Building Security into the Business Acquisition Process

Dan Shoemaker

June 2007

ABSTRACT: This article presents the standard process for acquiring software products and services in business. It is based on the recommendations of the Agreement processes specified by the IEEE 12207 Standard. This standard presents the commonly accepted practices for ensuring a well-defined and persistent assurance process for acquired software. With the help of 12207, it is possible to integrate best practice in acquisition and supply into a single uniform approach. That approach will guarantee that security considerations will be a central part of product selection, monitoring, and acceptance. The ensuing set of policies and procedures provides rational control over all aspects of the process of securing acquired products. Properly followed, they will ensure an adequately secure software deliverable.

THE IMPORTANCE OF A STANDARD MODEL

It is almost impossible to assure the reliability of any complex thing without following a well-defined and rigorous process. And, since our national economy and defense depend on the reliable execution of software, the most important process in this day and age might very well be the one that helps us say with certainty that all of our software is free from exploitable defects.

That necessity has been stressed in every Presidential Homeland Security Directive from HSPD-1 (October 2001) to HSPD-7 (December 2003), and it is embodied in the National Strategy to Secure Cyberspace [NIAC 2004]. The increasing trend toward building systems out of purchased parts just serves to enhance the particular importance of getting the acquisition of software right [Cisco 2003].

It goes without saying that software is hard to buy. That is because the buyer is essentially purchasing a complex and invisible set of underlying functions without any direct control over how they are built. Therefore, the best practical way to ensure that a quality like security is built into the final deliverable is to use procedures that are specifically aimed at increasing the acquirer's understanding and control over the process. Those procedures must guarantee that security considerations are a central part of the routine vendor selection, monitoring, and

Software Engineering Institute
Carnegie Mellon University
4500 Fifth Avenue
Pittsburgh, PA 15213-2612

Phone: 412-268-5800
Toll-free: 1-888-201-4479

www.sei.cmu.edu

acceptance process, and in order to ensure their reliability they should be well-defined and persistent within the enterprise's organizational architecture.

John Steven, in an article on BSI, uses the term Enterprise Software Security Framework (ESSF) to describe this concept. The purpose of the ESSF is to organize everybody's responsibility for achieving secure software into a "who, what, when" structure of defined roles and interrelationships. To create this structure Steven suggests that the ESSF must include roles and responsibilities beyond the security analyst and software application development teams.

Extending security responsibility into the organization itself seems like prima facie good practice, since common sense alone would suggest that software security involves more than just the technical aspects of the product. On the other hand, the greater the number of participants, the harder it is to ensure that the process will be well coordinated. Therefore, a common framework to synchronize the potential roles and activities would be helpful. Fortunately an internationally recognized standard process for doing this does exist in the form of the Agreement processes, which are specified as part of the IEEE 12207.0 Standard, entitled Industry Implementation of International Standard ISO/IEC 12207:1995, Standard for Information Technology, Software Life Cycle Processes [IEEE 1998].

Background

Until 1995, there was not a single complete standard that itemized all potential forms of activity within the software lifecycle. ISO 12207 [ISO 1995] and its American cousin IEEE 12207.0 [IEEE 1998] addressed that need. The 12207 framework describes a complete set of practices for software, which range all the way from conceptualization through retirement. Within this life cycle, the Agreement processes (which specify the expected actions of the "Customer" in Acquisition 5.1 and those of the "Supplier" in Supply 5.2) specify all of the activities needed to acquire software products and services. In essence, 12207 fuses best industry practice into a single approach that underwrites effectiveness in software acquisition work.

With the help of 12207 it is now possible to integrate best practice into a single uniform process. The ensuing set of procedures provides rational control over all aspects of the software life cycle. By standard, the processes itemized in 12207 constitute a complete set, which applies to all development, operation, and maintenance work.

Specific tasks itemized in the standard apply to all types of software activity. Tailoring of those tasks is done by identifying the unique project issues, problems, and criteria and documenting the adjustments from the standard definitions. Subsequent tailoring is done by decomposing process components to their logical level of refinement and repeating that step for every other required process.

A STANDARD ACQUISITION PROCESS

In simple terms, the 12207 Acquisition process describes the customer role in the purchasing process. The activities specified by 12207 for acquisition are intended to convey the typical actions that should be undertaken by any organization that wishes to acquire a software system or service.

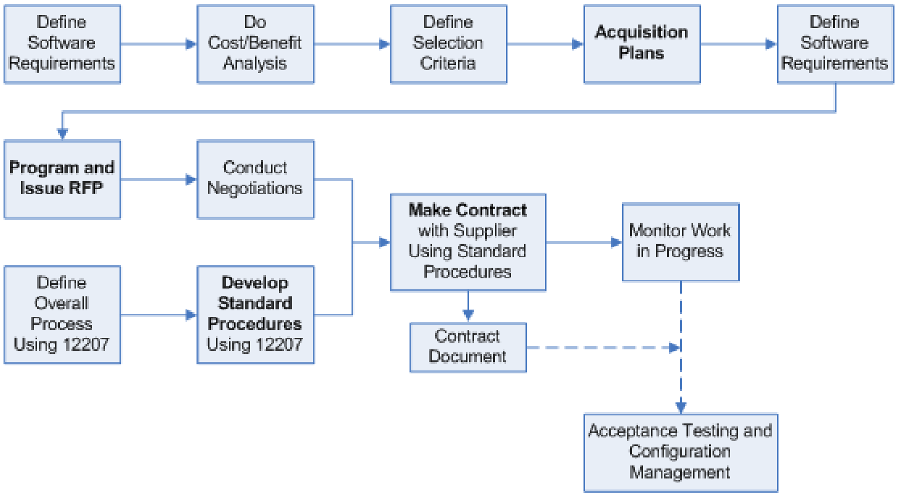
In addition to the activities specified for acquisition, it must also be kept in mind that another process operates in conjunction with it. That process is Supply, which delineates the activities of the vendor. Taken together, these two processes, Acquisition and Supply, describe the standard activities associated with reaching an agreement to provide specific software functionality. Table 1 lists each of the activities and tasks that are specified for Acquisition.

Table 1. Tasks for ISO 12207 Section 5.1, Acquisition Process

5.1.1 Initiation	
5.1.1.1	Prepare a concept or a need to acquire, develop, or enhance a product or service
5.1.1.2	Prepare a set of requirements including relevant design, testing and compliance standards
5.1.1.3	Prepare a risk and cost-benefit analysis for acquisition
5.1.1.4	Prepare a set of acceptance criteria and criteria for evaluation
5.1.1.5	Prepare acquisition plan based on requirements, analyses, and criteria in prior steps.
5.1.2 Request for Proposals	
5.1.2.1	Document acquisition requirements depending on acquisition option selected
5.1.2.2	Tailor processes defined by ISO 12207 as appropriate to meet requirements
5.1.2.3	Define contract milestones
5.1.2.4	Specifically delegate implementation of requirements to responsible organizational entity

5.1.3 Contract Preparation and Update	
5.1.3.1	Establish plans for supplier selection
5.1.3.2	Institute and carry out a negotiation process including contract preparation
5.1.3.3	Institute a process for change control
5.1.4 Supplier Monitoring	
5.1.4.1	Prepare a plan for supplier review
5.1.4.2	Systematically review supplier during product preparation period
5.1.5 Acceptance and Completion	
5.1.5.1	Perform acceptance reviews and testing
5.1.5.2	Institute systematic configuration management

Figure 1 depicts the Acquisition process flow as specified by 12207.



Define Software Requirements
Do Cost/Benefit Analysis
Define Selection Criteria
Acquisition Plans
Define Software Requirements
Program and Issue RFP
Conduct Negotiations
Define Overall Process Using 12207
Develop Standard Procedures Using 12207
Make Contract with Supplier Using Standard Procedures
Monitor Work in Progress
Contract Document
Acceptance Testing and Configuration Management

Figure 1. Standard acquisition process actions

The Memorandum of Agreement

The Acquisition process originates with an expression of the need to obtain a “system, software product or software service” [ISO 1995]. That intention might originate from formal business planning and analysis. Or it might amount to nothing more than a request from a top-level executive to perform a software assisted function. At any rate, the Acquisition process must begin with an initiating decision, which is then documented.

That documentation is usually a memorandum of agreement (MOA) that serves as a pre-contract “contract.” The MOA typically emerges from the business side of the organization. It serves as an internal roadmap that both locates the accountabilities for the conduct of the overall acquisition process and justifies the purchase in business value terms.

The primary advantage of the MOA is that it removes all ambiguity about the roles and responsibilities of the various parties in the rest of the process. From a security standpoint it is essential to get a precise clarification of all necessary commitments and accountabilities. Since software assurance can be a costly process, it is tempting to cut corners. So a clear statement of obligations at the beginning of the project will help to ensure that the security requirements of the process are followed if the going gets tough.

The MOA also specifies, in governance terms, how the product will fit into the overall portfolio of the enterprise. In far too many cases acquisition projects are not managed at the right level in the portfolio. Instead they tend to be carried out as individual purchases independent of the overall plans and strategies of the enterprise. In that respect the “need to acquire” MOA ensures that the product will be correctly integrated into the overall software base and that the processes that do the integration are performed with the maximum attention to good practice.

The Acquisition Project Plan

The express statement that a system, product, or service must be secure and developed in a secure fashion leads to the next step in the process. That is the preparation, after careful analysis, of an acquisition project plan. That plan should have all of the necessary milestones and accountabilities to ensure that the work is done in both a timely and correct fashion. More importantly, it should specify the criteria by which proper performance will be judged.

One of those criteria is the level of desired security. That specification has to be explicit, since security is a property rather than a concrete function, and the presence or absence of a desired property is hard to judge. The definition of terms and the stipulation of the process have to be spelled out in concrete behavioral terms. That includes the description of any quantitative indices that will be employed, as well as the method of analysis.

Once the overall plan is developed, it is approved by all stakeholders. Failure to ensure that all legitimate stakeholders are explicitly committed to project criteria and accountabilities is another common misstep in software acquisitions. In typical projects, once the funds have been authorized, the only people who are involved in the actual build process are the technical people who are responsible for developing and installing the product. That creates the potential for endless confusion at acceptance time, when the deliverable either doesn't meet the customer's business needs or is late or over budget. To help prevent that outcome, the project plan should provide extensive opportunity for all parties to be kept up-to-date and involved.

The Request for Proposals

Then, after the plan is prepared and approved, a Request for Proposals (RFP) is developed and circulated. The RFP is a sent to all logical vendor sources to provide a solution, for a price quoted, that addresses the problem specified in the requirements (usually by a certain deadline). The RFP is the key document in the process, and it is particularly essential from a security standpoint. That is because it is the basis for the contract and the reference by which all of the characteristics that the software must exhibit are judged. Or in simple terms, if it isn't contained in the RFP, it won't be built.

Proper RFP preparation is absolutely essential to security. The RFP is the proxy for the product at the beginning of the process. It serves as the roadmap that guides the purchaser and supplier to the eventual contract. The RFP is also the source by which the eventual deliverable will be evaluated. Thus it establishes the basis for management control over what can be a very involved set of activities. More importantly, a properly prepared RFP, expressed in lay terms, will ensure that decision makers, many of whom are non-technical, are still involved in project oversight and management.

The lack of input from upper management, particularly from the business side, is a chief source of security weaknesses in the contract and the eventual product as built. For example, the failure of the technical side to properly understand all of the applications and markets that the software is intended to be deployed into can lead to insufficient security protection or outright logical errors in the design assumptions. Therefore it is important to involve non-technical, business oriented people in the front end of the process to define all relevant environmental and business considerations.

An excellent article on BSI by Epstein, Matsumoto, and McGraw [Epstein 2006] illustrates the pitfalls of ignoring this requirement. That article points out thirteen security snares associated with ensuring the product:

1. Assuming the vendor will take care of security.
2. Not asking about security at all.
3. Asking about the wrong kinds of security things.
4. Allowing discomfort with the technology to overcome the need for software security.
5. Relying on a cursory risk assessment.
6. Believing you're secure for no apparent reason or for the wrong reasons.
7. Misapplying vulnerability metrics.
8. Trusting the vendors (too much).
9. Building a proof of concept that ignores security "for now."
10. Believing security is somebody else's problem.
11. Giving up hope.
12. Putting too much weight on security standards and security features.
13. Doing it all yourself.

All of these concerns are addressed by a well written RFP. So it cannot be stressed enough that this document is an important part of the strategic governance of the project. Thus it should be considered to be a formal business artifact rather than a purely technical one.

The Software Requirements Specification

The precise aim of the RFP is to communicate unambiguous understanding of the software's functional and non-functional requirements. Thus the aspect that gives the RFP its teeth is a formal and highly detailed software requirements specification (SRS). A good SRS fully itemizes two things:

1. all of the functions required, detailing the externally observable behaviors that will be used to judge whether a specified requirement exists in the code
2. the criteria that the customer will use at the acceptance stage to confirm whether the required functions have been provided

The SRS spells out in formal and contractual terms the exact set of requirements that will be necessary to achieve a satisfactory product solution. The SRS is therefore the single point of failure in the software assurance process, since it is the nexus for the rest of the Acquisition process. Obviously, from a security standpoint the specification should include all of the required security functions, but it can also define general security conditions and properties that the software and the process itself must exhibit.

To ensure this, the SRS document should detail the precise behaviors that the software must exhibit, as well as the criteria that will be used to determine that all requisite security functionality has been provided. Since the latter specification is almost never part of a typical business SRS, it is important from a security standpoint for both the business and the technical side of the operation to get together to make certain that all requisite security properties have been included in the SRS. The practical goal of the SRS is to allow everybody who is going to be involved with the system to contribute to its definition. This insures a complete and systematic specification of all qualitative requirements. But equally as important, it also generates buy-in.

The pragmatic world of business demands an open process. That is because all requisite functionality must be explicitly described in order for it to be supplied by the vendor. Since security is typically not the central point of a purchased piece of software, that property is not always included in a product specification. In fact, since security requirements might actually work at cross-purposes with the needs of the developers, they can be seen as counter-productive. For instance, inspections often have to be championed by somebody in upper management. That is because the time it takes to do things like code reviews becomes an added burden in the production process. So all of the needs of all competing groups have to be laid out and worked into a logically correct approach. That usually requires tradeoffs. An open SRS process provides the forum to do that sort of horse trading. The important point to keep front-and-center during the process is the understanding that security is not an expendable commodity.

Contacting Suppliers

Once the organization has defined all of the things that will be required in the software product, the next step is to contact the people who can provide it. Those people are the suppliers [ISO 1995, Clause 5.2]. The accepted means of initiating contact with a supplier is by sending the Request for Proposals to the widest range of suppliers possible. In that respect it is bad practice to sole source software RFPs, particularly to incumbent suppliers. That is because sole sourcing violates two fundamental business principles. First, a sole source RFP constrains the range of solutions to only those items that the chosen vendor can provide. Second, it eliminates the advantages that come from increasing competitive pressure. Or in simple terms, a vendor receiving a sole source bid will not be encouraged to give the project their best shot because they have no incentive to do so. The lack of motivation has serious ramifications in the case of security. That is because additional security processes and functionality involve additional cost, and if there is no incentive, it is likely that they will be omitted to save money.

Vendors have various ways of receiving RFPs. The easiest way to find out how is to call each and ask. Whatever the approach, a good SRS should leverage the three Ps: performance, price, and protection. In addition it should explicitly focus on deliverables (business has bought a lot of expensive vaporware in the past by buying promises, not products). To enforce a commitment to the deliverables, the SRS should be seen as a pre-contract and expressed in contractual language.

Organizations don’t need an SRS for everything they buy. But a deliberate RFP process offers a number of benefits for any serious purchase. That's because it forces the organization's technical, security, legal, and financial staff to analyze all aspects of the deliverables. And it puts the typical non-technical decision maker squarely in the driver’s seat during the actual negotiation with the supplier. That is because it clarifies all of the requirements of the product at the decision maker’s level and in their terminology.

As we said previously, competitive pressure is a vital element in the Acquisition process because it allows the organization to get the best deal possible. Essentially, if potential suppliers are compelled to compete with each other, the competition will drive down the eventual price as well as ensure that attention is paid to value-added qualities such as security. A supplier's conference is one way of maximizing the features of the deliverable by increased competitive pressure. In that case, the RFP is sent to a large number of suppliers. Then all of the suppliers who received the RFP and wish to bid are called together for a face-to-face conference. There the RFP is explained and clarified in the presence of all the competitors. That encourages the suppliers to bid the maximum amount of functionality at the lowest possible price. Another little known benefit of such a conference is that it might also provide the acquiring organization with valuable input about the actual state of each vendor’s products. That is because every major company does competitive intelligence to find flaws in competitors’ products, and they will be more than willing to share that information with prospective customers if they think it will help their own case.

Evaluating Bids

No matter how the RFP is communicated, by the deadline the interested suppliers will respond to it with a formal proposal, or bid. The organization's next step is to determine which one of these bids comes closest to meeting its needs. There are a number of ways to turn that evaluation into a more exact process. One of the most common approaches in industry is to rank bids based on a risk versus return score. Higher scores are given to projects that meet or exceed expectations for criteria like technical soundness, contribution to the business goal, and overall resource constraints. Tradeoffs from a security standpoint include

> **strategic improvements vs. maintenance of current operations -** Efforts to modernize can actually increase the risk of a security breach because new software takes resources away from the maintenance of older systems. Because legacy code has to be maintained, there has to be an assessment of the value of adding commitments, no matter how beneficial the acquisition might be based on other criteria. Increasing the maintenance operation's workload is always a security risk unless the improvement can be justified.
>
> **high versus low risk -** If the only goal is to minimize risk, the software's actual performance might be constrained. High risk, high return systems can enhance the value of the IT function, provided the risks are capably and carefully managed. Most organizations, however, can only handle a limited number of such projects. As a result, management must always balance the amount of risk they can afford against the ability to manage it.
>
> **impact of one project on another -** Every new system is likely to affect or be affected by the software in the current systems. Management must recognize the context in which the new software will operate and make decisions accordingly. Particularly when it comes to security, compromises in older systems, or in the interface with new systems, often translate to downstream ripples that can very adversely affect the safety and security of the overall organization.
>
> **opportunity costs -** Management must always consider the impact on long-range investment. For instance, will large current investment and operating costs preclude or delay better future opportunities? Will large current capital expenditures create even larger maintenance costs in the future?

After consideration of all of the factors, senior management should have enough information to make knowledgeable investment decisions. The outcome of these comparisons is a set of concrete points of reference that will allow managers to perform the rational tradeoffs necessary to reach an optimal decision about a purchase.

The acquiring organization negotiates a contract after (and only after) the organization has completed the formal evaluation. At this stage, from a terminology standpoint, the organization performing the acquisition is called the owner. "The owner may contract any or all of the acquisition activities to an agent, who will in turn conduct these activities according to the Acquisition process" [ISO 1995].

This is a very important distinction to keep in mind when considering the activities outlined throughout the rest of this section. Because of outsourcing, these actions may be carried out by either the owner or the owner's designated agent. The standard stipulates that in the normal course of the process these activities may involve both entities. But it should be understood that sole responsibility for executing each activity specified must be vested in one of the parties, not both.

STANDARD ACQUISITION ACTIVITIES THAT IMPACT SECURITY INITIATION

To be strictly correct, the acquiring organization should undertake an explicit process aimed at developing a formal statement of intent. The need-to-acquire statement is particularly important from the standpoint of security because it is at this point that the sensitivity and criticality of the information is specified. Obviously, the rest of the process will flow differently if the software is particularly sensitive or critical. So the place where that needs to be established is in the initiating documentation. It is extremely insecure practice to conduct the acquisition activity without an understanding of the environment.

Next, as we discussed previously, the acquirer develops and documents an explicit set of requirements that characterize the identified need in detail. This standard document is known as a Software Requirements Specification (SRS). It expresses each explicit requirement in clear behavioral or functional terms. This deliverable considers all aspects of the acquisition including the "business, organizational and user as well as safety, security, and other criticality requirements along with related design, testing, and compliance standards and procedures" [ISO 1995].

In many ways the SRS is the warhead end of the assurance function in the sense that it characterizes the exact form of the product. Therefore it must be precisely detailed to support all later assessments. There are several ways in which the development of this document can be approached. If the acquirer hires a consultant to carry out the actual problem analysis and description process, the acquiring organization only has to approve the final requirements set. However, it is more likely that the acquiring organization will do the actual analysis and description activity. Nevertheless, the eventual product of this step is a precise specification of the functional and nonfunctional requirements of the software that is to be acquired. This document must be fully inspected and approved by all parties.

It is at this point, and no other in the process, that the security requirements are specified. That is because in order to ensure that the problem space is completely and correctly defined, all necessary functionality has to be described as an integrated set. No software element functions in a vacuum, and so in order to ensure the proper relationships all components have to be detailed and their interconnections specified. As we said earlier, the acquirer has to consider a range of tradeoffs in deciding which potential course of action to take in actually acquiring the product.

This is true even in the case of decision making about the options for conducting the actual Acquisition process itself. Generally that includes a thorough analysis of the risk, cost, and benefit for each potential acquisition option. By stipulation these options include [ISO 1995]

- Purchase an off-the-shelf product that can be proven to satisfy the requirements.
- Develop the software internally.
- Develop the software through contract.
- A combination of the first three options.
- Enhance the existing software.

All of these have security risks associated with them. Even with EAL levels, the purchase of COTS products is at best a speculative exercise, since the structure of the underlying code is essentially unknown. In the case of internal development, the structure might be known but the costs of assuring it are usually astronomical. So the organization runs the risk of sacrificing security for cost savings. Although some visibility can be built into the contracting process, the actual work is in the hands of a third party and that fact carries many of the risks implied by the first two items. Enhancing a current product might not meet the strategic purposes of the organization. Consequently the strategy that is selected must be examined from every one of these perspectives to select the one that satisfies the specific security requirements of the situation.

The final stage in the initiation process is the development, documentation, and execution of the acquisition plan. Again, the standard specifies that this plan embrace the following elements [ISO 1995]:

- requirements for the system
- planned employment of the system
- type of contract to be employed
- responsibilities of the organizations involved
- support concept to be used
- risks considered as well as methods to manage the risks
- acceptance strategy and conditions (criteria)

RFP Preparation

Obviously the form and content of the actual acquisition requirements (RFP) depend to a great extent on the acquisition option selected (above). By standard the RFP must include, as appropriate [ISO 1995]

- system requirements
- scope statement
- instructions for bidders
- list of similar products
- terms and conditions
- control of subcontracts
- technical constraints (e.g., target environment)

If the stipulations of the standard are adhered to at this point it is likely that the eventual deliverable will be secure. That is because all of the typical points of failure will have been considered and the optimum solution selected. That is particularly true in the case of “control of subcontractors,” since outsourcing is a likely source of breach.

At this point, and employing the stipulations of the 12207 Standard (Annex A and B), the acquiring organization will also tailor out the operating procedures for the acquisition project itself. That includes a complete description of all of the relevant security processes, activities, and tasks that fit the particular environment of the project. Of primary importance at this stage is the requirement that the acquirer must specify in detail any and all applicable supporting processes (such things as the attendant documentation, quality assurance, and configuration processes). That includes everything that has to do with assurance of the product and process.

In addition the acquirer must identify who (or alternatively which organization) will perform each supporting and assurance process. That includes an unambiguous characterization of the various roles and responsibilities of the individual parties involved in the conduct of each supporting process, as well as their accountabilities. This is a necessity for the practical reason that the supplier has to know what the acquiring organization’s exact requirements are with respect to required operating processes. This has to be specified before the supplier can begin to develop an intelligent response.

RFPs generally also define the milestones at which the supplier’s progress will be reviewed. These milestones are critical from a security standpoint since they lock the supplier into a set schedule of joint reviews. It is at this stage in the process that tactical decisions that are made about the development can be reviewed and altered if necessary.

Finally, because the owner of the system may not be the organization actually doing the acquisition (for instance in the case where a consultant does this for a client, or in contracted and outsourced situations), the standard requires that “the requirements should be given to the organization that actually performs the acquisition activities” [ISO 1995].

One of the real advantages of the process specified in the 12207 Standard is that they are logically complete. The idea that a third party might actually do the acquisition always underlies the process. However, since it is a contingency rather than standard operating procedure, the need to ensure coordinated understanding between multiple parties is often overlooked. The standard ensures that coordination is built into the process at all levels.

Contract Preparation and Update

As we said it is the general responsibility of the acquiring organization to adopt a formal model for supplier selection. This usually includes considerations such as standard selection methodology (employed to evaluate the proposal) and the evaluation criteria. In addition such particulars as the weights and method employed to assess compliance are usually specified in advance in the RFP. This is the point in the process where the organization’s upper-level decision makers can exercise direct control over the process of ensuring a secure product. That is because the security is shaped through the criteria that are employed. These are embedded here.

The actual assessment and selection then is based on the supplier’s demonstrated capability to deliver the system, software product, or service as specified. This is evaluated based on the factors itemized in the criteria for evaluation. The standard Acquisition process that we are discussing here must be tailored for application to a given project. The acquirer may “Involve other parties, including potential suppliers, before contract award, in tailoring these standard processes.” However, by rule, “The acquirer will make the final decision on the tailoring” [ISO 1995]. To comply with the requirements for supplier selection that are outlined in the standard, the acquirer must also “include or reference the tailored process in the contract” [ISO 1995].

This final item makes eminent good sense both because use of a standard acquisition process is a relatively novel approach and because the terms and conditions imposed by the defined processes have cost implications that the supplier needs to know about before formulating the contract. The final stage in the contract preparation stage is the negotiation and signing of the contract itself. This is a security requirement in and of itself. That is, the contract must be assured to address all documented requirements of the acquisition project, including cost and schedule. The contract normally addresses such legal issues as usage, ownership, warranty, and licensing rights associated with the product or service. However, all of this should be confirmed by a third party source such as an auditor or a software contract legal specialist.

The legal form of the contract itself is monitored and controlled through a formal change control mechanism. Generally the acquiring organization is responsible for the control of any and all modifications to the contract that might result from ongoing refinement of the understanding of the requirements. Contract control is a particularly important aspect of ensuring security and it is often overlooked. However, as a result of a lack of control serious security vulnerabilities can be introduced into the build downstream in the process because of undocumented changes. Since these are not part of the documentation, they are never inspected and rectified.

Inspection outcomes are usually negotiated with the supplier under the Joint Review process and resolved under the Problem Resolution process. These are both supporting processes that are specified in the standard. Under these processes all changes to the contract are analyzed to determine their potential impact on project plans, costs, benefits, quality, and schedule.

Supplier Monitoring

Once the contract is agreed on, the acquiring organization monitors the activities of the supplier throughout the life cycle of the development process. This is done within the specifications of the 12207 Standard's SQA, Joint Review, and Audit supporting processes. In addition the standard recommends that the acquiring organization augment the monitoring function with the standard’s Verification process and Validation process, as needed. As part of Joint Review, the standard requires the acquiring organization to provide all necessary information required by the supplier “In a timely manner and resolve all pending items” [ISO 1995]. Obviously, since the actual development process is in the supplier’s hands, a robust monitoring process is an essential part of ensuring security.

Acceptance and Completion

Eventually the supplier will be ready to deliver the completed system, software product, or service to the acquiring organization. When that time comes, the acquiring organization must be prepared to conduct a formal acceptance activity. This acceptance procedure and its associated criteria are usually itemized in great detail in the contract, which is the primary artifact used to control acceptance of the final product. The standard recommends inclusion of such items as “Preparation of test cases, test data, test procedures, and test environment.” In addition, “The extent of supplier involvement should be defined” [ISO 1995].

Using that acceptance document, the acquiring organization conducts acceptance reviews and tests of the deliverable. That includes examination and confirmation of all security functionality specified in the requirements specification. The product is considered accepted when all acceptance conditions are satisfied.

During development, the artifact is under control of the supplying organization's configuration control system (everything except the contract itself, which is controlled by the acquiring organization throughout the process. Therefore, after acceptance, the acquiring organization must make arrangements to migrate the artifact from the Supplier's configuration management system to its own operation.

SUPPLY: THE ESSENTIAL OTHER PROCESS

If there is an acquirer, there is always a supplier. So in essence in order for the Acquisition process to be complete, it is necessary to understand the other side of the coin. These two processes are inseparable from the standpoint of ensuring adequate and proper security functionality, and so it is not a simple matter of viewing the Supply process as an adjunct to Acquisition. In essence, the Supply activities are so tightly integrated with the activities of the acquirer that this should be considered a single process. That unification of purpose is so important that the acquirer-supplier relationship has been characterized in later models as a single “Agreement” process (see [ISO 2002]).

12207 recognizes the inseparability of supplier from acquirer by defining the activities and tasks carried out by the supplier in reference to the Acquisition process. Accordingly, the activities outlined in Supply should be considered as part of overall agreement on the form of the product. The Supply activities itemized by the standard represent the current best practices deemed necessary to provide an acceptable solution to an acquiring organization's needs. Table 2 summarizes the Supply activities.

Table 2. Itemization of Tasks for 12207 Section 5.2, Supply Process

5.2.1 Initiation	
5.2.1.1	Review requirements in RFP.
5.2.1.2	Decide to bid or accept the contract.

5.2.2 Preparation of response	
5.2.2.1	Define and prepare a proposal including recommended tailoring of ISO 12207.
5.2.3 Contract	
5.2.3.1	Negotiate and enter into a contract with the acquirer organization.
5.2.3.2	Supplier may request modification to the contract as part of change control.
5.2.4 Planning	
5.2.4.1	Define framework for project management and quality assurance.
5.2.4.2	Select software life cycle model;…map ISO 12207 onto life cycle model.
5.2.4.3	Design management and quality assurance plan, including acquirer involvement.
5.2.4.4	Consider options for developing software product against risks of each option.
5.2.4.5	Develop project management plan(s) based on the planning requirements.
5.2.5 Execution and control	
5.2.5.1	Implement and execute project management plan(s).
5.2.5.2	Develop, operate, and maintain in accordance with ISO 12207 Clauses 5.3, 5.4 and 5.5.
5.2.5.3	Monitor and control progress of the software throughout contracted life cycle.
5.2.5.4	Manage and control subcontractors in accordance with 5.1.
5.2.5.5	Interface with independent verification, validation, or test agent.
5.2.5.6	Interface with other parties specified in contract and project plans.
5.2.6 Review and evaluation	
5.2.6.1	Coordinate contract review, interfaces, and communication with acquirer.
5.2.6.2	Hold joint meetings, acceptance reviews and testing, joint reviews, and audits.

5.2.6.3	Perform verification and validation in accordance with 6.4 and 6.5 respectively.
5.2.6.4	Provide reports, reviews, audits, testing, and problem resolutions to acquirer.
5.2.6.5	Give acquirer access to supplier and subcontractor facilities for review product.
5.2.6.6	Perform quality assurance activities in accordance with 6.3.
5.2.7 Delivery and completion	
5.2.7.1	Deliver the software product or service as specified in the contract.
5.2.7.2	Provide assistance to acquirer in support of the delivered software product.

The Supply process is essentially a project management activity. It coordinates and monitors the general function of providing a specified solution to a customer. It does not, strictly speaking, embody development tasks. Instead it is chiefly composed of various planning and control functions.

The Supply process usually begins with the receipt of an acquirer’s RFP. The first step after that event is to determine the specific resource requirements, which is really just good project management practice. From the standard’s perspective, Supply does not strictly involve development. To achieve its ends, the Supply process will employ most aspects of the Development, Maintenance, and Operation Primary processes. However, it must be noted that these are separate processes specified elsewhere in the standard.

The point that must always be kept in mind is that development, operations, and maintenance are different activities that are meant to work in a complementary fashion. This is an important distinction to make, since the steps in both Acquisition and Supply are not technical per se. Instead they are intended to coordinate the more technical processes specified in the Development and Operations sections of the standard.

Additionally, if subcontractors are used, the activities specified in the Acquisition process will also come into effect as a subset of Supply. In creating the overall process framework, the supplier should adopt two general organizational processes, Management and Infrastructure process development. Finally, as with Acquisition the supplier must tailor individual project processes to the specifications of the 12207 Supply process using the Tailoring process outlined in annex A of the standard.

Supply Initiation

The Supply process starts when a supplier organization receives an RFP from a potential customer. Once this document has been received, common sense would dictate that the first task would be to conduct a thorough review of the acquirer's RFP. This is done in order to identify all of the potential constraints on the problem as well as to inventory potential solutions.

The goal is to define the product space. That space is inhabited by all of the possible solutions that will satisfy all constraints. That includes all specified security considerations such as sensitivity and reliability. Obviously, tradeoffs are a necessity, since in most cases highly constrained sensitivity requirements will (for instance) impact the general availability and performance of the system.

Within the product space domain, any product would be acceptable. Each would exhibit different external behavior but all would satisfy all constraints. So in simple terms, the outcome of the problem definition activity is a characterization of the range of practical solutions that would meet all of the known constraints. Sources of constraints may include

- users - who have to decide what they need rather than what they want
- customers - who often stress financial factors over function
- technology - where the solution requires leading the target
- security - where there are specific security requirements (e.g., federal systems)
- laws - violations of which will produce unacceptable solutions (e.g., SOX)
- standards - which if violated could produce unacceptable solutions

In actuality the normal result of constraint identification is the creation of a negative problem space, which happens when two requirements can't fit the same project (for example, when the desired technology costs too much). So the real task of constraint identification is to perform tradeoffs. Or in simple terms one constraint is relaxed to accommodate another (for example, if we leave out this function we can afford the system). Both the acquiring and supplying organization's policy and procedure rulebooks must be consulted at this point to provide the acceptable terms for this analysis. Once everybody is satisfied that the RFP is both feasible and cost justifiable the supplying organization makes a decision as to whether to bid or accept the contract.

From the standpoint of software security this commitment process should be formally documented. That is because the cost of ensuring that the code is secure rather than just operational can be prohibitive. However, those costs don’t begin to appear until downstream in the process. What this leads to is cost cutting, which can be risky where software assurance is concerned. As a consequence, a signed commitment at the beginning of the bidding process is needed to lock in the accountabilities required to ensure that the project is executed as planned.

Preparation of a Response

Once the decision has been made to bid, the supplying organization puts a formal bid proposal together and communicates it to the acquiring organization. This proposal serves as the formal response to the RFP. There is one item in this activity that needs to be noted because it is not typical operating procedure for most bidding processes.

As an adjunct to submission of the bid, the standard also requires that the supplying organization stipulate in its document a recommended approach to tailoring the provisions of the standard to the execution of the potential project. Since the standard is the controlling framework for the entire process, it is important that everybody involved be on the same page when it comes to rolling out the project. Otherwise the assurances concerns that arise from a lack of coordination will most likely occur. As such, prior to submission of the bid document it is important for the supplying organization to get a minimum understanding of the requirements and workings of the standard and translate that into a project architecture.

The Contract

If the bid is accepted, the supplier negotiates a contract with the acquiring organization. Once entered into, this document becomes a legally binding agreement. Alterations to the contract may take place in the course of events. However, if this is the case these changes must be dealt with within the context of the conventional legal system.

It should be noted that the standard stipulates that the responsibility for ongoing maintenance of changes to the agreement rests with the acquiring organization. If there is any alteration to the contract, the new form of the agreement is kept by the acquirer in the contract change control system.

Maintenance of the contract under strict change control is an absolute necessity from a security standpoint. That is because of the potential for undocumented requirements and the subsequent code. It is impossible to conduct a rational process if some of its elements are unknown, so the logic of the need for contract control should be straightforward. However, the realities of the development process can make strict control of changes a difficult task. That is why it is important to have a process in place that is both understandable to all participants and stringently adhered to.

Planning

Usually the supplying organization is responsible for the development of the management and project commitment plan (unless otherwise stipulated in the RFP). This comprehensive plan is generally developed straight out of the requirements provided by the acquirer. The supplying organization establishes the procedures for executing and overseeing the project directly from the detailed specifications itemized in that document.

This project plan should include the resource estimates and definitions of the scope and extent of acquirer involvement at each stage. Ultimately, in addition to the overall plan for the work, the supplier is also responsible for developing a framework for software assurance. And, if not otherwise stipulated in the contract, the supplier is also responsible for the selection of a life-cycle model appropriate to the "scope, magnitude, and complexity of the project." Once this is done, the standard requires that the supplier map the processes, activities, and tasks onto this model.

Finally, when all of the planning assumptions have been laid down, the supplying organization decides on the best approach to the delivery of the product contracted for. This is done by weighing the relative advantages of the various approaches against the risks associated with each. The standard cites the following options as viable [ISO 1995]:

1. Develop the software product or provide the software service using internal resources.
2. Develop the software product or provide the software service by subcontracting.
3. Obtain off?the?shelf software products from internal or external sources.
4. A combination of a, b, and c above.

These are the same options that the acquirer considered, and the supplier undertakes the same consideration process in resolving them. However, once this assessment is complete and the decision is made, the supplying organization must create an actual management plan for that particular project. This plan evolves directly out of the established organizational architecture of the supplier and the general supporting process model and the strategic development options dictated by the standard. According to the ISO 12207 Standard, the following items (among others) can be considered in the plan [ISO 1995]:

- project organizational structure and authority and responsibility of each organizational unit, including external organizations

- engineering environment (for development, operation, or maintenance, as applicable), including test environment, library, equipment, facilities, standards, procedures, and tools
- work breakdown structure of the life-cycle processes and activities, including the software products, software services, and non?deliverable items to be performed, together with budgets, staffing, physical resources, software size, and schedules associated with tasks
- management of the quality characteristics of the software products or services (separate plans for quality may be developed)
- management of the safety, security, and other critical requirements of the software products or services (separate plans for safety and security may be developed)
- subcontractor management, including subcontractor selection and involvement between the subcontractor and the acquirer, if any
- quality assurance (see 6.3)
- verification (see 8.4) and validation (see 6.5), including the approach for interfacing with the verification and validation agent, if specified
- acquirer involvement, that is, by such means as joint reviews (see 6.6), audits (see 6.7), informal meetings, reporting, modification and change; implementation, approval, acceptance, and access to facilities
- user involvement by such means as requirements-setting exercises and prototype demonstrations and evaluations
- risk management, that is, management of the area of the project that involves potential technical and scheduling risks
- security policy, that is, the rules for need?to?know and access to information at each project organization level
- approval required by such means as regulations, required certifications, proprietary, usage, ownership, warranty, and licensing rights
- means for scheduling, tracking, and reporting
- training of personnel (see 7.4)

Execution and Control

Once the planning and management framework has been established, the supplying organization does the work specified in the contract. By the terms of the standard, all development, operations and maintenance work is dictated by the processes, activities, and tasks stipulated under ISO 12207 Development process (5.3), Operation process (5.4), and Maintenance process (5.5).

For the duration of the project, the supplying organization oversees and controls the work in progress. This includes all of the stipulated assurance elements of the plan. It should be noted here that these fundamental practices are carried out as a discipline, in the sense that the oversight, control, and qualitative components of the plan are enforced throughout the period of the contract and for the life cycle. The standard stipulates that this is a continuous recurring operation, which allows the supplying organization to both oversee the progress of the technical work, costs, schedules, and project status and to identify problems as they occur and record the resolutions.

In addition, the supplier practices strict oversight and control over any and all subcontractors who might be involved in a given project. This is done in accordance with the entire set of stipulations of the Acquisition process (5.1). What this means is that the supplying organization is responsible for communication with the sub-contractors. That responsibility encompasses all of the requirements necessary to ensure that every aspect of the software artifact, including those elements of the system, software product, or service prepared by the subcontractor, meets the requirements of the contract.

In order to ensure that this occurs, the supplier has the responsibility to work with any contractually required verification, validation, or test representative. Finally, the standard defines a sort of “other” category for any unforeseen agencies that the supplier might interact with, in the sense that it requires that the “supplier shall interface with other parties as specified in the contract and project plans” [ISO 1995].

Review and Evaluation

Reviews are very important in supply operations because they enforce project oversight and control features. By standard the supplier is required to “coordinate contract review activities, interfaces, and communication with the acquirer’s organization” [ISO 1995]. There is a range of possible reviews that might take place. These are much too extensive to detail here but they all fall under the generic heading of a software technical reviews.

Generally these reviews can be factored into two categories, formal and informal. The key to identifying which type of review it is lies in who controls the review. If the producer does the presentation of the artifact, the review is considered informal. This type of examination is usually called a walkthrough. If the artifact is presented to a team or individual for examination prior to the actual review, it is formal and usually called an inspection, or audit.

Walkthroughs are very common in the software industry. They can be as informal as one programmer demonstrating a segment of code to another and asking for help. Or they can be regularly scheduled sessions involving whole teams or groups of individuals. The advantage of walkthroughs is that very little time is consumed, so the resource commitment and cost is negligible. The disadvantage of walkthroughs is that they are not rigorous. The producer controls the presentation, which means that they can walk a reviewer past holes in the logic that neither one will be able to see or do anything about.

Inspections and audits have rigor but they require resources and have concomitant costs. The whole point of an inspection review is that the producer surrenders the artifact for analysis by a third party. That third party can be either external or internal, but the analysis is essentially performed outside of the production process. This means that there is a much higher error detection and correction rate. In addition, inspections always generate documentation in the form of reports and recommendations. Inspections and audits are almost always scheduled events. They are most often part of the fabric of the project plan and contract.

In most cases (usually by stipulation of the standard) the supplier performs these walkthroughs, inspections, and audits (including acceptance testing) with representatives of the acquiring organization. The standard requires that joint review activity should be "Conducted in accordance with Join Review 6.6, and Audits 6.7. The supplier shall perform verification and validation in accordance with 6.4 and 6.5 respectively to demonstrate that the software products or services and processes fully satisfy their respective requirements...The supplier shall perform quality assurance activities in accordance with 6.3" [ISO 1995].

Finally, the essence of the inspection process is the reports that issue out of the technical reviews. The supplier is obligated by standard to make the results of all evaluations, reviews, audits, tests, and problem resolution meetings available to the acquirer. This is usually specified in the contract. The standard also requires that the supplier "Provide the acquirer access to the supplier's and subcontractors' facilities for review of software products or services as specified in the contract and project plans" [ISO 1995].

Delivery and Completion

Finally, the supplier delivers the system, software product, or service to the acquiring organization. In order to smooth the transition when the artifact is completed, the standard also stipulates that the supplier must commit to provide routinely contracted support service to the acquiring organization.

SUGGESTIONS FOR FURTHER READING

1. Berry, John. "IT ROI Metrics Fall Into Four Groups." Internet Week, July 16, 2001.

2. Boehm, B. “Improving Software Productivity.” Computer 20, 9 (September 1987).
3. Boehm, Barry W. “Software Engineering Economics." IEEE Transactions on Software Engineering SE-10, 1 (January 1984).
4. Broadman, J. G. & Johnson, D. L. “Return on Investment from Software Process Improvement.” Software Process - Improvement and Practice, July 1995.
5. Brynjolfsson, Erik. “The Productivity Paradox of Information Technology.” Communications of the ACM 36, 12 (December 1992): 67-77.
6. Card, D. & Comer, E. “Why Do So Many Reuse Programs Fail?” IEEE Software 11, 5 (September 1994): 114-115.
7. Curtis, W. “Building a Cost-Benefit Case for SPI.” SEPG 1995 (CD-ROM). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1995.
8. Dart, Susan A. "Achieving the Best Possible Configuration Management Solution.” CrossTalk, September 1996.
9. Cisco Systems, Inc. "Defense Agencies Meet Readiness Challenges with Commercial off the Shelf (COTS)-Based Systems” (A Cisco Intelligent Network White Paper). Cisco Systems, 2003.
10. Dion, R. “Process Improvement and the Corporate Balance Sheet.” IEEE Software 10, 4 (July 1993): 28-35.
11. Dorofee, A. J.; Walker, J. A.; & Williams, R.C. "Risk Management in Practice." CrossTalk 10, 4 (April 1997).
12. Dover, Sanford. “A Standard Response.” CIO, June 1993.
13. Edelstein, Vera, Roger Fujii, Craig Guerdat, Pasquale Sullo Internationalizing software engineering standards Computer, Volume 24 , Issue 3, March 1991
14. OMB. Evaluating Information Technology Investments. Office of Management and Budget, 1999.
15. Feiler, Peter. Configuration Management Models in Commercial Environment (CMU/SEI-91-TR-007, ADA235782). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1991.
16. Fenton N. “How Effective are Software Engineering Methods.” Journal of Systems and Software 22 (1993).
17. Hayes, W.; & Zubrow, D. “Moving On Up: Data and Experience Doing CMM Based Software Process Improvement.” SEPG 1995 (CD-ROM). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1995.
18. Hersleb, J.; Zubrow D.; Siegel J.; Rozum, J.; & Carlton, A. “Software Process Improvement: State of the Payoff.” American Programmer 7, 9 (September 1997).

19. Humphrey, Watts S. A Discipline for Software Engineering. Reading, MA: Addison-Wesley, 1995.
20. Humphrey, Watts S. Managing the Software Process. Reading, MA: Addison-Wesley, 1994.
21. Humphrey, Watts S. “Comments on a Critical Look.” IEEE Software, July 1991.
22. Humphrey, Watts S. & Sweet, W. A Method for Assessing the Software Engineering Capability of Contractors (CMU/SEI-87-TR-023, ADA187230). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1987.
23. International Organization for Standards. ISO/IEC 12207. Geneva, Switzerland: ISO, 1995.
24. International Organization for Standards. TR-15504. Geneva, Switzerland: ISO, 1998.
25. Jones, Capers. “The Pragmatics of Software Process Improvements.” Software Engineering Technical Council Newsletter 5 (Winter 1996).
26. Jones, Capers. “Software Defect Removal Efficiency.” Computer 29, 4 (April 1996).
27. Joos, R. “Software Reuse at Motorola.” IEEE Software, September, 1994.
28. Keefe, Patricia. “Take a Bold Step.” Computerworld, July 30, 2001.
29. Koller, Mike. “Accurate ROI Requires Impartiality.” Internet Week, September 25, 2001.
30. Kotler, Tony. “A Hard Lesson in ROI.” Line56.com, July 12, 2001.
31. Lee, E. “Software Inspections: How to Diagnose Problems and Improve the Odds of Organizational Acceptance.” CrossTalk 10, 8 (1997).
32. Lewis, David. “Demand for ROI Rises.” InternetWeek, March 27, 2001.
33. Lim, W. C. “Effects of Reuse on Quality, Productivity and Economics.” IEEE Software, September 1994.
34. Lipke, W. & Butler, K. “Software Process Improvement: A Success Story.” CrossTalk, Number 38, November 1992.
35. Manas, Todd. “Making the Balanced Score Card Approach Pay Off.” ACA Journal 8, 2 (1999).
36. Marash, S. “The Future of ISO 9000.” The Corporate Board, May 1994.
37. Marshall, Alexa. “Software Configuration Management: Function or Discipline?” CrossTalk, October 1995.
38. Mayor, Tracy. “Value Made Visible.” CIO, May 1, 2000.
39. McGarry, F. & Jeletic, K. Process Improvement as an Investment: Measuring its Worth. NASA Goddard Space Flight Center, SEL-93-003, 1993.
40. McGibbon, Thomas. A Business Case for Software Process Improvement Revised. DoD Data Analysis Center for Software (DACS), 1999.
41. Miller, Dan. “The ROI on CRM.” The Industry Standard, August 6, 2001.

42. National Infrastructure Advisory Council (NIAC). National Strategy to Secure Cyberspace. Office of the President, 2004.
43. O'Brien, M. Software Production Management. Oxford, U.K.: NCC Blackwell Ltd., 1992.
44. Paulk, M.; Curtis, B.; Chrissis, M.; & Weber, C. Capability Maturity Model for Software (Version 1.1) (CMU/SEI-93-TR-024, ADA263403). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1993.
45. Roach, Stephen S. "Services Under Siege—The Restructuring Imperative." Harvard Business Review, Sept.-Oct. 1991, pp. 82-92.
46. Rozum, J. Concepts on Measuring the Benefits of Software Process Improvement (CMU/SEI-93-TR-009, ADA266994). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1993.
47. Shoemaker, Dan & Jovanovic, V. "ISO 9000: The State of the American Software Industry. Journal of Computer Information Systems, Winter 1996.
48. Skamarock, Anne. "Quantifying ROI." NetworkWorldFusion, July 9, 2001.
49. Software Engineering Institute Web site, www.sei.cmu.edu.
50. Strassman, P. A. The Business Value of Computers. New Canaan, Connecticut: The Information Economics Press, 1990.
51. Tomayko, James. Software Configuration Management, Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1997.
52. Violino, R. "Measuring Value: Return on Investment." Information Week 637 (June 30, 1997): 36-44.
53. Yamamura, G. & Wigle, G. B. "SEI CMM Level Five: For the Right Reasons." CrossTalk 10, 8 (August 1977).
54. Zimmerman, Michael. "Configuration Management: Just a Fashion or a Profession?" White Paper, usb GmbH, 1997.

REFERENCES

[Cisco 2003]

Cisco Systems, Inc. "Defense Agencies Meet Readiness Challenges with Commercial off the Shelf (COTS)-Based Systems" (A Cisco Intelligent Network White Paper). Cisco Systems, 2003.

[Epstein 2006]

Epstein, Jeremy; Matsumoto, Scott; & McGraw, Gary. "Software Security and SOA: Danger, Will Robinson!" IEEE Security & Privacy 4, 1 (January/February 2006).

[ISO 1995]

International Organization for Standardization (ISO). Standard for Information Technology – Software Life Cycle Processes (ISO/IEC 12207:1995). Geneva, Switzerland: International Organization for Standardization, 1995.

[IEEE 1998]

IEEE/EIA. IEEE/EIA Standard Industry Implementation of International Standard ISO/IEC 12207: 1995 (ISO/IEC 12207) Standard for Information Technology Software Life Cycle Processes (IEEE/EIA 12207.0-1996). New York: IEEE, March 1998.

[ISO 2002]

International Organization for Standardization (ISO). *Standard for Systems Engineering* – System *Life Cycle Processes (ISO/IEC* 15288:2002*)*. Geneva, Switzerland: International Organization for Standardization, 1995.

[NIAC 2004]

National Infrastructure Advisory Council (NIAC). The National Strategy to Secure Cyberspace. Office of the President, 2004.

This material is based upon work funded and supported by Department of Homeland Security under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center sponsored by the United States Department of Defense.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of Department of Homeland Security or the United States Department of Defense.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS-IS” BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

DM-0001120