



# DEPARTMENT OF DEFENSE IMPLEMENTATION GUIDE

PSM ADDENDUM



## DEPARTMENT OF DEFENSE IMPLEMENTATION GUIDE

*The PSM guidance is equally applicable to government and industry projects. This addendum provides specific information to help implement software measurement in the DoD acquisition environment. This environment is generally characterized by:*

- *formal two-party contractual agreements*
- *the use of integrated product teams*
- *acquisition policy driven management controls and performance measurement requirements*
- *the development and maintenance of relatively large and complex information, communications, and weapons systems*

*This addendum focuses on the DoD specific requirements related to the PSM tailoring process. It is comprised of 5 chapters:*

- ***Chapter 1, Measurement in the Acquisition Process*** - describes some of the effects of acquisition reform in the DoD environment.
- ***Chapter 2, Measurement for Integrated Product Teams*** - provides some insight into implementing measurement within the program IPT structure.
- ***Chapter 3, Putting Measurement on Contract*** - provides examples of contract wording to implement measurement on a contract. It describes the current acquisition and contract implementation guidelines that have proven to support identified measurement requirements.
- ***Chapter 4, Using Work Breakdown Structures*** - shows various Work Breakdown Structures (WBSs) and how to use them to assist in using measurement to facilitate management of the work being performed.
- ***Chapter 5, Integrating Measurement with Earned Value*** - describes the Earned Value program management methodology and how to implement it in conjunction with a measurement process.



## TABLE OF CONTENTS

<b>CHAPTER 1 - MEASUREMENT IN THE ACQUISITION PROCESS .....</b>	<b>539</b>
1.1 Pre-Acquisition Activities .....	539
1.2 Acquisition Life Cycle.....	540
1.3 Software Measurement in the Acquisition Process .....	542
<b>CHAPTER 2 - MEASUREMENT FOR INTEGRATED PRODUCT TEAMS .....</b>	<b>545</b>
<b>CHAPTER 3 - PUTTING MEASUREMENT ON CONTRACT .....</b>	<b>547</b>
3.1 Overview of Contracting Process.....	547
3.2 Contract Planning and Preparation .....	547
3.3 Proposal Evaluation.....	548
3.4 Negotiation .....	548
3.5 Contract Modifications .....	549
3.6 Sample RFP Wording.....	549
3.6.1 Requirements for Software Measures.....	549
3.6.2 Developer Access .....	550
3.6.3 Data Alternatives.....	551
3.6.4 Draft Measurement Plan.....	551
3.6.5 Proposal Evaluation Data .....	551
<b>CHAPTER 4 - USING WORK BREAKDOWN STRUCTURES.....</b>	<b>553</b>
<b>CHAPTER 5 - INTEGRATING MEASUREMENT WITH EARNED VALUE.....</b>	<b>559</b>



## CHAPTER 1 - MEASUREMENT IN THE ACQUISITION PROCESS

Acquisition reform has streamlined the Department of Defense and other government agencies' methods for systems acquisition, development, and sustaining engineering. The government has a fiduciary responsibility to ensure that funds are adequately managed. Measurement has become recognized as a cost-effective tool for the acquisition and technical management of software-intensive systems. This chapter explains how measurement and the Practical Software Measurement (PSM) process fits into the overall acquisition process.

### 1.1 Pre-Acquisition Activities

---

Software technology and mission needs evolve rapidly. As such, a better understanding of the capabilities of technology and user requirements must often be obtained before the formal acquisition process can begin. The DoD relies on two strategies to develop, demonstrate, and evaluate emerging technologies prior to the start of an acquisition. These include Advanced Technology Demonstration (ATD) and Advanced Concept Technology Demonstration (ACTD). Although these demonstrations precede the formal acquisition process, they still require measurement to provide insight into program status.

ATDs evaluate the feasibility and maturity of an emerging technology. They provide a relatively low-cost approach to assess technical risks and uncertainties of critical technologies prior to incorporating these technologies into an acquisition program. A successful ATD often leads to the start of an acquisition program, or its results may be integrated into a larger acquisition effort.

ACTDs help to respond quickly to urgent military needs. ACTDs employ available technologies that frequently have been successfully demonstrated in an ATD. In an ACTD, a system is designed, fabricated, and then demonstrated in realistic exercises. This builds an understanding of the utility of the system, supports development of a concept of operations, and elicits requirements by placing a limited, demonstrable capability into the hands of the user.

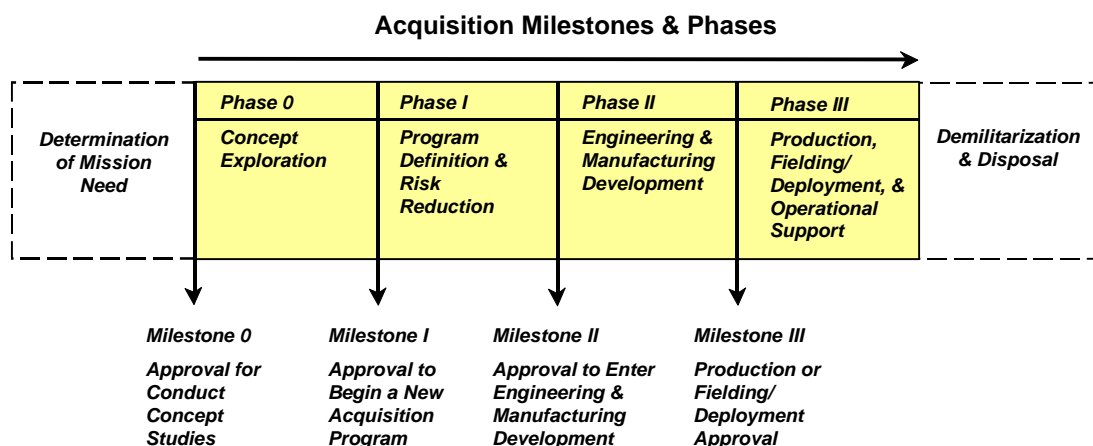
ATD and ACTD programs need to be managed correctly. The PSM process can be applied to these efforts, just as it does to any other program. However the range of issues may be narrower, since the objectives of these demonstrations are limited. Moreover, the ideal ATD or ACTD should not only demonstrate that something can be done, but should also provide quantitative information about the likely cost and

resulting quality of a product from the demonstrated technology. This requirement can be supported by measurement.

Measurement results from ATDs and ACTDs can be useful in the early stages of the acquisition life cycle, as explained below.

## 1.2 Acquisition Life Cycle

The acquisition life cycle begins with the definition of a need and a strategy to satisfy that need. Typically, needs are derived from the organization's mission and threat environment. A successful acquisition program delivers the required capability within established cost, schedule, and quality objectives. The DoD Acquisition Life Cycle depicted in Figure 8.1-1 defines a series of acquisition phases and related milestones. This sequence of activities is required for all major acquisition programs, included in Acquisition Category (ACAT) 1, IA and II. Each milestone results in a decision on whether to progress to the next phase. Traversing the complete life cycle may require many years.



**Figure 8.1-1. Acquisition Life-Cycle Phases Milestones**

The acquisition life cycle defines a systematic approach for approving work and ensuring that the intended product is delivered within program objectives. Figure 8.1-2 lists the objectives of each acquisition phase and the measurement activities that support achieving those objectives.



<b>Objectives</b>	<b>Measurement Activities</b>
<b>Phase 0</b> <ul style="list-style-type: none"> <li>• Evaluate alternative strategies</li> <li>• Identify most promising concept(s)</li> <li>• Identify risks and initiate risk management</li> <li>• Develop acquisition strategy and objectives</li> <li>• Define cost and schedule baseline</li> </ul>	<b>Phase 0</b> <ul style="list-style-type: none"> <li>• Analysis of historical data from similar programs to support Rough Order of Magnitude (ROM) estimates</li> <li>• Quantitative Risk Management</li> <li>• Analysis of Alternatives</li> </ul>
<b>Phase 1</b> <ul style="list-style-type: none"> <li>• Define detailed design and capabilities</li> <li>• Demonstrate critical technologies</li> <li>• Prove critical process attainable</li> <li>• Develop supporting analysis for a Milestone 2 decision</li> </ul>	<b>Phase 1</b> <ul style="list-style-type: none"> <li>• Independent Cost Estimates</li> <li>• Requirements measurement process</li> <li>• Analysis of Alternatives</li> <li>• Contractor proposal evaluations</li> <li>• Historical analysis</li> <li>• Measurement plan</li> <li>• Cost as an Independent Variable (CAIV)</li> </ul>
<b>Phase 2</b> <ul style="list-style-type: none"> <li>• Develop a stable requirements baseline</li> <li>• Validate manufacturing/production processes</li> <li>• Produce the first system</li> <li>• Test system capabilities against mission needs and specification requirements</li> </ul>	<b>Phase 2</b> <ul style="list-style-type: none"> <li>• Independent Cost Estimates (ICE)</li> <li>• Requirements measurement process</li> <li>• Contractor proposal evaluations</li> <li>• Historical data analysis</li> <li>• Measurement plan</li> <li>• Measurement process in place</li> </ul>
<b>Phase 3</b> <ul style="list-style-type: none"> <li>• Establish production and support base</li> <li>• Achieve operational capability meeting the users needs</li> <li>• Conduct follow-on operational and production verification testing</li> </ul>	<b>Phase 3</b> <ul style="list-style-type: none"> <li>• Full system measurement process</li> </ul>
<b>Phase 4</b> <ul style="list-style-type: none"> <li>• Determine if major upgrades to a system in production are warranted</li> <li>• Establish the appropriate baseline</li> </ul>	<b>Phase 4</b> <ul style="list-style-type: none"> <li>• Sustaining engineering measurement process</li> </ul>

**Figure 8.1-2. Phase Objectives and Measurement Activities**

Neither the acquisition life cycle nor PSM make any assumptions about the technology or methodology to be employed in developing the system under acquisition. Examples of different acquisition approaches include systems based on Non-Developmental Items (NDI), use of Joint Applications Development (JAD) techniques, and

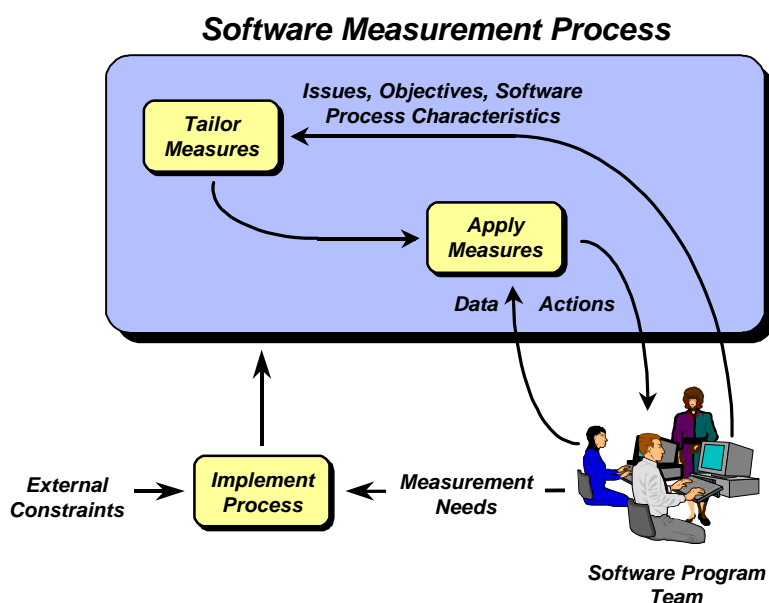
implementation of Rapid Application Development (RAD) processes. Measurement still applies in these acquisition approaches, as explained below.

For programs considered as NDI, where little or no development is required, an Integrated Process and Product Development (IPPD) approach should be employed as an independent evaluation and planning activity. In this case, measurement would not only consider performance and delivery cost, but also the cost of integrating and fielding the system, training, maintenance, long-term support, logistics, disposal, and follow-on products.

The identification of a need and the development of a set of requirements are required during a RAD or JAD. Additionally, some set of testable requirements must be developed along with associated measures. This enables the developer to validate the development process and assess the quality of the product.

### 1.3 Software Measurement in the Acquisition Process

During each phase of the acquisition life cycle, a measurement process as illustrated in Figure 8.1-3 can be applied to support contract requirements for a software-intensive system. The acquisition life cycle contains two major activities related to a contract, acquisition planning and acquisition management. A separate contract may be established to support each phase.



**Figure 8.1-3. Software Measurement Activities**

During acquisition planning, the contract is established and the mechanisms necessary to effectively manage the contract are put in place, including:

- The **Integrated Product Team** (IPT) is established as a cooperative forum for making decisions.
- The **Work Breakdown Structure** (WBS) is defined to itemize the products to be delivered and the tasks to be performed.
- The **Earned Value** (EV) Plan is drafted to assign budget and schedule to each of the products and tasks defined in the WBS.
- The **Risk Management Plan** is drafted to identify potential obstacles to program success and contingencies for dealing with them.
- The **Software Measurement Plan** is drafted to define the data to be collected and the analysis to be performed to determine whether the program is progressing according to plan.

As shown in Figure 8.1-4, software measurement results feed into the Earned Value reporting and risk management processes. Risk, measurement, and Earned Value information are all used by the IPT to track program status and to make decisions.

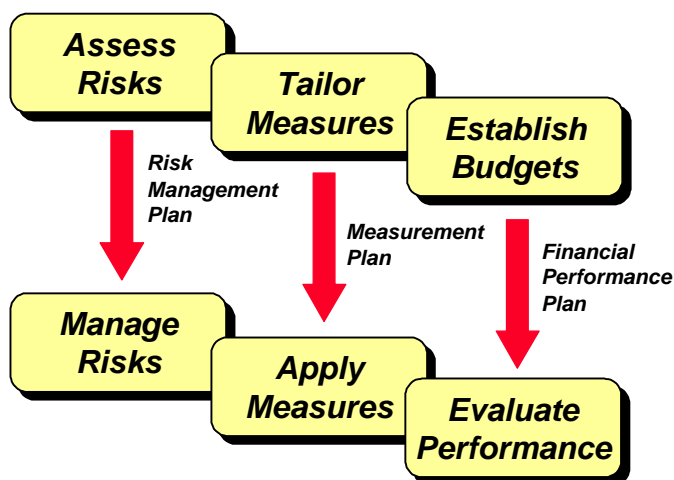


Figure 8.1-4. Quantitative Software Management



## CHAPTER 2 - MEASUREMENT FOR INTEGRATED PRODUCT TEAMS

Implementing many of the streamlining initiatives established by DoD acquisition reform policy requires a close working relationship between the developer and the acquiring organization. The IPT provides a mechanism for implementing this relationship. IPTs may be the primary users of measurement results in many programs. This chapter explains how measurement is implemented through an IPT.

In May of 1995, the use of IPTs became policy for all DoD programs. The use of IPTs has been incorporated into DoD Directive 5000.1 and DoD Regulation 5000.2, and defines different levels of IPTs, such as the Overarching IPTs (OIPTs), the Working-Level IPTs (WIPTs), and the Program IPTs.

The IPT concept is based on the approach of all parties working together to ensure successful implementation of individual programs. IPTs can be formally chartered or they can be informal working groups. They can function at levels that range from one team for multiple programs in an organization to small teams addressing one aspect of a single program. Implementation of the IPTs concept does not mean that an organization needs to restructure. The team is not the end goal, but rather the means through which much of the work is accomplished, including measurement. Figure 8.2-1, extracted from “A Guide for Leading Successful Integrated Product Teams,” describes how the government participates in various IPTs. It is easy to see that measurement plays an important role in almost every aspect of “Focus” and “Participant Responsibilities.”

A typical industry IPT would consist of two tiers. The first-tier team provides strategic direction, corporate oversight, and review. The measurement input to this team is at a high “management” level to provide summary information and trend analysis. This team should be a cross-functional team to optimize the chances for success.

The second tier of a typical industry IPT is made up of multiple sub-teams. These sub-tier teams, should also be multi-disciplinary, rather than functionally oriented. Each team should have a broad perspective of the product, process, and organization, rather than a centralized viewpoint. Each team should also have a specific charter that identifies expectations and responsibilities for program support. Sub-tier team leaders should also be members of the next higher tier team. Sub-tier team’s measurement requirements are determined by their domain. The teams are responsible for aggregation of their measures for the first-tier team.

<b>Organization</b>	<b>Teams</b>	<b>Focus</b>	<b>Participant Responsibilities</b>	<b>Measurement Relationship</b>
OSD and Components	OIPT	<ul style="list-style-type: none"> <li>• Strategic Guidance</li> <li>• Tailoring</li> <li>• Program Assessment</li> <li>• Resolve Issues Elevated by WIPTs</li> </ul>	<ul style="list-style-type: none"> <li>• Program Success</li> <li>• Functional Area Leadership</li> <li>• Independent Assessment</li> <li>• Issue Resolution</li> </ul>	<ul style="list-style-type: none"> <li>• Milestone Estimates</li> <li>• Feasibility Studies</li> <li>• Earned Value</li> <li>• Estimates to Complete</li> </ul>
	WIPTs	<ul style="list-style-type: none"> <li>• Planning for Program Success</li> <li>• Opportunities for Acquisition Reform (innovation, streamlining)</li> <li>• Identify/Resolve Program Issues</li> <li>• Program Status</li> </ul>	<ul style="list-style-type: none"> <li>• Functional Knowledge &amp; Experience</li> <li>• Empowered Contribution</li> <li>• Recommendations for Program Success</li> <li>• Communicate Status &amp; Unresolved Issues</li> </ul>	<ul style="list-style-type: none"> <li>• Milestone Estimates</li> <li>• Feasibility Studies</li> <li>• Estimates to Complete</li> <li>• Earned Value</li> </ul>
Program Teams & System Contractors	Program IPTs	<ul style="list-style-type: none"> <li>• Program Operation</li> <li>• Identify &amp; Implement Acquisition Reform</li> </ul>	<ul style="list-style-type: none"> <li>• Manage Complete Scope of Program, Resources &amp; Risk</li> <li>• Integrate Government &amp; Contractor Efforts for Program Success</li> <li>• Report Program Status &amp; Issues</li> </ul>	<ul style="list-style-type: none"> <li>• Design, Implementation, and Testing measures for all identified issues</li> </ul>

**Figure 8.2-1. DoD IPT Types, Focus, and Responsibilities**

The IPT concept differs from the traditional program organization concept, which usually focuses on single-function disciplines. IPTs are responsible for designing the product and its associated processes, and also for planning, tracking, and managing their own work and the processes by which they do their work. Successful application of IPPD rests heavily on the ability to form, align, empower, and lead these cross-functional teams.

The teams focus is on achieving set goals and objectives. Measurement is a means for creating and maintaining that focus. When measures provide meaningful indicators, IPTs can clearly understand their progress and better allocate resources for identified risks and the remaining tasks. Identification and management of risks are key responsibilities of each IPT.

## CHAPTER 3 - PUTTING MEASUREMENT ON CONTRACT

When software is acquired from an external organization, the program's measurement requirements need to be formally defined. This chapter explains how measurement is implemented in a contract between a government organization and a private contractor. These concepts also apply in varying degrees when software is acquired from another government organization via a Memorandum of Agreement (MOA) or Inter-Service Support Agreement (ISSA).

### 3.1 Overview of Contracting Process

---

The PSM measurement-tailoring process often supports a formal acquisition in which proposals are solicited and a developer is selected. Measurement must be coordinated within each of the four activities of the contracting process:

- contract planning and preparation
- proposal evaluation
- negotiation
- contract modifications

The following subsections describe each of these activities in more detail. Section 3.6 provides sample wording that may be inserted into a Request For Proposal (RFP) or a contract, along with the rationale for each contract requirement.

Through the contracting process, the program management team ensures that the necessary measurement mechanisms are in place to support the acquisition objectives for the program's current phase. This contracting process applies to both the development and sustaining engineering phases, although the issues and measures may differ. When adding measures to an existing contract, the contract planning and preparation and proposal evaluation activities are generally not implemented.

### 3.2 Contract Planning and Preparation

---

During contract planning and preparation, software measurement requirements are identified and documented. The RFP provides a vehicle to communicate these requirements to potential contractors. Section 3.6 contains sample wording that may be inserted into a RFP for this purpose. In the RFP, the program management team

may also request historical data to substantiate the developer's proposal and to conduct an independent feasibility analysis of the proposed software development plan. Section 3.6 also provides wording to request this data. In parallel with RFP development, the program management team usually develops independent estimates of size, schedule, effort, and cost to evaluate the contractors' proposals.

### 3.3 Proposal Evaluation

---

Contractors respond to an RFP with a proposal that explains how their measurement process will meet the program manager's information needs. Each measurement process proposed by the prospective contractors must be evaluated during the proposal evaluation process. This evaluation includes assessing the developer's understanding of the issues specified in the contract, the effectiveness of the process, and the measures that the developer plans to use to address the issues. The evaluation should assess the adequacy of the proposed measurement data definitions. An on-site evaluation at each developer's facility may be performed to validate the proposed measurement process identified in each proposal.

The proposal evaluation team also needs to assess the feasibility of each proposal's estimates of size, schedule, effort, and cost. The team may use software development cost and schedule estimation models to compute performance parameters and look for inconsistencies that need to be reconciled. In addition, the developer's estimates should be compared to the independent estimates developed by the program office. Feasibility of the proposed plan is also evaluated with respect to the historical data provided by the contractor.

### 3.4 Negotiation

---

Once a developer has been selected, negotiation helps to define the measurement requirements in the contract. In the proposal, the developer should have identified any concerns with the program office's specified issues and measures, and proposed appropriate alternatives. Alternate measures must adequately address the program office issues and be used within the developer's process to manage the software development.

The developer's proposal should identify any problems associated with the program office's measurement guidance, including the data items to be collected, the collection and reporting levels, and the method for counting actual data. The developer should describe the proposed implementation of the measures, including definitions, estimation techniques, actual measurement methods, and data reporting mechanisms. All of these items must be agreed upon during negotiations. The results of the



negotiations should be documented in the contract or in an approved software measurement plan.

### 3.5 Contract Modifications

---

It is important to understand that the software issues will change during the program. The measurement and contracting process has to be flexible to accommodate these changes. Different measures may be required to address new or modified issues, and changes may be required for data definitions, data elements, or reporting mechanisms.

Contract modifications may also be necessary to implement measurement on existing programs that did not originally require measurement. Even in these situations, the program management team should define program issues and measurement requirements. The team should work with the existing developer to determine if any measures are already available to address these issues.

### 3.6 Sample RFP Wording

---

This section contains sample wording that may be inserted into a Request For Proposal (RFP), contract, or other agreement between the program manager and developer. The sample wording may be used to request software measurement data, address questions about that data, and develop a software measurement plan.

Each of the following sections contain a description of the rationale for each request, followed by sample wording (in quotes in the shaded area) that may be directly inserted into an agreement.

#### 3.6.1 Requirements for Software Measures

Contract wording to require collection of measurement data should be specified. In the RFP, the program management team should identify the software issues and the measures required to address them. The program management team should define the characteristics of each required measure, including the data items to be collected, the collection and reporting level, and the method for counting actual data as complete. The following paragraph specifies monthly reporting, but this may be adjusted as appropriate for each program.

*“The developer shall provide the software measures specified in Paragraph XXX on a monthly basis. For each measure, data shall be provided for each data item at the specified collection level. Data shall not be considered as actuals until the criteria for counting actuals has been successfully met.”*

Requirements for most software measures should include both planned and actual performance data. Any changes to the planning data should be identified, quantified, and provided to the program manager. A few measures may not be accompanied by planning data (such as defect and requirements stability data).

*“For all of the measures specified in Paragraph XXX, the developer shall provide an initial plan and submit periodic actual data. Any time that the planning data for any of the detailed measurement parameters changes, the developer shall provide an updated plan within 30 days of the change.”*

For each measure, the developer should propose measurement definitions, methodologies, and data reporting mechanisms.

*“For each measure specified in Paragraph XXX, the developer shall provide a measurement definition, an estimation methodology, the method used to measure actual data, and the data reporting format and associated mechanism. This information shall include a description of any tools utilized.”*

*Planned and actual data shall be based on the same measurement methodology. Any changes in definitions, estimation methodologies, or actual measurement approaches shall be documented within 30 days of the change and shall require approval of the Program Manager.”*

The data should be provided in a timely manner, as soon as possible after data collection occurs. The sample wording in this section recommends that data should be reported within 30 days, but this time period may be modified. The lag time between data collection and reporting should be minimized to provide early-warning indicators.

*“The required measures shall be delivered within 30 days after the data is collected.”*

### 3.6.2 Developer Access

Throughout the development, the program management team should periodically review the measurement process. In addition, there will be questions about some of the data. The acquisition program office needs to have access to the developer to answer questions and to gather the subjective data required for interpretation of the quantitative data.

*“The developer shall provide direct access to the program team to facilitate open communications with respect to the measurement process. The developer shall also provide a rationale for changes, answer questions, and provide clarifications regarding the measurement process and associated data and information.”*

### 3.6.3 Data Alternatives

The measures specified in the RFP represent the initial issues of the program manager. The developer may request substitution of an alternate software measure if the alternative measure provides similar insight into the same software issue. The alternative measure should be readily available and used internally in the developer's process.

*"In the event that a specified measure is unavailable, the developer shall submit a request for substitution. This request shall identify an alternative measure with a data definition, a rationale for the change, a description of how this measure addresses the identified issue, and a description of how this measure will be used internally. The alternative measure must be readily available from the software development process."*

### 3.6.4 Draft Measurement Plan

The developer should be required to develop a measurement plan that specifies which issues and measures will be addressed during the program. The plan should identify the proposed software measurement process and specify how the developer will use the measurement information.

*"The developer shall submit a draft measurement plan that specifies the issues to be addressed, the measures to be utilized, and definitions of specified measures and measurement methodologies. This plan shall identify the measurement approach to be used, including a description of how measurement information will be used in the developer's internal management of this program, how data will be collected, points of contact, responsibilities, and organization communications and interfaces."*

### 3.6.5 Proposal Evaluation Data

Proposal evaluation should include an assessment of the feasibility of the software development plan, based on information provided in the proposal, historical data on the developer's performance, and independent estimates prepared by the program management team. Information used for this assessment includes:

- **Required Productivity** - The developer should provide an assessment of the productivity required to successfully complete the project, based on the planning parameters provided in the proposal. The developer should also include a definition of any tools or methodologies used.
- **Product Size, Effort, Milestone Dates** - The developer should submit estimated data for each of these measures to allow the proposal evaluation team to do an

independent feasibility assessment on each bidder. The data should describe the data definitions and estimation methodology.

- **Historical Data** - The developer should submit actual data (product size, effort, milestone dates, cost, and productivity) from completed programs. Data should be provided from programs that are similar in domain, size, and complexity to the proposed program.

The first two items are usually required parts of the proposal, whether or not the measurement approach described in this Guide is applied. The following sample RFP wording is suggested to collect historical data to substantiate the potential developer's proposal and to conduct the feasibility analysis:

*“The developer shall provide historical data from at least three completed programs to support the proposal. The technical characteristics of the historical programs shall be similar to the proposed system with respect to domain, size, and complexity. If the developer does not have experience within these criteria, data from other completed programs shall be provided. The data shall include measures of size, schedule, effort, cost, and productivity by WBS element. Any models and methodologies used shall be documented for each historical program to a sufficient level of detail to allow replication by the evaluation team.”*

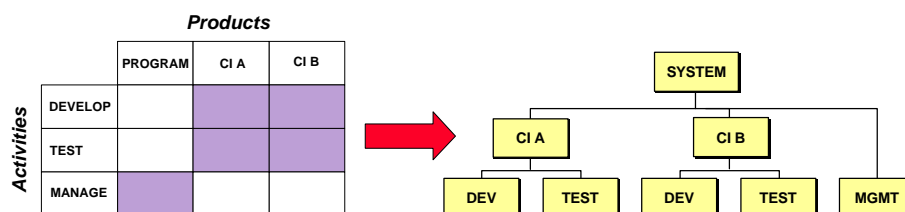
A software measurement plan should be developed to ensure the required information has been identified and a definitive process is established. Figure 2.4-2 in Part 2 of the PSM Guide contains a sample outline of a Software Measurement Plan. This plan should be modified as needed to accommodate different program information needs and developer processes. It may be included as part of the Software Development Plan (SDP), Software Maintenance Plan (SMP), or similar planning document.

## CHAPTER 4 - USING WORK BREAKDOWN STRUCTURES

This section contains examples of Work Breakdown Structures (WBS) for information, communication, and weapons systems. These examples can be used as a basis for a contract WBS and as a tool for collecting some measures, such as work unit progress. This material describes how to use the WBS with the development organization.

A WBS is an important management tool used to identify all accountable areas in the development, operations, or maintenance process. In a commercial environment, the WBS must be directly linked to the cost accounting system. This linkage would be at the same level of the WBS as the work packages that are managed and reported. In a government environment, a time card system can be implemented that would be directly tied to the WBS. By applying measures at these WBS levels, a manager can quickly focus on areas that directly address their issues. A WBS is normally first developed to define products at three or four levels in the system structure. The WBS can be expanded to include the process information that is directly related to the product.

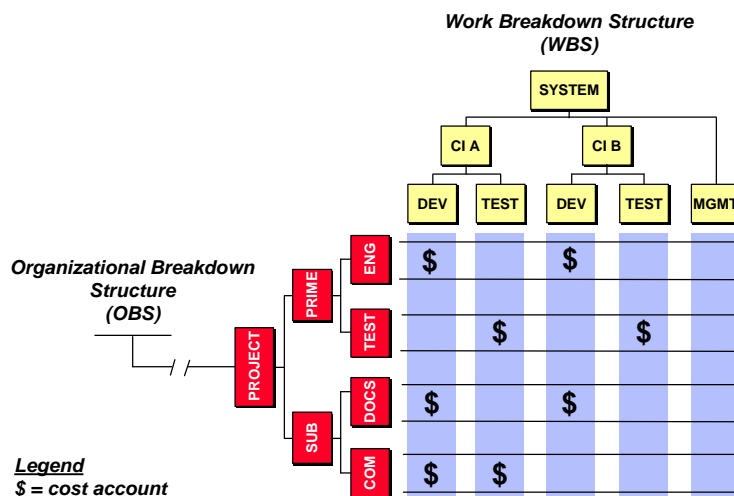
Figure 8.4-1 shows an example of a simple software program with two CIs and a two-activity process model. Each of the activities applies to each of the CIs. In addition the management activity applies across the system. The intersections between the process and product structures define five work packages. Usually a budget and schedule are assigned for each work package. These elements of a WBS are commonly organized into a hierarchy diagram as shown in Figure 8.4-1.



**Figure 8.4-1. Mapping Program Products and Activities**

A work package could correspond to something as large as developing an entire Configuration Item (CI) over a period of years or as small as testing a single unit within one week. Most programs define work packages for each major activity, such as requirements analysis, design, implementation, integration and test, and rework, for each CI. However, to adequately address specific program issues it may be necessary to collect one or more types of data at a more detailed level.

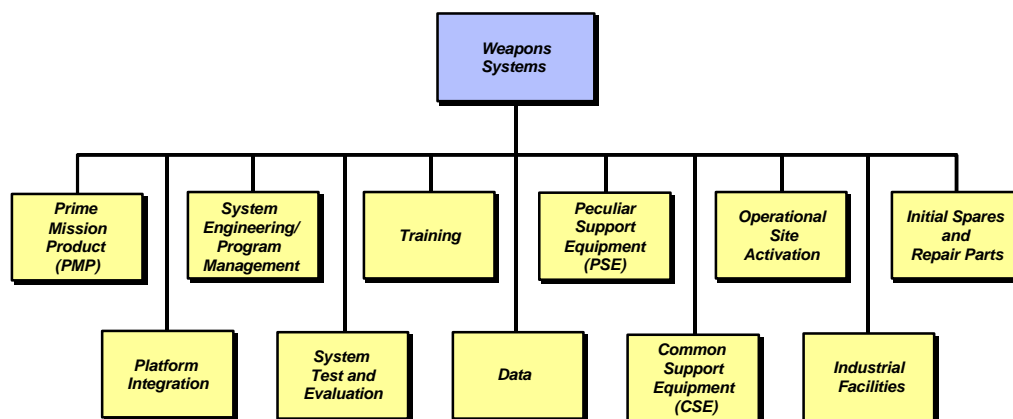
Figure 8.4-2 shows the final task in defining the WBS structure, mapping responsibility for work packages to organizational elements.



**Figure 8.4.2. Cost Accounts**

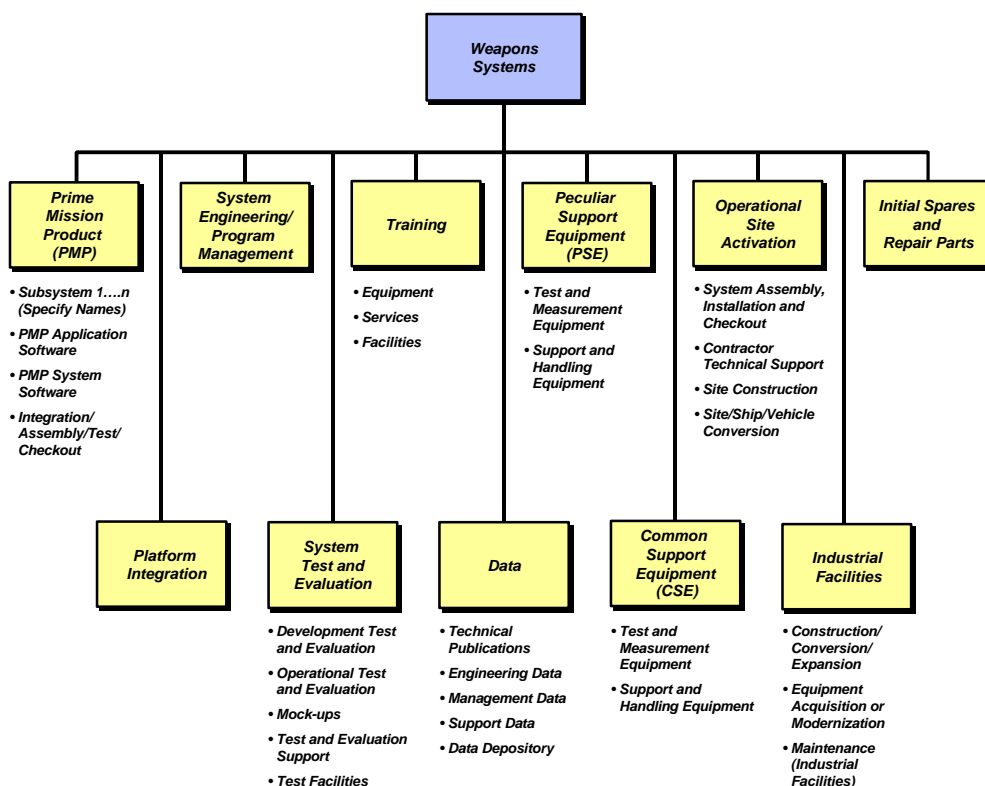
Figure 8.4-2 shows that WBS and organizational elements intersect. These intersections usually correspond to the cost accounts that track budgets and expenditures in most financial systems. These cost accounts define the interface between the software measurement and financial reporting processes. Planning the measurement process appropriately enables it to support the analysis of a program's financial situation.

Figure 8.4-3 contains a high-level WBS that can be used as a starting point for development and support of a weapons system. This level may be the minimum level required for contracts that require no insight for reporting purposes.



**Figure 8.4-3. Weapon System WBS**

Most likely, at least one level lower is reported in a WBS, as described in 8.4-4. The challenge is to use the WBS to assist in reporting of issue-related information. This is accomplished by expanding those areas of concern in the WBS to allow data to be collected on the identified and expected issues. A WBS also needs to accommodate the constraints of the cost accounting or data collection system. A good guide to help develop a WBS is Military Handbook 881-B. It is important to remember that every data element collected should be linked to a WBS element, either directly or indirectly, and the link should be documented. Once the links are established, the automated collection mechanisms can be put into place.

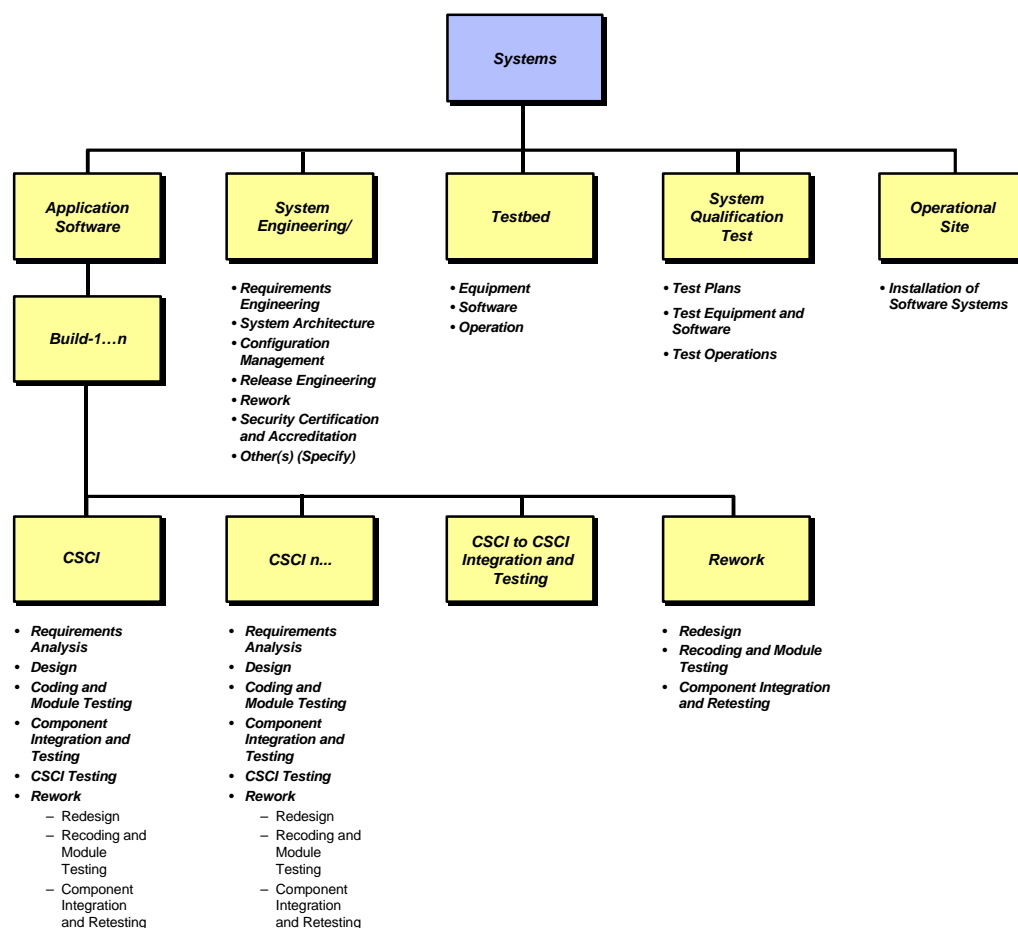


**Figure 8.4-4. Weapon System WBS (reference MIL HDBK 881B)**

In a commercial environment, those measures that include cost or effort data require the cost accounting system to have the work unit codes tied directly to the desired WBS reporting level. For time card systems, an audit check is normally conducted on the time cards before they go into the accounting system. This may result in a delay in the period of time when the actions are completed and information can be reported. Some companies have upgraded their systems to daily reporting and have periodic audits to ensure accuracy. For an in-house or government operation, it would be more difficult to implement a time card system in which the reported effort is tied to a specific WBS element. A time card system with work unit codes tied to a WBS would normally be developed and maintained within the organization.

For those measures requiring other data, such as defects, similar mechanisms must be implemented at the appropriate WBS levels. The critical link back to the WBS is often left out of a data collection effort. For example, defects could be linked to rework in a specific CI. Therefore, a WBS element for rework on the CI must be included, and a work unit code must be identified for the cost accounting system.

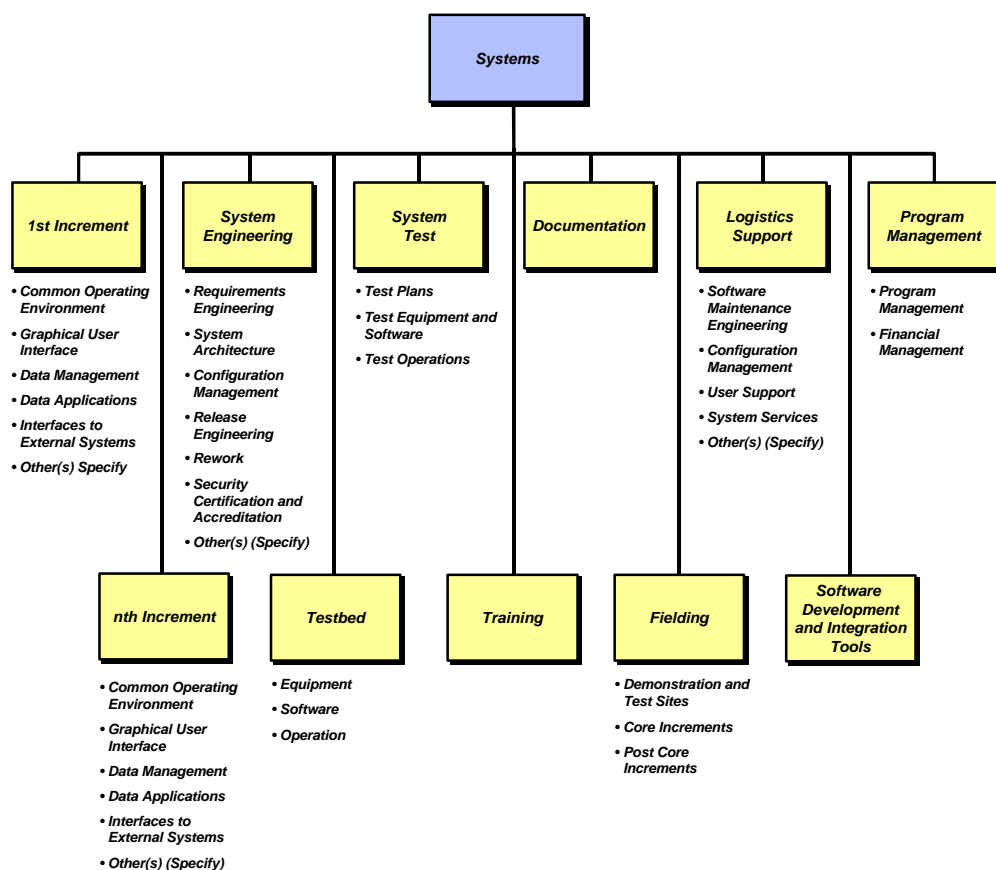
Figure 8.4-5 contains an expanded sample WBS that may be a subset of Figure 8.4-4, or could describe a stand-alone software management effort. This sample WBS could also be used within an organization to identify the lower-level elements that are targeted for data collection.



**Figure 8.4-5. Expanded Sample Software WBS**

Figure 8.4-6 contains an example of an expanded WBS that can be used in information system development and sustaining engineering. This example contains many elements that appear to be outside the normal activities in a software management effort, such as security certification and accreditation. A WBS should identify all areas that may affect the total cost and schedule of the software or the rest of the system.





**Figure 8.4-6. Expanded Sample Information System WBS**

After contract award, it is important to modify the WBS used during the selection process and to map to the selected developer's negotiated WBS. The revised WBS ties the estimated measures of the government to the actual measures collected by the developer. The revision also ensures that the cost account elements map to the same WBS that is used for data collection.



## CHAPTER 5 - INTEGRATING MEASUREMENT WITH EARNED VALUE

This Chapter describes how Earned Value can be used as part of the measurement process.

Earned Value is an performance management approach that some organizations use to assess the cost and schedule against the amount of work being performed. Earned Value requires cost and schedule estimates to be identified with specific work packages that have their own cost account. An example of this is illustrated below for a work unit package in the detailed design, code and unit test phase of a software development.

Earned Value is a management technique that relates resource planning to technical, cost, and schedule requirements. All work is planned, budgeted, and scheduled in time-phased “Planned Value” increments that constitute a cost and schedule measurement baseline. An Earned Value Measurement System (EVMS) uses Budgeted Cost of Work Performed (BCWP) as a basic Earned Value indicator. The Budgeted Cost of Work Scheduled (BCWS) is used as the Planed Value indicator to determine the Schedule Variance. The BCWP is compared with the Actual Cost of Work Performed (ACWP) as the Actual Cost indicator to determine the Cost Variance. The two major objectives of an Earned Value system are to encourage contractors to use effective internal cost and schedule management control systems, and to permit the customer to evaluate the status of deliverable products. The following example was modified from the OUSD(A&T) Web site to illustrate the ease of use of Earned Value on a software work unit package during detailed design, code and unit test.

The example baseline plan in Figure 8.5-1 shows that six work units (A to F) should be completed at an estimated cost of \$100 for the period covered by this report. A and B are associated with the detailed design of a component. A credit of \$10 to A would be given when a component is released to the team/individual responsible for the detailed design of the component. An additional \$15 credit is given to B when the component design completes a successful peer design walkthrough. A credit of \$10 to C is given when the team begins coding of the component, and a credit of \$25 to D when the component has completed a successful peer code walkthrough. Finally, when a component enters unit test, it is given a credit of \$20 to E. A credit of another \$20 is given to F when it has successfully completed unit test. In some situations a percentage of the allocated effort can be given credit if it is partially completed. This depends on how the Earned Value system is established and how credit is allocated for the work.

	<b>Unit Design</b>		<b>Unit Code</b>		<b>Unit Test</b>		
	A	B	C	D	E	F	Total
Planned Value (\$)	10	15	10	25	20	20	100

**Figure 8.5-1. Baseline Plan Work Units**

As work is performed, it is “earned” on the same basis as it was planned, in dollars or other quantifiable units such as labor hours. Planned Value compared with Earned Value measures the dollar volume of work planned vs. the equivalent dollar volume of work accomplished. Any difference is called a schedule variance. In contrast to what was planned, Figure 8.5-2 shows that the code walkthrough D was not completed. Work unit test E began, but nothing had been completed as shown by F. Therefore, \$35 of the planned work was not accomplished. As a result, the schedule variance shows that 35 percent of the work planned for this period was not done.

	<b>Unit Design</b>		<b>Unit Code</b>		<b>Unit Test</b>		
	A	B	C	D	E	F	Total
Planned Value (\$)	10	15	10	25	20	20	100
Earned Value (\$)	10	15	10	10	20	0	65
Schedule Variance	0	0	0	-15	0	-20	-35=-35%

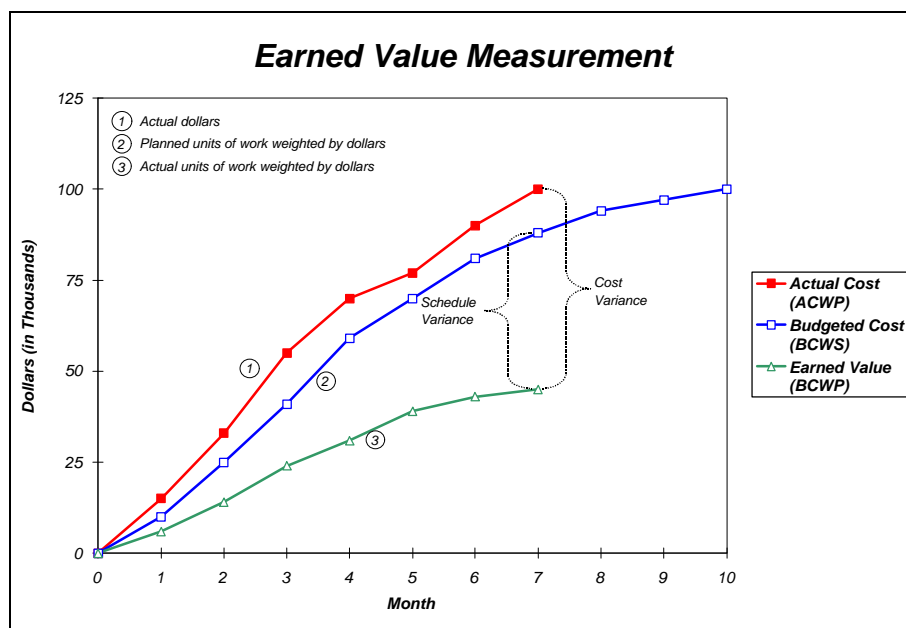
**Figure 8.5-2. Schedule Variance Work Units**

Earned Value compared with the actual cost incurred (from contractor accounting systems) for the work performed provides an objective measure of planned and actual cost. Any difference is called a cost variance. A negative variance means more money was spent for the work accomplished than was planned. Figure 8.5-3 shows the calculation of cost variance. The work performed was planned to cost \$65 and actually cost \$91. The cost variance is 40 percent.

	<b>Unit Design</b>		<b>Unit Code</b>		<b>Unit Test</b>		
	A	B	C	D	E	F	Total
Earned Value (\$)	10	15	10	10	20	0	65
Actual Cost (\$)	9	22	8	30	22	0	91
Cost Variance	1	-7	2	-20	-2	0	-26=-40%

**Figure 8.5-3. Cost Variance Work Units**

The organization can graphically represent Earned Value as a single chart for any part of the development or work package, or for the entire program, as depicted in Figure 8.5-4.



**Figure 8.5-4. Earned Value Measurement**

Earned Value can be used as an indicator for any identified unit of work that is associated with estimated and actual cost/effort and schedule. Other measures can be used as technical indicators for variance analysis, including requirements stability, design stability, program size, and computer resource utilization. The Earned Value approach benefits program management by requiring disciplined planning. The availability of the Earned Value measures show the real variances from plans to identify necessary corrective actions.

## SUMMARY

Acquisition reform has streamlined DoD acquisition. In place of previously mandated requirements, a series of guidance documents have been developed to promote the use of best practices, including PSM. Each of the services has developed individual streamlining approaches and guidelines for acquisition and development activities. These guidelines are designed to facilitate management of individual programs, and still provide the insight required for effective management of the program.

The Office of the Undersecretary of Defense for Acquisition and Technology (OUSD/A&T) is responsible for implementing most of the acquisition reforms. Their Web address is [www.acq.osd.mil](http://www.acq.osd.mil). Specific initiatives by each service include the “Lightening Bolt Initiatives” by the Air Force, the “Acquisition Thrusts” by the Navy and the “Initiatives with Thrusts” by the Army. A list of the initiatives for each service can be found on the web at [www.safaq.hq.af.mil/SAFAQ/](http://www.safaq.hq.af.mil/SAFAQ/) for the Air Force, [www.acq-ref.navy.mil/](http://www.acq-ref.navy.mil/) for the Navy, and <http://acqnet.sarda.army.mil/> for the Army.

The following examples from the Navy web site provide suggestions to facilitate getting reform into practice.

- Commit to quality customer focus and continuous improvement.
- Manage with early insight on program issues, rather than after-the fact oversight.
- Manage overall life-cycle cost not just initial acquisition cost.
- Treat total cost as an independent variable relative to user requirements.
- Make cost performance trade-offs early in the acquisition process.
- Put high priority on logistics and support cost visibility.
- Integrate oversight requirements with contractor program management scheme.
- Use past performance as a key factor.
- Achieve quality with statistical process control rather than with end item inspection.
- Control only the performance specification giving contractors freedom for design innovation.

Acquisition reform and acquisition streamlining is an ongoing effort that must be supported by an effective measurement process. The measurement process must be flexible to accommodate the ever-changing management issues and to ensure the indicators have a solid foundation. The use of PSM will help ensure that the ideals of acquisition reform are achieved.