). It should be noted that the alternatives should not be limited to a single COTS-based one. It is most likely that a comprehensive business case will investigate alternatives for at least one custom solution as well as *more* than one COTS-based solution. That is the most thorough way to evaluate the feasibility of using a COTS-based solution—and, if a COTS solution is to be the answer, which one to choose.

- **Analyze the financial implications.**

The analysis must consider the likely costs and benefits—at a gross level—that a COTS-based solution will experience over its lifetime and compare that with the same for a comparable custom solution or other COTS-based solution. It is critical to take a system lifetime perspective: many people look only at the comparative development costs and see a COTS solution as a clear winner. These estimates must account for the costs associated with events such as repeated product upgrades and technology turnover, which generate a need for constant testing and continuous monitoring of the marketplace for the entire lifetime of a COTS-based system.

Aspects to be considered are such things as

- the projected return on investment (ROI) over the system lifetime
- total cost of ownership
- cost factors peculiar to the solution
- the cost of risk mitigation
- backup plans
- the cost of contractor, vendor, or government incentives
- ramp-up costs

There are many new or changed cost factors to be accounted for specifically when dealing with a COTS-based system. The lists in Table 1 are not necessarily comprehensive, but they do bring together many of the more important aspects that must be analyzed for a successful result.

- **Analyze alternatives and make recommendation(s).**

All of the alternatives must be analyzed in their entirety. This involves determining pros and cons, especially with respect to the success factors, assumptions, and strategic plans articulated earlier. The recommendations that are made to the decision makers must bring forward the alternative solutions that best satisfy the collection of considerations.

- **Revisit the COTS business case.**

A business case—and especially a COTS business case, if a COTS solution is the chosen alternative—is not a static thing to be developed once and put on a shelf. Circumstances change even with a custom solution; they most certainly change frequently and often unexpectedly with a COTS-based solution. The business case you have developed should be revisited periodically, e.g., once a year, and also at key reassessment events, such as a significant change in requirements or a sudden shift in the marketplace. Each project must collect the cost and resource data and analysis rationale associated with the business case on an ongoing basis and use that when it is time to revisit the business case.

| New Program-Related Cost Factors | New Market-Related Cost Factors | New Product-Related Cost Factors |
|---|---|---|
| the cost of migration to a CBS approach (at either or both the system and components levels) | vendor demise | licensing options and license management |
| testbeds | cascading upgrades | frequent product upgrades |
| cultural change and training | end of technology life | product feature reduction or bloat |
| information collection and dissemination | technology and market watch | changes to license arrangements |
| guidance, examples, handbooks | evaluation | product replacement |
| incentives | market research | dropped support for a product |
| iterative development | technology forecasting | COTS product sustainment |
| engineering an evolvable architecture | responding to marketplace volatility | warranties and data rights |
| reacting to marketplace changes | | COTS product end-of-life events (e.g., product being dropped by its vendor or its vendor going out of business) |
| (re)integration | | |
| technology refresh (including those associated with non-developmental items) | | |

*Table 1: New Cost Factors for COTS-Based Systems*

## 4.2 Using the Business Case

It is amazing how often business cases are generated, only to be ignored, with decisions based instead on factors or circumstances that are either not supported by the business case or perhaps not even represented in it. It is imperative that you use the results of the original business case and subsequent revisits for decision making.

This is not necessarily easy. There is always that upper level executive who makes unwarranted determinations or who doesn’t want to take the time or resources to make decisions based on the facts and the best knowledge of his experts. There are inevitably political pressures that will be brought to bear on such decisions. But ultimately it is the staff of the project —and the system’s users—who will have to live with the results. Shortsightedness never wins over care, especially when it comes to the marketplace.

# Annex: The Influence of the DoD Acquisition System

The Department of Defense process for acquiring systems is very complex, with innumerable checkpoints to ensure accountability for the expenditure of funds. The sequential, lock-step nature of these decision support systems is at odds with many of the practices suited to the acquisition of COTS-based systems. As of this writing, the Defense Acquisition System is defined by DoD Directive 5000.1, which "[p]rovides policies and principles for all DoD acquisition programs." [DoD5000.1]. While the Clinger-Cohen Act [Clinger-Cohen] acknowledges the importance of commercial technology, it does not countermand all of the existing procedures by which DoD COTS-based software-intensive systems are acquired. The Defense Acquisition System creates blatant conflicts with a COTS-based system implementation.

One example of this disconnect is the dissimilar speeds at which different aspects proceed. While it can take 3 to 6 years to receive funds for a new requirement, the typical life cycle for commercial software products is 6-18 months. Or contrast the typical 12 months spent in briefing coordination prior to a major milestone review with the 6-month spiral development iteration that is often very conducive to development and sustainment of COTS-based systems. Another way in which funding cycles can be a problem is that vendors may unexpectedly offer a discount on products you need, often because they need to enhance their bottom line at the end of a quarter or fiscal year. But slow-moving funding cycles can mean that you do not have the funding resources necessary to take advantage of the offer. In today's climate, creative approaches may be required.

Another example of the disconnect is the degree of certainty or confidence expected at each stage in the DoD acquisition process versus the high degree of uncertainty with which each iteration of a spiral for a COTS-based system development begins. The DoD's Acquisition Management System affords the program manager greater latitude in defining specific goals at the beginning of each development phase than does its Requirements Generation System. This latitude and the accompanying requirements flexibility are essential for success with COTS-based systems. On the other hand, the Planning, Program, and Budget System forces the program manager to specify, several years in advance, specific capabilities to be acquired or pre-planned product enhancements to be fielded, demanding an uncanny ability to predict where the marketplace is going to be 2, 5, or even 10 years hence.

# References/Bibliography

**[Brownsword 99]** Brownsword, L.; Oberndorf, P.; & Sledge, C. *COTS-Based Systems for Program Managers* (tutorial). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1999. Available on CD-ROM.

**[Carney 98]** Carney, David J. *Quotations from Chairman David*. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1998. <http://www.sei.cmu.edu/publications/documents/99.reports/lrb/little-red-book.html>.

**[Clinger-Cohen 96]** Clinger, W. & Cohen, W. Public Law 104-106 National Defense Authorization Act for Fiscal Year 1996, February 10, 1996.

**[DoD5000.1]** U.S. Department of Defense. *The Defense Acquisition System* (DoD Directive 5000.1), 2000. <http://www.dtic.mil/whs/directives/corres/html/50001.htm> (active as of December 2002).

**[DoD5000.2]** U.S. Department of Defense. *Operation of the Defense Acquisition System* (DoD Instruction 5000.2), 2001. <http://www.dtic.mil/whs/directives/corres/html/50002.htm> (active as of December 2002).

**[Oberndorf 00]** Oberndorf, Patricia; Brownsword, Lisa; & Sledge, Carol. *An Activity Framework for COTS-Based Systems* (CMU/SEI-2000-TR-010 ADA383836) Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2000. <http://www.sei.cmu.edu/publications/documents/00.reports/00tr010.html>.

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*

1. AGENCY USE ONLY
(Leave Blank)

2. REPORT DATE
December 2002

3. REPORT TYPE AND DATES COVERED
Final

4. TITLE AND SUBTITLE
Rules of Thumb for the Use of COTS Products

5. FUNDING NUMBERS
F19628-00-C-0003

6. AUTHOR(S)
Michele Motsko, Patricia Oberndorf, Ellen-Jane Pairo, James Smith

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

8. PERFORMING ORGANIZATION REPORT NUMBER
CMU/SEI-2002-TR-032

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
HQ ESC/XPK
5 Eglin Street
Hanscom AFB, MA 01731-2116

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
ESC-TR-2002-032

11. SUPPLEMENTARY NOTES

12A DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified/Unlimited, DTIC, NTIS

12B DISTRIBUTION CODE

13. ABSTRACT (MAXIMUM 200 WORDS)

More and more organizations are realizing the benefits—and sometimes the necessity—of incorporating commercial off-the-shelf (COTS) products in the systems they acquire and use. But COTS products are not necessarily the right solution for every system. When is it wise topursue a COTS-based systems approach, and when is it best to hold back? How can sound COTS-based-system practices be reconciled with an organization’s regulatory and policy constraints? This report provides some information to help guide these decisions.

14. SUBJECT TERMS
commercial off-the-shelf, business case, success factors

15. NUMBER OF PAGES
33

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT
UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. Z39-18 298-102