Software Engineering Institute | Carnegie Mellon University

Finding a Vendor You Can Trust in the Global Marketplace

Art Conklin

Dan Shoemaker

August 2008

ABSTRACT: This article introduces the concept of standardized third-party certification of supplier process capability. It is based on the principle that capable suppliers provide capable products. At a time when security concerns are preeminent, it should be clear that purchasing software from unknown or unvetted suppliers is a risky proposition, and the trend toward global outsourcing only exacerbates the problem. Yet businesses lose competitive advantage if they deal only with local suppliers. The approach outlined here allows acquirers to trust even previously unknown suppliers if those suppliers have undergone third-party assessment of the security capability of their processes. It allows acquisition officers to deal with a much wider range of suppliers and increases the competitive pressure necessary to ensure cost-efficient products.

THE HAZARDS OF SPREADING THE NET

Off-the-shelf products have a considerable business advantage. They are generally less costly than custom work, and they are usually immediately available. Nevertheless, if it is hard for a company's internal programming staff to guarantee bug-free code, think about the odds against getting defect-free software from programmers who work on the other side of the world. That is the reason why it is so critical to be able to identify vendors who can produce secure code, and that prospect gets us around to the need for reliable certification of supplier capability.

It is generally understood that the quality of a product is tied to the capability of the people who create it [Humphrey 1987, Jones 1994, McGibbon 1999, and Paulk 1993, to name a few]. Consequently, since vulnerabilities arise from software defects, it is important to be able to ensure that your software is developed by the most capable organizations. The problem is that the marketplace is essentially anonymous. So an acquisition officer is left with only two options: (1) deal with the usual suspects, all of whom know that you are their captive, or (2) buy a product from an unfamiliar vendor and hope that it isn't malicious.

Since there is almost no justification for the latter approach, the former is typically the method of choice, even though taking competitive pressure out of the

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

bidding potentially means higher prices and less service. The ideal, of course, would be to maximize competitive pressure by spreading the net as wide as possible. The problem is that it would require precise knowledge of the exact level of capability of each of the suppliers who swam into the net, and it is impossible to know the capabilities of every potential supplier given the global nature of the software business.

On the other hand, if the level of capability of every supplier was recorded somewhere, it would then be a simple matter of dealing with the ones who had the requisite level of capability. A similar concept has been used for years in the quality universe in the form of the ISO 9000 quality system registries. So there is no reason why registrations of security engineering capability cannot be kept. In fact, a third-party registry system is already in place for certifications under ISO 15408, colloquially known as the Common Criteria.

However, there are three serious problems with the Common Criteria (CC). First, although the Common Criteria Recognition Arrangement (CCRA) allows the CC to have international effect, that agreement has been signed by only 14 countries, none of which is in Asia [Jackson 2007]. Thus, it is not a globally accepted model. Second, the CC doesn't actually specify what security functions should be incorporated into the product. Instead, it evaluates the product against a protection profile (PP) put together by the customers. The evaluation then simply determines whether the product has walked the talk [Jackson 2007].

From a business case standpoint, neither of these first two are fatal flaws. However, the third concern is a definite showstopper. CC certifications are only for individual products. The cost of certifying each product is relatively prohibitive and—even worse—the effort and time required to prepare evaluation evidence and other evaluation-related documentation is so cumbersome that by the time the work is complete, the version under evaluation is generally obsolete [Jackson 2007].

THE BUSINESS CASE FOR CERTIFYING TRUST

The Common Criteria process might be costly and a little cumbersome. But the idea of standardized third-party certification of the precise level of security capability is very attractive. From a business standpoint, a financially justifiable and generally accepted certification would accomplish two valuable aims. First, the acquisition community would have objective independent assurance that they could trust a supplier, even if that supplier were in another culture 3,000 miles away. Second, suppliers would be able to gauge their exact level of capability and then determine how to improve it.

Those twin advantages would reduce uncertainties in the acquisition process by identifying the risks associated with a given contractor. Moreover, by identifying the inherent risks, standard certification of capability would also provide an objective basis for trading off business requirements and project cost against each vendor's areas of weakness when contract time rolls around. Finally, given the inter-dependence between process capability and the quality of the product, certification would also represent the best means of putting appropriate risk controls in place for whoever was eventually selected.

Thus, the opportunity to manage risks while increasing competitive pressure in the bidding process makes a strong case for some form of universal certification of supplier process capability. Given that, the first order of business is to find a reasonable approach. This paper introduces a simple means for characterizing supplier security confidence based on a commonly accepted model.

THE FORM OF THE ASSESSMENT

The problem with individual product certifications is that they are costly and potentially very cumbersome. Nonetheless, audited, third-party certification is almost the only way to ensure a trust relationship between anonymous acquirers and suppliers, particularly in a global marketplace. So the question is, "what would be the most effective basis for a certification process?"

In the Common Criteria, the requisite engineering and supporting processes are specified through a standard protection profile (PP). This PP is then written into the contract. The approach we are advocating here is similar to that. However while CC ratings are given for a specific piece of software, in the case of this approach, the targets of evaluation (TOE) are the security engineering processes that the supplier employs to produce all of its products. The business advantage is that the focus is on the overall production process rather than each individual product. It would allow supplier organizations to certify their competency with a single external audit, which is much less costly and cumbersome than having to test each individual product.

Just like the Common Criteria, our approach allows the capability of a given supplier's development process to be evaluated against a requisite process capability profile, which is prepared in advance by the customer. The assessment then builds a set of process ratings for each individual supplier that can be aggregated into a general supplier capability rating. The rating for each supplier can then be compared to the target capability requirements for that particular project.

The protection profile we are suggesting is based on a two-dimensional model, which is designed to characterize the specific capability level of each given process. The first dimension of this model is the process dimension. The process dimension defines the base practices associated with a given level of capability. This sort of process capability paradigm has been used since 1987 to characterize the general trustworthiness of a software organization's processes [Humphrey 1987]. And models such as CMM, CMMI, and SPICE are all part of that culture of capability determination. Consequently, the idea of employing a process-based capability assessment to certify the relative security capability of the organization is not a radical new concept [McGibbon 1999].

All capability assessments are based on a standard set of best practices. These practices represent the measurable objectives, or outcomes of the process. Nevertheless, the chief problem with using activities selected from any of the models listed above is that, although they are all well established as definitions of capability maturity in software and system development, these models do not specifically address good security engineering practice. However, there is one capability maturity model that does. That is the Systems Security Engineering Capability Maturity Model [SEI 2003].

The SSE-CMM Process Dimension

The SSE-CMM defines the practices required to do correct software security engineering. Those practices comprise a complete set of activities over the entire life cycle of a trusted product or secure system [SEI 2003]. Each of these activities has a goal statement that specifies its unique purpose and the precise tasks to be performed to produce a correct work product. The eleven general process areas related to secure software production are

1. Administer Security Controls
2. Assess Impact
3. Assess Security Risk
4. Assess Threat
5. Assess Vulnerability
6. Build Assurance Argument
7. Coordinate Security
8. Monitor Security Posture
9. Provide Security Input
10. Specify Security Needs
11. Verify and Validate Security

The SSE-CMM also includes a set of processes related to project and program management. These eleven process areas are

1. Ensure Quality
2. Manage Configuration
3. Manage Project Risk
4. Monitor and Control Technical Effort
5. Plan Technical Effort
6. Define Organization’s Systems Engineering Process
7. Improve Organization’s Systems Engineering Process
8. Manage Product Line Evolution
9. Manage Systems Engineering Support Environment
10. Provide Ongoing Skills and Knowledge
11. Coordinate with Suppliers

The SSE-CMM Capability Dimension

The second dimension is the capability dimension. This dimension characterizes the level of capability maturity of each of the processes presented in the last section. Capability is assessed using a standard set of management attributes, which are applicable to any process. In effect, these common attributes characterize the generic management capability of any assessed process. Improving the security engineering process would then involve increasing the generic level of management practice (below), while ensuring that all the requisite base practices discussed in the last section are performed.

The SSE-CMM capability dimension is structured in the same manner as the better-known Software CMM (SW-CMM). The five maturity levels in the SSE-CMM indicate increasing capability, and each level is composed of several base practices. The difference is that the SSSE-CMM targets security engineering concerns rather than conventional software development. (All of the following in italics is from [Paulk 1993].)

Capability Level 0 - Individual heroism

Base practices do not exist

Capability Level 1 Performed Informally - Base Practices Are Performed Informally

Base practices are generally performed. The performance of these base practices may not be rigorously planned and tracked. Performance depends on individual knowledge and effort. However, work products of the process testify to their performance. Individuals within the organization recognize that an action should be performed, and there is general agreement that this action is performed. There are identifiable work products for the process.

Capability Level 2 Planned and Tracked - Planning Performance, Disciplined Performance, Verifying Performance, and Tracking Performance

Performance of the base practices is planned and tracked. Performance according to specified procedures is verified. Work products conform to specified standards and requirements. The primary distinction between the Planned and Tracked Level and the Performed Informally Level is that the performance of the process is planned and managed and progresses toward a well-defined process.

Capability Level 3 Well-Defined - Defining a Standard Process, Performing the Defined Process, Coordinating the Process

At this level, base practices are performed according to a well-defined process using approved, tailored versions of standard, documented processes. The primary distinction from the Planned and Tracked Level is that the practices of the Well-Defined Level are planned and managed within an organization-wide standard process.

Capability Level 4 Quantitatively Controlled - Establishing Measurable Quality Goals and Objectively Managing Performance

Detailed measures of performance are collected and analyzed at this level. This leads to a quantitative understanding of process capability and an improved ability to predict performance. Performance is objectively managed. The quality of work products is quantitatively known. The primary distinction from the Well-Defined Level is that the defined process is quantitatively understood and controlled.

Capability Level 5 Continuously Refined - Improving Organizational Capability

Quantitative process effectiveness and the efficiency goals (targets) for performance are established, based on the business goals of the organization. Continuous process improvement against these goals is enabled by quantitative feedback from performing the defined processes and from piloting innovative ideas and technologies. The primary distinction from the Quantitatively Controlled Level is that the defined process and the standard process undergo continuous refinement and improvement, based on a quantitative understanding of the impact of changes to these processes.

Each of the practices within the process framework can be assessed at one of these five levels of capability. The external audit develops evidence that a particular process purpose has been achieved. In order to attest to an increase in capability, the activities at each new level must represent proof that the requisite management attributes are present.

Process capability attributes are grouped by capability levels. The intention is to provide a rational way to both characterize current process capability and to suggest a set of specific steps that can be taken to improve it. The actual security engineering capability of a given process is characterized by performance of an increasingly more mature set of activities, which serve as commonly accepted proxies for enhanced capability. Presence or absence of those “standard” activities can then be used to authenticate the process’s degree of reliable execution.

ENSURING SECURE TRUST RELATIONSHIPS

From an acquirer-supplier standpoint, the ability to evaluate the security engineering capability of each process is at the heart of establishing a trusted relationship. That is because it can be assumed that if the supplier organization is operating at a specific level of capability, then it is essentially trustworthy.

In evaluating all bids, it is important to know the specific risks involved in dealing with any given supplier. Since defects are tied to capability, the ability to characterize risk requires an in-depth understanding of how adequately each bidder’s security engineering processes match up with the project’s software assurance requirements. Those goals are expressed through a target security engineering capability statement that is issued as part of the RFP.

This statement is similar to the protection profiles used by the Common Criteria. It characterizes the entire range of security processes the vendor must execute and their requisite level of capability maturity. The statement lists the processes deemed essential to meeting the requirements of the bid and states the level of capability required. For instance, a simple capability profile using the highest level SSE-CMM categories might look like the one in Figure 1. It must be kept in mind when viewing this example that these high-level process areas are characterized by constituent base practices, and it is at that base practice level that the actual capability profile would be itemized.

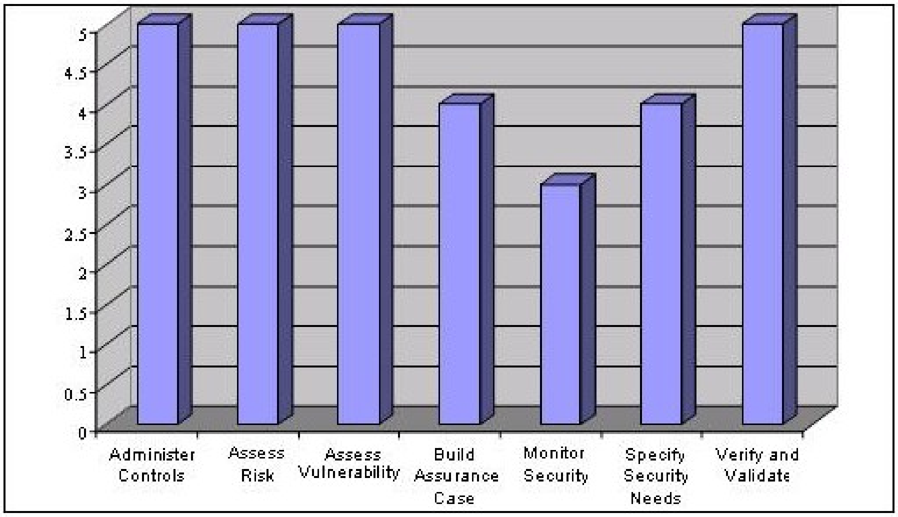

Figure 1. Example of required levels of maturity for security engineering practices

The profile designates the level of adequacy required for each security process. Although these targets require judgment, the actual assessment is always objective and audit based. Each process's current capability is ranked against the target attributes and an explicit process adequacy rating is obtained for each vendor, as shown in the example in Figure 2.

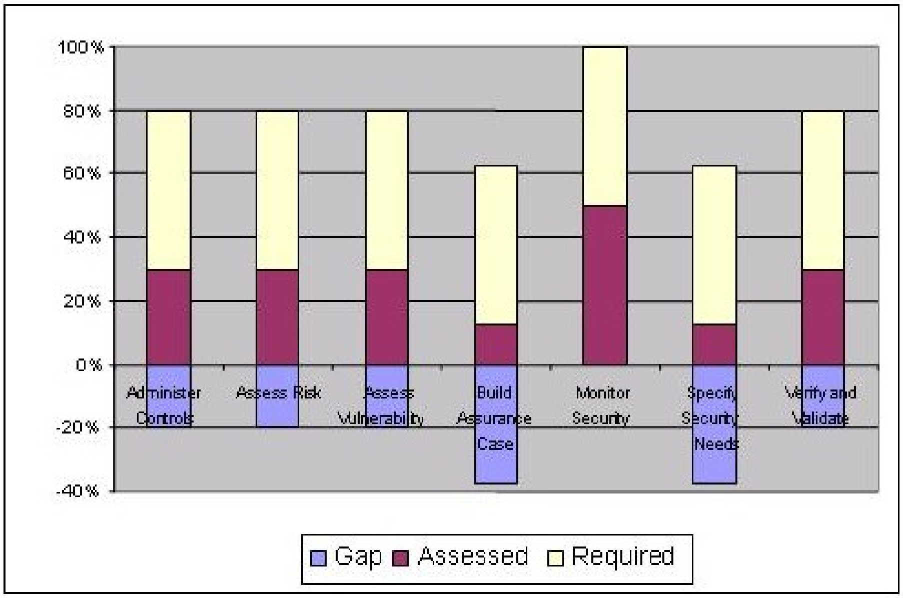

Figure 2. Comparison of vendor maturity levels to required security levels

The relationship between the required level of process capability and the actual level of capability is summarized in a process capability report. That report itemizes the strengths and weaknesses of each supplier's processes against the target requirements. The outcome of the comparison should include identified, strengths, weaknesses, and risks inherent in each assessed process.

Risk is then expressed in terms of the gaps and the inherent consequences associated with each gap. The report should help the customer decide whether a given vendor's processes will meet project requirements. It should also help identify areas where security problems might crop up or where defects might be injected. That information can be used to dictate the form of any subsequent protections built into the contract.

SOME GENERAL CONCLUSIONS

The concept of a standard process to assess and certify vendor competency is introduced here. The advantages of such an assessment process should be clear. First, a standardized assessment can characterize organizational security capability for internal management consumption. More importantly however, any subsequent external audits would provide objective evidence that could be used to certify the general capability of the organizations security engineering practices. The ability to maintain a centralized registration of the level of security capability can then serve as a basis for accrediting trusted vendors worldwide.

The advantage of a reliable, common assessment approach would be that the customer could then make a "buy" decision based on precise knowledge about the trustworthiness of any number of competing suppliers, whether the customer actually has knowledge of those suppliers or not. And equally important, each of the suppliers can get a better fix on what is required to satisfy the terms of the contract. That will allow them to make decisions about their own capability as well as what they have to do to improve it. Assuming the supplier is ethical, this will ensure that they know that they can meet the goals of the acquisition prior to attempting to bid.

Also, given the trend toward multitiered outsourced arrangements in software development and services, there is another reason for doing capability certification. Using this method, it would be possible to create a hierarchy of profiles that specify both the process capability requirements of prime contractors as well as any subcontractors. In that respect then, it is much easier to ensure that the supply chain that underwrites complex system development does not have any weak links.

A serious process capability determination that involves a number of competing suppliers has to employ a common assessment basis to ensure consistent verification of each supplier’s capability. The ideal situation would be a single umbrella assessment that would underwrite trust among all customer and supplier organizations. Independent auditing of individual capability could then ensure that any company that wishes to acquire a software product can find a trustworthy supplier anywhere in the world. The key to that effort is the availability of third-party certification of vendor capabilities.

The main problem is that the international body typically responsible for global certification, the International Standards Organization (ISO), has not jumped into the fray yet. So there is currently no universally recognized registrar where certifications of process capability can be kept. Therefore, any audited certifications of the type we are discussing here would have to be between individual acquirers and suppliers. Obviously, that informal arrangement does not work as well as having a single repository of capability certificates. Nevertheless, if agreement can be obtained among the registration authorities, it should be much easier for CIOs to sleep at night because it will be possible to make much more informed decisions about whom they are dealing with.

Assessment is not a cure-all. But as secure software requirements become more pressing and the software business becomes more global, this approach can provide a realistic, viable, and attractive basis for ensuring that outsourced arrangements do not lead to security disasters. Given all that has been said earlier, businesses require a utilitarian road map to support their decision making. The two-dimensional model we have discussed in this article creates such a detailed and consistent process framework.

BIBLIOGRAPHY

[Borland 2005]

Borland Software Corporation. “Software Delivery Optimization: Maximizing the Business Value of Software.” Borland Vision and Strategy Solution Whitepaper, 2005.

[Cisco 2003]

“Defense Agencies Meet Readiness Challenges with Commercial off the Shelf (COTS)-Based Systems,” Cisco, 2003.

[Construx]

Construx Software Builders, http://www.construx.com

[DSB 2007]

Defense Science Board. Mission Impact of Foreign Influence on DoD Software. Office of the Undersecretary of Defense for Acquisition, Technology and Logistics, Washington, D.C., September 2007.

[Humphrey 1987]

Humphrey, Watts S. Characterizing the Software Process: A Maturity Framework (CMU/SEI-87-TR-11). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1987.

[Humphrey 1994]

Humphrey, Watts S. Managing the Software Process. Reading, MA: Addison-Wesley, 1994.

[Jackson 2007]

Jackson, William. “Under Attack: Common Criteria Has Loads of Critics, but Is It Getting a Bum Rap?” Government Computer News, 8/13/07.

[Jones 1994]

Jones, Capers. Assessment and Control of Software Risks, Englewood Cliffs, NJ: Prentice-Hall, 1994.

[Jones 2005]

Jones, Capers. “Software Quality in 2005: A Survey of the State of the Art.” Software Productivity Research, Marlborough, MA, 2005.

[McConnell 2007]

McConnell, Steve. “The Business Case for Software Development.” Construx Software Builders Inc., 2007.

[McGibbon 1999]

McGibbon, Thomas. “A Business Case for Software Process Improvement Revised.” DoD Data Analysis Center for Software (DACS), 1999.

[Paulk 1993]

Paulk, M., Curtis, B., Chrissis, M., & Weber, C. Capability Maturity Model, Version 1.1. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1993.

[Redwine 2006]

Redwine, Samuel T. (ed.). Software Assurance: A Guide to the Common Body of Knowledge to Produce, Acquire and Sustain Secure Software, Version 1.1. U.S.

[SA-CMM]

Cooper, Jack and Fisher, Matt (eds.). Software Acquisition Capability Maturity Model (SA-CMM), Version 1.03 (CMU/SEI-2002-TR-010). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2002.

[SEI 2003]

Software Engineering Institute. Systems Security Engineering Capability Maturity Model SSE-CMM, Version 3.0. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2003.

Copyright © Carnegie Mellon University 2005-2012.

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.

This material has been approved for public release and unlimited distribution except as restricted below.

Internal use:* Permission to reproduce this material and to prepare derivative works from this material for internal use is granted, provided the copyright and “No Warranty” statements are included with all reproductions and derivative works.

External use:* This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other external and/or commercial use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

* These restrictions do not apply to U.S. government entities.

DM-0001120