



CarnegieMellon
Software Engineering Institute

Pittsburgh, PA 15213-3890

**SEI Monographs on the Use of
Commercial Software in Government
Systems**

Case Study: Significant Schedule Delays in a Complex NDI-Based System

David Carney

June 1998

About this Series

Government policies on the acquisition of software-intensive systems have recently undergone a significant shift in emphasis toward the use of existing commercial products. Some Requests for Proposals (RFPs) now include a mandate concerning the amount of COTS (commercial off-the-shelf) products that must be included. This interest in COTS products is based on a number of factors, not least of which is the spiraling cost of software. Given the current state of shrinking budgets and growing need, it is obvious that appropriate use of commercially available products is one of the remedies that might enable the government to acquire needed capabilities in a cost-effective manner. In systems where the use of existing commercial components is both possible and feasible, it is no longer acceptable for the government to specify, build, and maintain a large array of comparable proprietary products.

However, like any solution to any problem, there are drawbacks and benefits: significant tradeoffs exist when embracing a commercial basis for the government's software systems. Thus, the policies that favor COTS use must be implemented with an understanding of the complex set of impacts that stem from use of commercial products. Those implementing COTS products must also recognize the associated issues—system distribution, interface standards, legacy system reengineering, and so forth—with which a COTS-based approach must be integrated and balanced.

In response to this need, a set of monographs is being prepared that addresses the use of COTS software in government systems. Each monograph will focus on a particular topic, for example: the types of systems that will most benefit from a COTS approach; guidelines about the hard tradeoffs made when incorporating COTS products into systems; recommended processes and procedures for integrating multiple commercial products; upgrade strategies for multiple vendors' systems; recommendations about when not to use a commercial approach. Since these issues have an impact on a broad community in DoD and other government agencies, and range from high-level policy questions to detailed technical questions, we have chosen this modular approach; an individual monograph can be brief and focused, yet still provide sufficient detail to be valuable.

About this Monograph

This monograph describes a program that is currently building a large system for the DoD. This system makes extensive use of pre-existing components: re-engineered legacy systems, components that are government-furnished equipment (GFE), and both government- and commercial-off-the-shelf (GOTS and COTS) software. Thus, although the system is not precisely a "COTS-based system," the issues faced in building this system are parallel enough to those in COTS-based systems to make it a useful case study.

The expected audience for this monograph is a general audience, and the major issues tend to be more programmatic and managerial rather than purely technical. The goal of the monograph is to focus on the complexities (which do include both technical and programmatic issues) that can hinder or dilute the expected benefits of using commercial components in complex government systems.

Case Study: Significant Schedule Delays in a Complex NDI-Based System

1 Introduction

The focus of this monograph series is on issues specific to commercial software. In this monograph, however, we make a slight change in focus. We describe a program tasked with building a major air defense system for the DoD that makes extensive use of pre-existing components: re-engineered legacy systems, components that are government-furnished equipment (GFE), and both government- and commercial-off-the shelf (GOTS and COTS) software. The commonly-used term for this diversion of components is “non-developmental items” (NDI).¹

While the system examined in this monograph is not precisely the type addressed by the other papers in this series, the issues faced in building this system are parallel enough to those in COTS-based systems to make it a useful case study. In particular, the aim of this monograph is to examine, through this program, the types of factors that can influence acquisitions that use heterogeneous pre-existing components, whether COTS or otherwise.

The project in question suffered a large schedule slip early in its lifetime. One of the goals of this study, therefore, is to examine those factors related to use of NDI that may have been contributors. However, the use of preexisting components is only one of several contributing factors. The project was also characterized by other features that reflect the government’s ongoing efforts to modernize its acquisition strategy through new and innovative procurement methods. Some of these features, as described and analyzed herein, figured heavily in the schedule slip. Others were less significant and are noted merely for background information. Common to all of these innovative features, however, is their newness to government and contractor personnel alike. This newness is significant, because while many such innovative concepts are demonstrably beneficial toward improving the acquisition process, their inverse aspect—lack of familiarity—was also a major cause of the project’s schedule delay. A prime focus of this monograph is on the novel factors, use of NDI being one of them, that characterize this project.

The author of this monograph was part of a team that interviewed members of this project in preparation for this case study, and at the request of the project’s Executive Officer. Where it is necessary to refer to the project by name, “Project X” will be used. Note, however, that a reference to “Project X” is typically to the entire project: contractor, subcontractors, and government personnel alike.

¹ The Federal Acquisition Regulations (FAR) make a much more precise distinction between “COTS” and “NDI” than we intend by this statement.

The remainder of this document is as follows. In Section 2, the project is briefly described. Section 3 presents the current status of the project.² Section 4 is an analysis of the project. Section 5 contains a summary of lessons and recommendations.

2 General Description of the Project

This section will describe project X from four vantage points: the expected functionality of the system, the make-up of the various organizations involved in the project, the project's use of innovative processes and methods, and the sources of the pre-existing components that are expected to be used in the system.

2.1 Description of the System

The system is a complex command-and-control system that performs several real-time air defense functions. It replaces and modernizes an existing fielded system. The high-level capabilities include surveillance, tracking, data communications, and weapons control; a set of applications will make use of these high-level capabilities to provide end-user functionality. There are also data link interfaces to numerous external hardware and software systems.

Project X improves on the existing system in several ways.

- The replacement system will duplicate the capability of the existing system, but will also include additional functionality.
- The replacement system is intended to be open and extensible.
- The replacement system will be written in an object-oriented language, replacing the obsolete language of the existing system.

Expansion of system capability is one obvious benefit of this project. Another key benefit is the expected reduction in maintenance costs currently incurred due to the obsolete programming language. This project is also consistent with the general desire throughout DoD to use modern programming technology and to speed up the acquisition process.

The system will be fielded first with an initial capability in the 1998-99 timeframe (eighteen months after project start), and the full capability will be fielded in 2000-01. The initial capability will be introduced through four builds, and there will be three more builds for the full capability.

2.2 Organizational Make-Up of the Project

Project X has a single prime contractor, with several subcontractors located throughout North America. The government management of the program is shared by two distinct military organizations. In addition, a large number of personnel from a major government support contractor actively contribute to the management of the project.

The project makes extensive use of integrated process teams (IPTs). Each IPT has a team leader drawn from the contractor (or one of the subcontractors) and also has a government focal point.

² By "current status," we mean the current status as of the author's interviews with the project staff; these took place toward the end of 1997.

Most of the IPTs are technical, with an overall IPT for project management. The technical IPTs are partitioned primarily according to the major functional areas of the system (e.g., surveillance, data communication). In addition, one IPT focuses on the end-user applications, and one focuses on the new functional capability of the replacement system.

2.3 Innovative Processes and Methods in the Project

Both by the contractor's initial proposal and by subsequent government direction, the project is making use of several innovative elements; three are especially important. First, the contractor and government are jointly participating in IPTs, as described above. There is an express intention for the government to avoid a "heavy-handed" approach in this project, and the IPTs are a vehicle to achieve this.

Second, the contractor is expressly making use of a spiral development process, as opposed to the traditional "waterfall" model. The expectation is that by streamlining the acquisition process (e.g., through reduction of documentation, using a "requirements tradespace"), costs will be reduced and time of development reduced.

Third, and perhaps most significant, a very large portion of the system is expected to be drawn from other sources: these pre-existing components are intended to provide a major amount of functionality to the fielded system.

2.4 Sources of Pre-Existing Components

There are several sources of these pre-existing components. The most critical one is an internal research and development effort by the contractor. This effort is producing a set of components that are expected to provide much of the end-user application software. They comprise roughly 88K lines of code, which will be about 35% of the overall system.

Software provided as "government-furnished equipment" (GFE) is another major source of pre-existing components. One large item of GFE for project X comes from a separate defense project, which is currently working on a different system, but one whose functionality contains much of the enhanced functionality needed in project X. By government direction, this other project is providing its output for use by the project X contractor.

In addition, project X will make use of the DoD's Defense Information Infrastructure/Common Operation Environment (DII/COE), which is provided as GFE by the Defense Information Systems Agency (DISA). DII/COE is also a major element in the system being built by the other defense system.

Yet another potential source of pre-existing components includes both government off-the-shelf (GOTS) software and various commercial off-the-shelf (COTS) components, several of which are being considered for possible inclusion in the system. These include infrastructure components (e.g., system administration software, network middleware) and a large number of components for data communications. A simple depiction of the system and its incorporation of pre-existing components is shown in Figure 1.

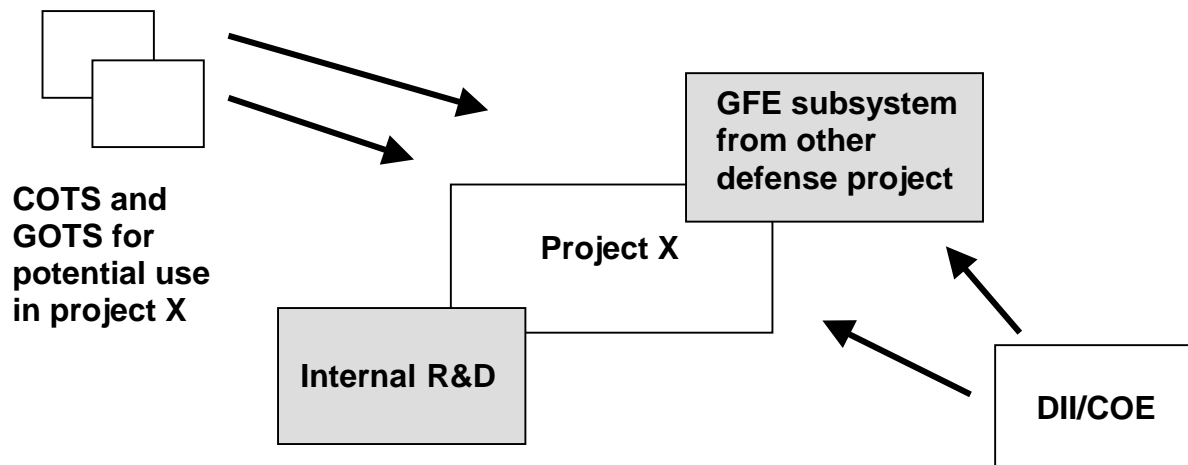


Figure 1: Incorporation of Pre-Existing Components

3 Current Status of the Project

Based on a series of interviews with both contractor and government project members, the current status of the project can be described in five key areas:

- the contractor’s internal R&D software program
- the other GFE software (both from the existing DoD project and DII/COE)
- the contractor’s software development and integration processes
- the structure and workings of the IPTs
- risks to the current project schedule

The first four of these items are described in Sections 3.1 through 3.4. Note that these descriptions are intended to be primarily factual. Afterwards, in Section 4, we present an assessment of this information, together with analysis of other significant information and the assessment of risks to the current schedule.

3.1 Contractor’s Internal Research and Development Program

The internal R&D program has been in existence for approximately two years. The program is essentially a reengineering one, since the goal is to abstract functionality from an existing air defense system; there is no significant addition of functionality. This R&D project is organizationally separate from project X, and is managed by a different manager, though both of these managers report to the same vice-president.

The contractors’ original plan for this effort was to create a functional kernel useful for multiple air defense applications, and then to market this commercially to several potential users. To that end, the code is being translated (to Ada95), and an extensive repackaging is being carried out to bring the code into conformance with object-oriented principles and also into conformance with DII/COE. In the original plan, use of an automated translation tool was expected to facilitate the translation.

Several things have modified the original R&D plan. First, the contractor judged that the translation tool did not provide acceptable results; the automatic translation has therefore been abandoned and the code is being reengineered by hand. Second, the expected customer base has not materialized, and project X is currently the only major consumer of this software. Third, the R&D effort fell behind its original schedule (i.e., the schedule that was assumed when project X was begun). This was partially due to the loss of automatic translation, but is at least equally due to staffing shortfalls.

There are two major results of these modified circumstances. First, the reengineering effort is now refocused at producing what is essentially a library of reusable components. These still provide a generalized set of air defense capabilities, most of which are useful to project X. However, these capabilities are not integrated in any substantial way, but will exist as a set of loosely coupled Ada packages. Second, the R&D schedule has been brought into closer harmony with project X. Scheduled builds of the R&D system now focus their functional makeup on the needs of project X, and those functions that are not needed (by project X) have been deferred to the last part of the revised R&D schedule. In addition, the team performing the R&D project have agreed to assist the project X team in their integration activities.

3.2 Other GFE Software

Another DoD project is expected to contribute its output as GFE to project X. This will be used in two ways. First, it will provide the enhanced functionality for project X's system, and second, it will provide several infrastructure capabilities. This other system is being implemented by a different large defense contractor, which is acting in this instance as a subcontractor to project X.

At the time of the interviews for this study, there had been little cooperation between these two programs. Expected delivery of components to project X was delayed by as much as six months, and there was no substantive co-location of personnel between the two. Some of the needed components (i.e., those that are part of project X's infrastructure) have dependencies and schedule impacts on the rest of the project. Other components (i.e., those that provide the enhanced functionality) are less critical, and are architecturally more modular; they can be added to project X whenever they are received. To date there has been apparently little investigation by the contractor about the precise technical approach for integration of these components (which are generally "black boxes") into the system.

In addition, the DoD's DII/COE is a component of project X, and is also a component of the other defense project. Both projects are therefore dependent on the ongoing DISA schedule for DII/COE releases. At the time of the interviews, project X had encountered severe delays in receiving all of the current DII/COE components from DISA. (By contrast, the other project, which also relies on DII/COE, had apparently gained a "fast-track" path to receive the DII/COE components in a more timely fashion.)

3.3 Software Development and Integration Process

In addition to the extensive reuse of NDI, the development approach of project X makes use of other innovative development processes. One of these is the intentional minimization of many artifacts of an "old" acquisition process. Excess documentation is to be avoided, (e.g., the written

proposals were asked to be limited), and where practical, use of oral reports and demonstrations is preferred.

Project X also makes use of the “spiral” development approach, as opposed to such traditional approaches as “waterfall.” Since there is often some confusion about what the “spiral” model really is, it is useful to cite Barry Boehm, who did much of the early work on this model:

Each cycle of the spiral [identifies] the alternative means of implementation...and the constraints imposed...The next step is to evaluate the alternatives...This may involve prototyping, simulation...or combinations of these...Once the risks are evaluated,...a plan for the next level of prototyping [is made]...and the development of a more detailed prototype. [Boehm 88]

The project has apparently been successful in minimizing excess documentation; the interview team saw few indications that the contractor was being delayed by a requirement to produce large (and implicitly unnecessary) documents.

However, in its use of a “spiral” development process, we observed some inconsistencies between intention and reality. In particular, there was little evidence of a willingness to leave some details undefined until later in the development process. Instead, an extensive requirements negotiation activity, one consuming much more time than planned (and one that would be typical in a waterfall process), had been conducted, and there was a perceived need on both sides to finalize all of the specific details, by defining all of the requirements in advance of any work on actually building the system. As of the interviews, most of the requirements were considered to be fully defined. (There was, however, some inconsistency in determining precisely how many requirements were still outstanding.)

3.4 Structure and Workings of the IPTs

There are eight integrated product teams in project X:

- program management
- system engineering
- installation and support
- surveillance
- data communications
- services and integration (of the enhanced functionality)
- displays
- applications

The number of personnel varies from IPT to IPT, but most of the teams have about two dozen members. The IPTs are each clearly divided into two subsets, which are quite distinct: one is government, the other contractor, and they reside in widely separated places. The government subteam is further divided into personnel from the different military organizations that are sponsoring the project.

There are periodic meetings of the teams and the project as a whole. While these meetings avoid such nomenclature as “critical design review,” they are nonetheless the occasions when the contractor describes the current status of various project and technical issues to the government members, and when the government makes requests of the contractor that certain steps be taken or priorities reordered.

4 Analysis: An Assessment of the Project

The individual assessments of the four key areas described above are presented below. However, many of these findings are interrelated, and we will therefore provide an additional section that synthesizes these findings in terms of the overall program.

4.1 Contractor’s Internal Research and Development Program

We concluded that the R&D project is now proceeding reasonably well. While there had undoubtedly been problems in the past, the program appears to have improved its management, and appears to be meeting its current schedule. Interviews with the key software designers on the project indicated that the technical capability to produce the intended product is in place.

Two areas of concern still exist. Staffing is a genuine problem, and though the contractor described a plan to hire more staff, recruitment of adequately experienced personnel will very likely be difficult. Another remaining issue is that in recasting the R&D effort from producing a unified subsystem to producing a set of loosely coupled packages, the difficulty and risk of integrating these packages has now been transferred to project X. One mitigation for this risk is that the contractor plans to have the R&D personnel assist with the integration effort, and presuming that this occurs, this risk should be minimized considerably.

4.2 Other GFE Software

The dependence on GFE from the other defense program (see Section 3.2) has two potentially adverse impacts on project X. The first is on the schedule: since the other program’s schedule is not fully conformant with that of project X, its own project requirements will clearly take precedence over those of project X. This was evident in the six-month delay that had already occurred at the time of the interviews. This dependence has a critical and a non-critical side. In the case of the enhanced functionality that it provides to project X, the dependence is non-critical, since the components are expected to be relatively modular, and are not critical pieces on which others will depend. However, there is also some critical infrastructure capability, and for this, any schedule delay from the other contractor will almost certainly result in a schedule slip.

A second issue concerns the presence of DII/COE in both systems, and the differences in delivery paths for both. (As noted above in Section 3.2, project X has had extreme difficulty in getting full delivery of DII/COE, while the other contractor has a “fast track” path for delivery.) From one perspective, it might be thought that the existence of two separate paths for DII/COE—one directly from DISA, the other as a subset of the components from the other contractor—suggests a benefit for project X (i.e., if the components are delayed through one path, they might be more quickly received through the other). However, it was indicated that there would be at least some “alteration” in the DII/COE components by the other contractor (but with no specificity of the extent of this “alteration”). This implies that the two separate paths might also imply two separate

sets of components, with the attendant issues of versioning, configuration management, and so forth, all of which are potential sources of delay, and worse, potential sources of system defects.

4.3 Software Development and Integration Process

The “spiral” model is not truly being used on this project. The most obvious manifestation of this is in the area of requirements. As described previously, if the project were truly following a spiral approach, at least one iteration of building the system would probably precede any attempt to finalize the requirements. Instead, project X appears to be relying on a traditional approach that demands full completion of requirements definition before implementation begins. This assessment is further corroborated by the overall project plan. There are now seven scheduled builds of the system. However, no applications software is scheduled to be included until the fourth build, and the major portion of the application capability is deferred until the final build. In a true spiral approach, it would be advantageous to incorporate end-user functionality as early as possible.

There is another aspect of project X’s development process that was of concern. Whether it uses spiral, waterfall, or any other approach, any project that relies on pre-existing components must make allowances for the constraints they impose. For instance, project X is expecting to use CORBA,³ a complex COTS product, as part of the system infrastructure. It is therefore reasonable that expertise in using this mechanism be available. The project plans, however, do not appear to factor in such elements as purchasing commercial CORBA implementations, training, or experimentation. We felt that this inexperience has the potential to result in additional schedule delays.

4.4 Structure and Workings of the IPTs

The IPTs are not working well, and are the cause of a serious deficiency in the program. The IPTs are not truly integrated teams at all. That there is a “government-only” side and a “contractor-only” side is evidence of this: discussions with individuals from both sides sometimes produced markedly different views of the current status. There are other indicators of this separation: as noted above, there are still traditional project reviews (however they are named) in which the contractor describes to the government the current status of various technical areas. But if the teams really were integrated, such reviews would be moot, since all members of the team (i.e., the government members included) would be conversant in the ongoing work of the team.

This results at least partially from the lack of any real degree of co-location. This is not merely between the government and the contractor, but between contractor, several sub-contractors (particularly including the other large defense contractor), and the two different military sponsoring agencies. This wide distribution of personnel among numerous sites causes a serious lack of coherence. While there are laudable attempts to overcome this drawback (e.g., many individuals spend one- or two-week sessions at other sites), this is insufficient, and we concluded that this issue is a major cause of the current schedule slip.

³ The Common Object Request Broker Architecture, which is most simply (but also simplistically) described as an object-oriented messaging capability.

One glaring corroboration of the potential negative impact of distributed development emerged accidentally during a discussion with the contractor about the internal R&D project. The project X personnel described their difficulty in getting necessary technical information with the R&D personnel because they were in a different building on the contractor's facility. But now that the R&D personnel had been relocated to project X's building —*a move fifty yards from one building to another*— the project X team was able to gain much more detailed technical information when it was needed. Given this (not unexpected) fact, it is hardly surprising that an aggregate distance of several thousand miles should prove to be at least as great a barrier to effective technical interchange between contractor, subcontractor, and government personnel.

4.5 Synthesis: How These Factors Interact

The previous sections have described individually the findings and assessments in four key areas; some of these factors clearly contributed to project X's schedule slip. However, the interview team also found that these factors were not independent, and that there was a deeper (and perhaps more important) cause of delay: many new and unfamiliar methods, procedures, and approaches were being introduced simultaneously in this program, and these all interacted. This produced unfortunate results in both the management and the technical areas

Management

There is a lack of familiarity on all sides both about IPTs and about a spiral development approach. Thus, on one hand, the contractor, aiming to appease the government's interest in innovation, has defined a set of nominally independent teams. But there is, apparently, no clear understanding of what this approach implies. For instance, if we replace the traditional project hierarchy by integrated teams, who then takes on the role of chief architect? Who is the person really in charge? Who knows the entire system? There was no one that we could find that answered this description. On the government side, the government participants in project X, are very mindful of the charge to "avoid heavy-handedness," and are therefore keeping hands off—but in the wrong way, by not really being truly integrated into the IPTs, and by not participating in critical decisions that need to be made on a daily basis. (It is ironic that in the one area where there really does need a "hands-off" approach from government, namely, letting requirements be loose and undefined until late in the program, there is far too much hands-on.) The result of all this is that, to all appearances, both sides had a remarkable posture of inactivity: the contractor was waiting for direction from the government, and the government was (mostly) keeping their "hands off."

The use of pre-existing components has a different impact on the program management style. Cost and schedule are the general drivers for pursuing this approach, but there are drawbacks in each case that offset the potential benefits, and that must be factored into project planning. Thus, saving time in development by using NDI (e.g., from the other defense contractor) is offset by the fact the other contractor's delays become your delays. As another example, the savings that result from buying rather than building a complex component (e.g., CORBA) may be offset by the potentially large cost of gaining sufficient expertise in using that component.

Technical

Most of the technical problems faced by the project X contractor also result from an unforeseen aspect of using NDI, COTS, and pre-existing components in general. If such an approach is used

on a project, there are specific technical implications that one must face. For instance, one might avoid the time and expense of building a complex component (i.e., by getting it as NDI), but then one must have the technical expertise to integrate a component whose internal workings are unknown, whose design assumptions are possibly at odds with the current system, and that may well exhibit undocumented behaviors. Thus, the need for expertise has not been removed, but is simply transferred to a different skill domain. In fact, there are many entirely new skills implied by using pre-existing components: evaluating components based on only partial knowledge; creating adequate testing procedures for “black-box” components; debugging systems with components whose source code is unavailable.

A different issue is relevant for the contractor and the government side. There is a great danger that unfamiliarity with the benefits and risks of an NDI approach will be translated into unrealistic expectations. It is simply wishful thinking to believe that a complex system like project X can be essentially and easily constructed from a set of heterogeneous pieces that were all created independently. To assume this is overly simplistic, and denies the widespread experience in the engineering community.

4.6 Risks to the Current Project Schedule

At the time of the interviews, a revised project schedule had been developed by the contractor that made a seven-month adjustment to the original schedule. We examined this schedule and concluded that there was a strong potential that this revised schedule would slip further. Some of the risks that could lead to this were essentially programmatic: lack of experiences with IPTs and the spiral approach, and insufficient engagement of the government personnel. Others were essentially technical: the difficulty of integrating heterogeneous components; introduction of new technology (e.g., CORBA); the applicability and suitability of the pre-existing components to the functional requirements of the system.

The final conclusion was that the revised schedule had a large element of risk.

5 Summary: Lessons and Recommendations

As a corollary to this case study, we made some high-level recommendations to mitigate these risks to the schedule. The major recommendations were programmatic: there appeared to be a need to establish urgency at every level throughout the program, and there was a compelling need by both government and contractor to clarify responsibility and authority for decisions.

But aside from mitigations of individual problems and risks, the interview team found that most of the problems uncovered during the interviews were essentially symptoms. The underlying issue is that many of these innovative methods and styles, especially when used simultaneously, can bring about competing goals and constraints, and these must be faced and prioritized. If the schedule has priority (as is evidenced by the aggressive, 18-month schedule for project X to arrive at initial capability), then there must be a willingness to relax some requirements, to evolve to full requirements, and someone must make a command decision that staff will be transferred or relocated as necessary.

If performance of the system has priority (as evidenced by the extensive requirements activity), then it would be entirely reasonable to initially establish the full understanding of the system. In

fact, in such a case then it would probably much more reasonable to use a waterfall, rather than a spiral development process.

And finally, if cost has priority (as evidenced by the desire is to use as much pre-existing material as possible), then there must be a willingness to accept the implications of this approach on the project schedule, on system performance, on the contractor's management style, and on the government's participation in the project.

References

[Boehm 88] Boehm, Barry W. "A Spiral Model of Software Development and Enhancement." *IEEE Computer*, May 1988.

Feedback

Comments or suggestions about these monographs are welcome. We want this series to be responsive to the real needs of government personnel. To that end, comments concerning inclusion of other topics, the focus of the papers, or any other issues are of great value in continuing this series of monographs. Comments should be sent to:

Editor
SEI Monographs on COTS
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213
cots@sei.cmu.edu