Effective Acquirer/Supplier Software Document Reviews

Gregory T. Daich
Software Technology Support Center

With all the quality initiatives of the last several decades, the software industry still continues to produce too many poor quality systems that are over budget and very late. Part of this problem can be attributed to software acquirers not understanding what they want and what they get at each stage of development. Improving the effectiveness of joint acquirer/supplier software document reviews can result in significantly less rework in later development phases. Project plans, requirements, and test plans are particularly important documents that acquirers should review using a more rigorous process than is practiced by many acquisition organizations.

What is in a Name?

Not all reviews are alike. What some people call a review (or a technical review), others call a free-for-all, come as you are, anyway you like it, gab session. Another name for some of these types of meetings is an informal design brainstorming session.

What some call a software inspection (or a peer review), others call a practice session for the latest speed-reading contest. What some call a formal review (or structured walk-through), others call a waste of time. Often valuable process and product information gets ignored or thrown away before it can be used to improve practices and show value for the effort expended.

Finally, what some call an audit (or software quality assurance review), others call a half-baked effort. Many reviews are severely handicapped due to missing or unavailable review resources or inadequate time.

Review processes can be called by a lot of different names. If your goal is to determine that a software work product is ready for the next phase of development or delivery, then you need review practices that:

- identify problems first before entertaining unsolicited corrections and determine ahead of time who will provide and review corrections and when those efforts will occur
- optimize the time spent by each participant through planning and focusing on what is important, thus helping them to be as effective and efficient as possible
- determine the benefits of each review and make use of critical information that can improve upstream processes and the review process itself
- bring all the key resources to bear on the effort such as subject matter experts; all references (source documents); and standards, rules, or checklists (telling what the organization has agreed a document should contain)
- provide the defined processes for planning and conducting reviews

Call your review process by whatever name you choose but insist that you have the above basic review practices in place in your organization. These concepts have significantly improved productivity and quality in numerous development and acquisition organizations [1].

Many managers incorrectly assume that their staff knows how to effectively and efficiently review technical documentation. With proper training many people have dramatically increased their skills to find and report defects. These people have learned where to look, how much time to spend, and what information to collect to optimize their review efforts.

What is the Problem Here?

The ultimate goal of most software product reviews is to determine product quality in order to verify that the product is ready for the next stage of development. However, acquisition organizations often do not have all the information they need to determine that readiness.

Reviews conducted by acquisition organizations generally fall into three main categories: project status management reviews, software product technical reviews, and product and process audits. These reviews depend on developer (supplier) participation to provide most of the inputs for these reviews; however, supplier participation during these reviews often leaves something to be desired. Suppliers do not want to make themselves look bad by finding problems that should have already been corrected. Thus their participation is self-relegated to simply answering questions from acquirers as needed. In other words, suppliers often do not fully participate as reviewers in joint acquirer/supplier reviews.

Developer reviews fall into the same basic categories as acquirer reviews (i.e. management reviews, technical reviews, and audits). Developer reviews should generally be more frequent and should look deeper into the software products than acquirer reviews. Unfortunately, many developer reviews are conducted haphazardly, if at all, with no data collected regarding critical defects. Thus, the potential to learn from past problems is lessened considerably. Also, a return on investment for the effort expended cannot usually be determined so no one can state objectively how well reviews are supporting development.

Acquirer reviews tend to involve higher volumes of material than developer reviews because it costs more to get suppliers and acquirers together. The result is that acquirers, with very little review support from developers and almost no useful information about prior conducted reviews, skim over the document. Acquirers simply do not have enough time to look deeply into the software products. Acquirer's fears often come to pass in later stages of development when too many defects are found and schedules slip due to under-planned debugging, fixing, and

retesting. Would it not have been nice if only most of those defects could have been detected earlier or maybe even prevented?

What do the Standards Say?

Four key standards that address document reviews are worth considering by the Department of Defense (DoD) and commercial acquisition organizations. They are Mil-Std-1521B, EIA/IEEE J-Std-016-1995, IEEE/EIA 12207.0-1996, and IEEE Std 1028-1997 [2, 3, 4, 5]. The following subsections discuss these standards in order of their creation.

Mil-Std-1521B

Mil-Std-1521B is a military standard that has been cancelled for DoD use mostly because the government wanted to get out of the software standards business and use commercial standards [4]. This standard has been approved for public release but no further updates are expected. It lists a number of technical reviews and audits, some of which have become known as overkill for some projects. These include:

- system requirements review (SRR)
- system design review (SDR)
- software specification review (SSR)
- preliminary design review (PDR)
- critical design review (CDR)
- test readiness review (TRR)
- functional configuration audit (FCA)
- physical configuration audit (PCA)
- formal qualification review (FQR)
- production readiness review (PRR)

Software acquirers conduct these formal reviews and audits. However, Mil-Std-1521B does not say how to conduct them. These reviews often result in numerous comments and corrections to documents, many of which contain completely opposite views on the same text.

On the developer side, many contractors have spent a considerable amount of time preparing briefing slides for a Mil-Std-1521B technical review. This time could have been spent more productively doing internal peer-type of document reviews prior to the technical review. It has been apparent that some less progressive organizations have not conducted internal peer reviews with some projects I have been involved with. It appears that some acquirers have the privilege of being the first to view some deliverables.

The Software Engineering Institute's (SEI) Software Capability Maturity Model (SW-CMM^{SM}) advocates conducting peer reviews [6]. The SW-CMM has a Key Process Area (KPA) that provides some guidance in conducting peer reviews. Software acquisition organizations also could benefit by adopting similar practices when participating in reviews.

The SEI has published another interesting Capability Maturity Model focused on software acquisition organizations (SA-CMM^{SM}). However, there is not a specific KPA dedicated to reviews such as there is in the SW-CMM [7]. Note that every one of the SA-CMM KPAs depend on reviews of various types.

I am convinced some contractors could have avoided some embarrassment and subsequent rework had they focused more on finding defects using an effective review process. The Mil-Std-1521B technical review would have been more successful and the quality of the software products would have been better at delivery.

EIA/IEEE J-Std-016-1995

EIA/IEEE J-Std-016-1995 is an important, relatively new standard that many government and commercial organizations have adopted which was derived from Mil-Std-498 [8]. Many organizations are familiar with Mil-Std-498 since it was derived from other standards they have used to develop several government systems.

J-Std-016 states, "The activities and tasks in the standard tell what to do, not how to do it" [2]. According to the dictionary, "how" information is the manner or way in which we may do something whereas "what" is the "something" we are talking about [5]. Some standards focus on "what to do" in the interest of not constraining the user. Also, the sheer volume of some of these standards make it impossible to delve into "how to do it." My position is that if an activity produces superior results and has general applicability, it should be standardized whether it is "what" or "how" guidance. We will discuss later the IEEE Std 1028, which provides some "how to do it" review process information. Additional review activities that have evolved through experience are recommended but are not covered in any commercial standard at this point.

J-Std-016 provides "uniform requirements for acquiring, developing, modifying, and documenting software." The only part of the standard that cannot be tailored and potentially removed for a given project are the tailoring requirements. Product evaluations are required for all software products built to satisfy a contract (i.e. an agreement between an acquirer and a developer). The standard was careful not to mention "review" in the discussion about software product evaluations, since analysis and testing are two other ways to evaluate a product.

The standard provides a list of criteria against which each type of software product can be evaluated. This is one of the best kept secrets of this standard, it seems, since some organizations are not using these criteria. Reviews tend to be the best method to verify that many of the listed criteria have been met. These criteria serve as a very useful starting point when defining review checklists.

While reviews could potentially be tailored out of the standard (and the organization's process) for a given project because they were deemed inappropriate, that would be like shooting yourself in the foot. Why would you want to inflict such a handicap on a project? Surely at least one documented, reviewable product will be produced in every development effort that will need to be evaluated (e.g. development plan or final software product). Since every project will produce some product that will require evaluation, we infer that product evaluations (reviews of work products) cannot rationally be tailored out.

J-Std-016 briefly discusses joint management reviews, which are similar to the Mil-Std-1521B technical reviews. J-Std-016 software product descriptions (templates) also provide some useful evaluation criteria. You should consider the applicability

of each section of a template for each project. This will help avoid the problem that I have termed the "factory approach to writing documents." Often, people will take these templates and fill in the blanks with something just to complete it. The objectives of the project are not carefully considered as the document is written. Another problem is people taking a document from a prior project and replacing old parameters with new data. The real objectives of the project can easily be incorrectly biased toward the previous project.

IEEE/EIA 12207.0-1996

IEEE/EIA 12207.0-1996 is a new standard that brings it all together. Not only is the development process covered but so are other primary life cycle processes, including acquisition, supply, operation, and maintenance. Also, supplemental guides provide additional implementation information and example life cycle data (document content and references to document templates in dozens of other useful standards and guides) [9, 10].

Again, this standard does not provide "how to" information but references IEEE Std 1028-1997 for that information. However, there are useful lists of criteria provided for many types of documents.

IEEE/EIA 12207.0-1996 addresses supporting life cycle processes including:
- documentation
- configuration management
- quality assurance
- verification
- validation
- joint review
- audit
- problem resolution

A joint review between an acquirer and a supplier is required for many of the activities in the development process and other primary and supporting life cycle processes. Basically, all deliverable documents that could have quality problems should be considered for joint review. The verification supporting process mentions reviews as one of the verification activities, with analysis and test as other options. Verification tasks that could benefit from various types of reviews include:
- contract verification
- process verification
- requirements verification
- design verification
- code verification
- integration verification
- documentation verification

The next section discusses the IEEE Std 1028, IEEE Standard for software reviews, that provides some "how to" information that development and acquisition organizations should consider.

IEEE Std-1028-1997

This standard defines systematic review practices applicable to

Type of Review	Review Purpose
Management Reviews	"The purpose of a management review is to monitor progress, determine the status of plans and schedules, confirm requirements and their system allocation, or evaluate the effectiveness of management approaches used to achieve fitness for purpose. Management reviews support decisions about corrective actions, changes in the allocation of resources, or changes to the scope of the project. Management reviews are carried out by, or on behalf of, the management personnel having direct responsibility for the system. Management reviews identify consistency with and deviations from plans, or adequacies and inadequacies of management procedures. This examination may require more than one meeting. The examination need not address all aspects of the product."
Technical Reviews	"The purpose of a technical review is to evaluate a software product by a team of qualified personnel to determine its suitability for its intended use and identify discrepancies from specifications and standards. It provides management with evidence to confirm whether: a) The software product conforms to its specifications b) The software product adheres to regulations, standards, guidelines, specifications, plans, and procedures applicable to the project c) Changes to the software product are properly implemented and affect only those system areas identified by the change specification"
Inspections	"The purpose of an inspection is to detect and identify software product anomalies. This is a systematic peer examination that: a) Verifies that the software product satisfies its specifications b) Verifies that the software product satisfies specified quality attributes c) Verifies that the software product conforms to applicable regulations, standards, guidelines, specifications, plans, and procedures d) Identifies deviations from standards and specifications e) Collects software engineering data (for example, anomaly and effort data) (optional) f) Uses the collected software engineering data to improve the inspection process itself and its supporting documentation (for example, checklists) (optional)"
Walk-Throughs	"The purpose of a systematic walk-through is to evaluate a software product. A walk-through may be held for the purpose of educating an audience regarding a software product. The major objectives are to: a) Find anomalies b) Improve the software product c) Consider alternative implementations d) Evaluate conformance to standards and specifications Other important objectives of the walk-through include exchange of techniques and style variations and training of the participants. A walk-through may point out several deficiencies (for example, efficiency and readability problems in the software product, modularity problems in design or code, or untestable specifications)."
Audits	"The purpose of a software audit is to provide an independent evaluation of conformance of software products and processes to applicable regulations, standards, guidelines, specifications, plans and procedures."

Table 1. *Purpose of reviews according to IEEE 1028-1997.*

acquisition, supply, development, operation, and maintenance processes. It states, "This standard describes how to carry out a review. Other standards or local management define the context

within which a review is performed, and the use made of the results of the review" [3]. It also says that it provides minimum acceptable requirements for systematic software reviews where systematic implies the following attributes:

   a) team participation
   b) documented results of the review
   c) documented procedures for conducting the review

IEEE Std 1028 defines the basic processes for the following types of reviews:

   • management reviews
   • technical reviews
   • inspections
   • walk-throughs
   • audits

Table 1 defines the purpose for each type of review. Many organizations will see common activities between their types of reviews and the reviews defined in this standard. However, there may be some significant activities in this standard that an organization has not adopted into its process that it should consider. If a review activity could result in finding more defects, if it could help an organization learn from past mistakes, or if it could better estimate the value of the efforts expended on reviews, then these activities may prove extremely beneficial.

Management reviews and audits have some similarities with the other types of reviews but also have unique characteristics. For example, a management review is the only type of review where financial status is examined and that is optional. Information from all types of reviews can be input to management reviews. However, document defects or anomalies are often found and recommendations have often been prepared by the time a management review is conducted.

Audits specifically review software processes in addition to software products. Some process information is gathered through interviews which are unique to audits. Also, the reports and the feedback are much different for audits than for the other reviews.

There are surprisingly only a few major distinguishing characteristics between technical reviews, inspections, and walk-throughs. Technical reviews basically are the same type of review as an inspection. However, inspections emphasize software product revisions as part of the review process, whereas technical reviews can conclude with a list of anomalies and recommendations. Technical reviews involve more people than just peers, such as management and acquirers. Inspections typically involve only peers, but this article addresses how inspections can involve acquirers and suppliers in joint reviews. The most significant difference between inspections and technical reviews is that inspections are more in-depth and usually cover a lower volume of materials.

Walk-throughs appear to be much less thorough than an inspection. More review materials are brought to bear with inspections than walk-throughs. However, more time is spent in walk-throughs discussing and investigating alternative solutions. This could be more of a problem than a help. If the walk-through diverges to mostly discussions about better design alternatives, then less time will be spent in finding and understanding problems before solutions are recommended. Every solution discussion should consider whether all reviewers are needed for that discussion.

Should an organization plan to conduct all of these types of reviews? Tom Gilb, in his book on software inspections, says that walk-throughs are for training [1]. He also says reviews are for gaining consensus, but the review Gilb is talking about is principally the IEEE Std 1028 management review does. Finally, he recommends that to find defects and determine document quality, inspections are the best method.

IEEE Std 1028 and this article treat reviews as a generic term that encompasses any type of document or process examination and evaluation. This standard stresses the importance of determining the objectives of each type of review by requiring this to be written as an entry criterion prior to conducting a review. There are several additional activities that Gilb and others advocate on top of this standard that are well-known in many practicing organizations. Some of the "how to do" information missing in IEEE Std 1028 include:

   • monitoring inspection rates (e.g. pages reviewed per hour)
   • providing useful reference citations (include page and section)
   • sampling for inspections
   • using entry criteria (e.g. number of defects found) to determine if a review meeting should be held
   • using developer inspection data to give acquirers additional insight into software product quality
   • using inspection practices as a mechanism for conducting joint technical reviews between acquirer and developer

These practices can significantly increase acquirer software document review effectiveness and efficiency. The next section outlines a process that a few acquisition organizations have used to help better determine software product quality and readiness for the next stage of development.

Note that the IEEE standards can be obtained by calling 1-800-678-IEEE.

What is an Acquirer/Supplier Software Document Review?

There are several types of acquisition-related software document reviews:

   (1) acquirer document review
   (2) contractor document review
   (3) joint document review
   (4) joint management review

Acquirer document reviews are reviews of documents that acquisition organizations write. These documents include policies, contracts, Statements of Work (SOWs), and various plans. In contrast, contractor document reviews are internal reviews of documents that may or may not be delivered to the acquirer. The software products that are delivered to the acquirer are of

particular interest because the acquirer usually must review and approve them. These documents include software development plans, requirements documents, design specifications, test plans, and user manuals.

The following two subsections discuss typical problems with acquirer document reviews and joint document reviews.

Acquirer Document Review

Acquirers and developers have achieved increased understanding of project deficiencies with the use of document inspections as described in the IEEE Std 1028-1997. Unfortunately, I have seen only a few acquisition organizations employ these types of inspections when reviewing software work products delivered by a contractor.

The techniques for reviewing acquirer-written documents often result in skimmed-over reviews, with individual reviewers obtaining little or no guidance on what to look for.

The underlying assumption is that you read, you find defects, you report them. What more is there to a review? If you ask for some clear objectives for a review, you can almost hear some people say, "You have been in this business how long? You should know how to review." With little or no direction, here is an example of what can happen:

Ten people are asked to review a SOW. Two hundred issues get reported, with many issues completely opposite from others. Some reviewers choose to rewrite portions of the SOW, while others write several paragraphs of text explaining why they think something is wrong.

These review practices cost extra project time and do not often result in finding and fixing enough serious problems.

Acquisition organizations conducting inspections of documents they produce can have similar effects to what some developers have experienced. These effects include high quality documents, increased understanding of document content, and reduced amount of downstream rework for themselves and their contractors.

Joint Document Review

Sometimes acquirers believe they do not have the time to fully participate in a joint document review (i.e. technical review of deliverable software product) so they watch from behind the scenes. When the developer signs off that all evaluation criteria have been met, some acquirers accept it. This often results in acquirers feeling uncomfortable about the product not really being ready for the next stage of development. Acquirers do not have the objective quality information they need to assess readiness. Their fears are often confirmed when the system reaches later life cycle stages and too many defects start surfacing. For example, many products have reached operational test and evaluation (OT&E) only to fail miserably at that stage. You wonder how they made it out of the developer level of testing. Yet, the contractor was able to obtain all the approvals needed to deliver the system to OT&E.

Most acquisition organizations in the DoD and many commercial organizations have experienced some downsizing, which often results in increased workloads and less time to manage and track the efforts of their contractors. DoD acquisition reform transfers more management and oversight responsibilities from acquisition organizations to contractors. But this does not lessen the need for acquirers to understand what they are buying and to approve what is delivered. More than ever, practical techniques and mechanisms are needed to gain improved insight into the quality of delivered software products in a shorter time than once permitted.

Sometimes, acquirers do not fully participate in document reviews because they lack specific knowledge. Can an acquirer always participate effectively in a technical review of a software product? Some acquirers knowingly or unknowingly focus on crossing "T's" and dotting "I's". What constitutes a significant issue? The next two sections elaborate on these review problems.

Subject Matter Experts

Many people assume that subject matter experts (SMEs) will naturally be effective in reviewing a document related to their expertise. In other words, SMEs do not need to follow a process for reviewing a document, they will see all the defects and we will all be saved. However, humans have a difficult time keeping more than five to nine concepts in their short-term memory at one time. How can you expect a SME to effectively review a document without looking at its references (sources, regulations, standards, guidelines, plans, and procedures)? Yet this is how many people review a document — looking only at the document and not at any of its references.

If a document under review is dependent on several references, those references should be checked to be sure that the document is correct and consistent with them. If a reviewer (SME or not) does not take the time to check the references, he or she is missing an opportunity to find serious defects early which can be corrected before others encounter them.

What if a specific SME is not available for a planned review? This is a main reason to use standards or rules for writing documents. We want to decide ahead of time what the doc-

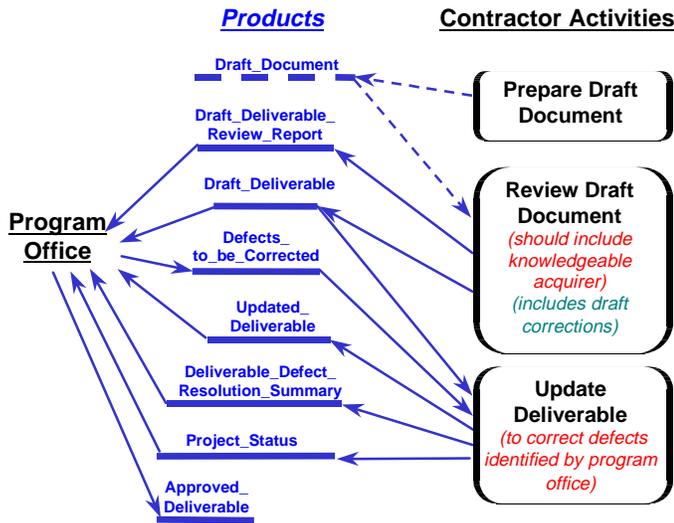Figure 1. *Joint document reviews — program office side.*

Products   Contractor Activities
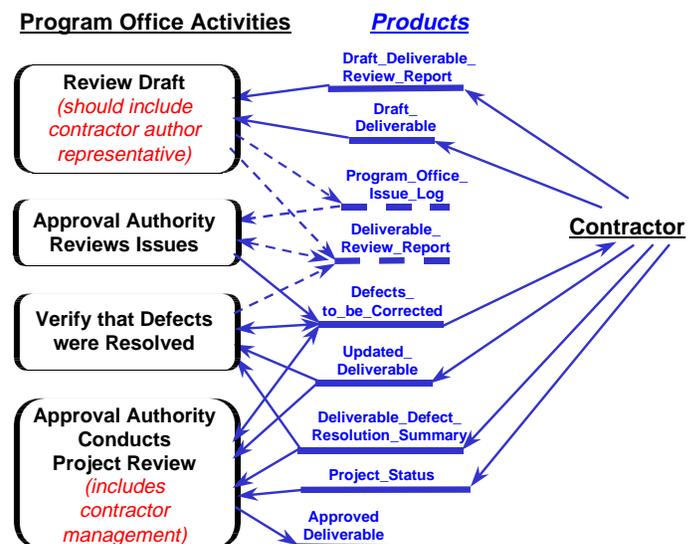
Figure 2. *Joint document reviews — contractor side.*

uments should contain. We can glean a lot of information from SMEs by encoding their knowledge, so to speak, in document rules or standards and checklists so we are not so totally dependent on the SME.

I have reviewed hundreds of technical documents in the last few years and found many defects. Correcting these defects has made a significant difference in improving document quality and project productivity. However, I usually am not a project SME for the documents I review because I support many projects in a consulting role. Often, SMEs already have reviewed these documents but missed several serious defects because they were not looking for certain critical types of issues. Sometimes they do not understand the types of problems that these kinds of defects can cause.

For example, I have seen test plans with no real test planning information and others that entirely missed the boat on what should be contained in key sections of the test plan. I have seen a software development plan that was essentially an organizational process and did not contain any specific information about the system to be built, the schedule, nor any project-specific risks.

One of the most insidious of all problems is a document with no clear statement of objectives for the project and the document. I have seen this problem in many types of documents. A document with no clearly defined objectives has a difficult time gaining buy-in from its readers or users because no one is sure what the document should do. Some process improvement-related documents (action plans or guides) often do not contain adequate citations of references, making their credibility questionable.

An author is, by definition, an SME and we would not think of conducting a review without them. But other SMEs may not be available who could contribute to the review. Getting them involved early when establishing the objectives and general direction for a project can help alleviate some schedule problems when you might have a little more flexibility with the schedule. This will also help the project start effectively by getting early agreement from key personnel that you are

headed in the right direction.

Finally, with a good set of rules (or standards) and checklists and an effective process, acquirers can be very productive in identifying document deficiencies in contractor-developed documents. This is true even if some acquirers do not yet know a great deal about the system. They must learn about it and they must signoff that a document is ready for the next stage of development.

The section below, What is the Recommendation?, outlines an effective acquirer/supplier software document review (joint document review) process. This process has helped acquirers find serious problems undetected by the developer and it has helped find more problems earlier than with traditional technical review practices.

Major vs. Minor Defects

A defect is considered serious or major if we estimate it could (not would) take more than an order-of-magnitude to fix later and correct its resulting consequences vs. fixing it now. Furthermore, fixing an actual software defect later often introduces more problems. Regression testing is done to assure that no new defects have been introduced in the unchanged parts of a system. This time should also be estimated when considering whether a document defect is major or minor. This perspective augments the definition of a major defect (anomaly) provided in IEEE Std 1028. This standard says that a major anomaly is one, "that would result in failure of the software product or an observable departure from specification."

A minor defect will not likely require more time to fix later vs. now. IEEE Std 1028 says that a minor anomaly causes the software product to "deviate from relevant specifications but will not cause failure of the software product or an observable departure in performance."

Table 2. *Joint document reviews — program office activities.*

Activity	Description
Review Draft	This is an IEEE Std 1028 inspection type of review of the Draft_Deliverable that should include the contractor author or author representative. The Draft_Deliverable_Review_Report provides supplemental information to help limit to a representative sample of document chunks how much of the Draft_Deliverable that should be jointly reviewed.
Approval Authority Reviews Issues	After the review team identifies significant issues, program office management identifies and/or approves Defects_to_be_Corrected from the Program_Office_Issue_Log that must be corrected prior to approval of the Draft_Deliverable.
Verify that Defects were Resolved	The review team verifies that all Defects_to_be_ Corrected have been corrected in the Updated_ Deliverable. This activity may need to occur repeatedly until the Draft_Document appears ready for approval.
Approval Authority Conducts Project Review	This joint management review between the program office and the contractor reviews the Deliverable_ Defect_Resolution_Summary along with the Updated_Deliverable to determine if the Updated_ Deliverable is ready for approval. Also, Project_Status may be reviewed as needed to monitor project status against the plans and schedules.

Activity	Description
Prepare Draft Document	The documents of particular interest that are prepared in this internal contractor activity are those that require approval by the program office for a particular development stage or for delivery. These include plans, requirements documents, designs specifications, source code, user manuals, etc.
Review Draft Document	This is an IEEE Std 1028 inspection type of review that shall include the author or author representative. The Draft_Document_Review_Report should be prepared and should be available for the program office to review. This activity should include a knowledgeable acquirer. Note that identified defects are corrected prior to delivery to the program office.
Update Deliverable	The contractor corrects the Draft_Deliverable to address all Defects_to_be_Corrected. This activity may need to occur repeatedly until the program office is satisfied and approves the Draft_Deliverable which then becomes the Approved_Deliverable.

Table 3. *Joint document reviews — contractor activities.*

What is the Recommendation?

This section describes an acquirer/supplier software document review (joint document review) process that can help acquisition organizations obtain useful software product quality information in order to make key decisions in a timely manner. This is an innovative systematic review process that implements powerful IEEE Std 1028 inspection technologies to provide critical document (and project) quality information. These practices can be used by acquisition organizations to knowledgeably approve documents and permit follow-on work to start. Figure 1 shows the major acquirer review activities and associated data. Figure 2 shows the major developer activities and associated data. The dashed lines represent internal activities and documents to either the program office or the contractor.

Table 2 discusses the acquirer document review activities. Table 3 discusses the developer document review activities. Table 4 discusses internal data to the program office, internal data to the contractor, and data transferred back and forth between the program office and the contractor.

Conclusion

As an industry, we need to look for ways to open up communication between acquirers and suppliers to willingly discuss problems and risks. The acquirer/supplier software document review (joint document review) process recommended in this article can do this by helping acquirers and developers gain better insight into document quality. Acquirers need more accurate quality information to determine readiness of developers to proceed to follow-on stages of development.

Both risk management and document inspections are recognized best practices by leading consultants in the industry. However, the traditional type of joint acquirer/supplier technical review does not qualify as a comprehensive, effective, and efficient IEEE Std 1028-type of document inspection. The traditional technical review is often a skim-over type of review that finds some problems but does not accurately assess a document's quality. The IEEE Std 1028-type of inspection permits reviewers to delve deeper in representative document samples to obtain better insight into document quality. This information, coupled with inspection data from internal developer inspections, permits acquirers to make more informed document approval decisions.

The joint document review discussed in this article provides a mechanism shown to be effective with several contractor and acquisition organizations we have worked with. As with all process improvement efforts, there were growing pains during

Table 4. *Joint document reviews — program office documents.*

Document	Description
Draft_Document	This is the contractor document that is ready to be reviewed (IEEE Std 1028 type of inspection) by the contractor. It is not yet ready for delivery.
Draft_Deliverable_ Review_Report	Since acquirers cannot attend all document reviews that contractors should perform, obtaining the statistics and defect information from the contractor document reviews could give useful insight into the contractor's review process. Too many issues or too few issues are both signs of a poor review process. The time spent and size of the document should be included in this report. This information can also help determine how much of the Draft_Deliverable to review.
Draft_Deliverable	This is the document that the contractor believes is ready for the next stage of development or for final delivery to the customer.
Program_Office_ Issue_Log	This is an internal log of issues found by the acquisition organization and the contractor authors or author representatives. This log is not given to the contractor. It is ordered by most significant issues to least so program office management can quickly understand the most serious issues.
Deliverable_ Review_Report	This is an internal report of the required review statistics. At a minimum, it includes the total amount of review time, number of major issues found, and number of pages reviewed. It is an internal document that's not given to the contractor. An estimate of the savings in downstream rework should also be included.
Defects_to_be_ Corrected	These are the most significant defects that must be corrected prior to approval of the deliverable. As a courtesy, it may include minor issues that can be cleaned-up as time permits.
Updated_ Deliverable	This document has been corrected by the contractor and should have addressed all Defects_to_be_Corrected. If it didn't, then the contractor will be asked to correct the document or the Defects_to_ be_Corrected document will need to be changed.
Deliverable_Defect _Resolution_ Summary	This document contains the status of each defect and summarizes the changes made to the Draft_Deliverable.
Project_Status	This information consists of normal project status information that may be required during the joint management review.
Approved_ Deliverable	This, of course, is the approved deliverable that now meets the acquirer's requirements as best as the acquirer and the contractor can tell at the time of approval. Some projects may need to move out and use a Draft_Deliverable prior to approval. This should be coordinated with the program office.

implementation. We initially had some questions about the level of contractor participation we would obtain. However, contractors, though initially reluctant, actively supported these joint document reviews. It was almost surprising how they willingly reported several significant issues. They wanted to actively participate for the benefit of all. I am convinced that these reviews even made a difference in how the contractor will conduct future internal document inspections.

"Mature" contractors want to find and fix problems to save downstream rework effort. "Mature" acquisition organizations want to approve documents that have been reviewed carefully with effective document review practices. This can ultimately result in acquirers being able to better handle their total workload.

If you would like more information about the joint document reviews (inspections) discussed in this article, please feel free to contact the author. A related article discussing demonstrating to your organization the power of these effective reviews was written for *CROSSTALK* in June 1999 [12]. ◆

About the Author

Gregory T. Daich is a senior software engineer with Science Applications International Corp. under contract with the Software Technology Support Center (STSC). He supports STSC's Software Quality and Test Group with more than 22 years experience in developing and testing software. He has taught more than 60 software test, document inspection, and process improvement seminars in the last five years.

Daich consults with government and commercial organizations on improving the effectiveness and efficiency of software quality practices. His consulting approach coordinates formal document inspections with analysis of test work products to identify opportunities for software test process improvement. These practices have also been applied in supporting and testing Year 2000 upgrades. He is the principal author for several guidebooks and workshops for conducting year 2000 compliance projects. These guidebooks address corporate- and project-level compliance efforts as well as year 2000 desktop (PC and Macintosh) software compliance.

Daich has a master's degree in computer science from the

University of Utah and a bachelor's degree in mathematics from Weber State University.

Software Technology Support Center
7278 Fourth Street
Hill AFB, Utah 84056-5205
Voice: 801-777-7172
Fax: 801-777-8069
E-mail: daichg@software.hill.af.mil
Internet: http://www.saic.com

References

1. Gilb, Tom, *Software Inspections*, Addison-Wesley, 1993.
2. EIA/IEEE J-Std-016-1995, "Standard for Information Technology — Software Life Cycle Processes — Software Documentation — Acquirer-Supplier Agreement."
3. IEEE Std 1028-1997, "Standard for Software Review Processes."
4. Mil-Std-1521B, "Technical Reviews and Audits for Systems, Equipments, and Computer Software," June 4, 1985.
5. IEEE/EIA 12207.0-1996, "Standard for Information Technology — Software Life Cycle Processes," March 1998 (industry implementation of International Standard ISO/IEC 12207 : 1995).
6. *Webster's II New Riverside University Dictionary*, The Riverside Publishing Company, 1984.
7. Paulk, Mark, et. al, *Capability Maturity Model for Software, Version 1.1*, CMU/SEI-93-TR-24, Carnegie Mellon University, Software Engineering Institute, February 1993.
8. Mil-Std-498, "Military Standard Software Development and Documentation," 5 December 1994.
9. IEEE/EIA 12207.1-1997, "IEEE/EIA Guide for Information Technology, Software Life Cycle Processes — Life Cycle Data," April 1998.
10. IEEE/EIA 12207.2-1997, "IEEE/EIA Guide for Information Technology, Software Life Cycle Processes — Implementation Considerations," April 1998.
11. Ferguson, Jack, et. al, , *Software Acquisition Capability Maturity Model, Version 1.0*, Carnegie Mellon University, Software Engineering Institute, December 1996.
12. Daich, Gregory T., "Preliminary Analysis Inspection Report (PAIR) …", Software Technology Support Center, *CROSSTALK*, June 1999.

The Data and Analysis Center for Software (DACS) Announces a New Technical Report

"Understanding and Improving Technology Transfer in Software Engineering"

This state-of-the-art report summarizes the history of software engineering technology transfer and suggests ways to help us understand how to shorten the time between innovation and effective practice.

It begins by examining earlier efforts to understand software-related technology transfer. Then we discuss the process of creating, evaluating, packaging, and diffusing technology. Next, this report considers each of these four activities in more detail, to determine how each contributes to the success of the overall transfer. Finally, areas that are ripe for further investigation are discussed.

This report may be viewed free on the Internet or downloaded for free in pdf form at: http://www.dacs.dtic.mil/techs/techtransfer2/

A bound, hard copy of this report is available for $50 and may be ordered from the DACS product order form at: http://www.dacs.dtic.mil/forms/orderform.shtml or by calling the DACS at (800) 214-7921.