Managing For Success
How long can you wait for CMMI® Compliance?

Manage your software and systems engineering projects in guaranteed compliance with the CMMI NOW!

processMax 2i and 3i include all the necessary policies, procedures, guidelines, criteria, templates, and forms in role-based, step-by-step instructions, ready for use—everything you need for compliance with the CMMI-DEV at Level 2 and Level 3 respectively. Integrated with robust document management, workflow, and automated measurement and reporting, processMax is the intranet web-based solution for effective and efficient management of your software and systems engineering projects. processMax enables you to start operating in compliance with the CMMI immediately! We guarantee that organizations using processMax will not fail a formal appraisal led by an independent SEI-licensed appraiser.

pragma SYSTEMS CORPORATION
www.pragmasystems.com
877.838.PMAX
info@pragmasystems.com

GSA Schedule Contract No. GS-35F-0559S. processMax is a registered trademark of pragma SYSTEMS CORPORATION. Although processMax makes use of portions of “CMMI” for Development, Version 1.2, CMU/SEI-2006-TR-068, copyright 2006 by Carnegie Mellon University, neither the Software Engineering Institute nor Carnegie Mellon University have reviewed or endorsed this product.

Copyright © 2008 pragma SYSTEMS CORPORATION.

This is a paid advertisement.
According to noted authors Tom DeMarco and Tim Lister, “The major problems of our work are not so much technological as sociological in nature”. They go on to say that most managers understand that people issues are more important than technical issues, but they seldom manage that way [DeMarco and Lister 1999]. Capers Jones, also a noted author and researcher, determined that, based on a study of 250 large software projects, there are six major factors that are different between successful and unsuccessful projects. These factors, shown in Table 1, are associated with project management rather than technical personnel [Jones 2004]. Finally, Rex Hughes said that, when software maintenance problems occur, managers tend to buy a new set of CASE tools, but that only solves 5% of the problem [Hughes 1991]. It can be inferred that 95% of software maintenance problems are managerial!

<table>
<thead>
<tr>
<th>Successful Projects</th>
<th>Falling Projects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effective project planning</td>
<td>Inadequate project planning</td>
</tr>
<tr>
<td>Effective project cost estimating</td>
<td>Inadequate cost estimating</td>
</tr>
<tr>
<td>Effective project milestone tracking</td>
<td>Inadequate milestone tracking</td>
</tr>
<tr>
<td>Effective project change management</td>
<td>Ineffective change control</td>
</tr>
<tr>
<td>Effective project quality control</td>
<td>Inadequate quality control</td>
</tr>
</tbody>
</table>

Table 1: Opposing Major Factors in Study Analysis [Jones 2004]

Project management must consider the attributes of a project. According to Kathy Schwalbe, a noted author of books on project management, a project has the following attributes:

- A unique purpose
- Is temporary
- Is developed using “progressive elaboration”
- Requires resources (people, hardware, software, etc.)
- Should have a primary sponsor, or customer
- Involves uncertainty

Projects differ from operations, work done to sustain a business, in that they end when objectives have been reached or they have been terminated. Schwalbe goes on to explain project management as “the application of knowledge, skills, tools, and techniques to project activities to meet project requirements” [Schwalbe 2009].

Software projects have been especially challenging to manage, primarily because of the unique attributes of software, such as intangibility, complexity, and (perceived) ease of change. Much has been written about software project management, especially this decade. Amazon.com lists several full-length books about software project management, including “Software Project Management for Dummies”! Roger Pressman, whose Software Engineering book is well-known, devotes seven chapters of his latest edition to managing software projects [Pressman 2005]. The Software Engineering Body of Knowledge (SWEBOK) has software engineering management as a major topic area and the contents are mostly software project management activities [SWEBOK 2004].

This issue of the Software Tech News contains several articles that can help the reader enhance his or her software project management knowledge and skills. The first two articles address earned value management (EVM), a key component of effective software project management. (There are several other earned value management articles in Software Tech News Volume 12, Number 1, April 2009, “Earned Value Management: Keeping Your Project on Track”.) The first article by Anita Cukr, a certified Project Management Professional who is a noted author and lecturer in EVM, addresses myths associated with EVM. One important conclusion of this article is that EVM is not merely a financial reporting system, but an integrated project management approach. The second article by noted EVM author Paul Solomon provides a fresh approach to effectively using EVM for software projects using agile development methods.

The next article by Don Reifer, a renowned leader in the fields of software engineering and software management, addresses risk management. This is also a key component of effective software project management that is not always understood or emphasized. The next article by Carol Dekkers, an international expert in the field of software measurement, technology and globalization, addresses scope management (or requirements management). This article presents an unusual analogous look at how the formal scope management concepts can be expressed in terms of triathlon athlete performance in training and racing. The final article by three Program Directors at Defense Acquisition University (DAU) describes DAU courses that can be useful in improving project management skills. DAU courses are usually geared to the Department of Defense environment, but can be useful to anyone involved in acquisition management or project management in general.

References:

[DeMarco and Lister 1999] DeMarco, Tom, and Lister, Tim, “Peopleware: Productive Projects and Teams” (Second
Dan Ferens is currently the DACS Director and serves as an instructor for a 12-part series in software affordability which has been taught mainly to Air Force Research Laboratory (AFRL) scientists, engineers, and managers in Rome, NY. Dan retired from AFRL in early 2007 after more than 35 years of service to the Air Force as a military and civilian employee. Dan has been involved in software estimating since he became a civilian in 1978, both as an AFRL analyst and program manager, and as a Professor at Air Force Institute of Technology where he taught classes on software estimation and other software engineering and management topics for 13 years. He is currently an Adjunct Instructor at SUNY Institute of Technology in Utica, New York where he teaches a class in information technology project management. He is a life member of the International Society of Parametric Analysts (ISPA), where, in 1999, he received the prestigious Freiman award for lifetime achievements in parametric estimating. He is also a member of Toastmasters, International where he holds the rank of Distinguished Toastmaster. Mr. Ferens has a Masters degree in Electrical Engineering from Rensselaer Polytechnic Institute, and a Masters Degree in Business Administration from the University of Northern Colorado. He and his wife, Marcie, currently reside in Fulton, New York.

Author Contact Information
Email: daniel.ferens@itr.com

The DACS website has been updated with new research information.

The DACS is a central distribution hub for software technology information sources. The DACS offers a wide-variety of Technical Services designed to support the development, testing, validation, and transitioning of Software Engineering technology.
Dispelling Some Myths about Earned Value Management (EVM)

EVM has been in use since the United States (US) Department of Defense (DoD) implemented it in the 1960s. Over the years, it has evolved from a US DoD mandated approach to an international industry standard.

by Anita Cukr

Earned Value Management (EVM) is a time-tested and proven project management approach that can provide management with uniquely integrated project performance insight. If supported by a properly implemented Earned Value Management System (EVMS) and EVM Performance Measurement Baseline (PMB), managers at various levels of a project have at their disposal recurring performance metrics that provide an integrated picture of cost, schedule, and technical progress. EVM has been in use since the United States (US) Department of Defense (DoD) implemented it in the 1960s. Over the years, it has evolved from a US DoD mandated approach to an international Industry Standard, with its guidelines captured in ANSI EIA 748-A. Its use has grown tremendously over the years to include diverse groups such as commercial industry, other countries, and all US Government agencies. The United States Office of Management and Budget has recognized the value of EVM as a risk mitigation approach and requires the use of EVM on Firm-Fixed Price and any other type of contract or task order that meets the major acquisition threshold if that contract or task order contains a significant amount of development effort (OMB Circular A-11, Part 7, Capital Programming Guide).

In spite of the success of EVM, misperceptions regarding this management approach are surprisingly common in the author’s experience. These misperceptions or “myths” can lead project managers to avoid the use of EVM for projects that would likely benefit from its use. The four “myths” to be explored are:

1. EVM is a financial reporting system
2. EVM data is too old to be effective for project management needs
3. EVM will not work on projects that involve Level Of Effort (LOE)
4. EVM is not an effective approach for projects that face rapidly changing requirements.

The article will conclude with a brief discussion of the use of the Integrated Baseline Review (IBR) process as a means of fostering effective implementation of EVM.

Myth 1: EVM is a financial reporting system

EVM is an integrated project management approach that includes financial, schedule and performance information. The EVM data is presented in currency terms as cost and schedule variances to the Performance Measurement Baseline. This EVM historical data is useful in projecting program at-completion cost and as such certainly provides excellent financial reports. However, project managers who view EVM as a financial reporting system may shortchange themselves by relegating its use to the financial community rather than the technical management community. Only if technical managers use EVM data on a regular basis will it reach its full potential as a project management approach which can be used to help make the necessary tradeoffs between cost, schedule, and performance constraints.

Since people new to EVM may get their first exposure to it by reviewing these reports, it is likely that they may promote the recurrence of the myth. Efforts to dispel this myth will require leadership emphasis and of course, training. An excellent opportunity for training technical managers in the use of EVM is found in the project Integrated Baseline Review (IBR) process, to be discussed at the end of this article. During the IBR process, the client and contractor program management technical teams thoroughly review the PMB, which is the integrated program plan (cost, schedule, scope, resources, risk) that is the critical foundation for the effective implementation of EVM.

Myth 2: EVM data is too old to be effective for project management needs

United States Government agencies require EVM reports by policy, but even without the policy requirement, Program Managers should ask for EVM reports, since they provide useful program management metrics. It is common for
Program Managers to require monthly EVM reports, and these reports typically reflect data that is actually close to two months old by the time the report is final.

Some may think this type of “old” EVM data has lost its relevance as a real time management tool. This kind of misperception reflects a lack of understanding of two key aspects of EVM. The first key aspect is the predictive value of EVM data, most famously documented in the case of the Navy A-12 Avenger II Program, cancelled due to performance problems detected by EVM data. There is also substantial research to support the use of EVM data in monthly summary reports as an excellent predictive tool. The second key aspect is that EVM is a scalable, practical management approach in which project line managers may assess progress real-time even though they may only provide formal reports monthly. The Project Manager may require a formal monthly summary report, but require the line managers to provide weekly, less formal reports, perhaps via staff meetings.

Myth 3: EVM will not work on projects that involve LOE

Confusion around terms and concepts can hamper effective program management. The use of the term “level of effort” in the discussion of EVM may be an example of such a confusing term/concept. Program Managers considering an EVM approach need to understand the concept of level of effort in the EVM context.

A useful first step is to recognize that all projects involve some level of effort, but typically they also involve some measurable outcome. If the projects involve risk, the use of EVM can help put more rigor in the program management process by focusing attention on the measurable outcomes. A crucial part of the EVM approach is a thorough, disciplined analysis of the work into plans with measurable outcomes, minimizing the use of plans without measurable outcomes (level of effort plans).

EVM is a management approach which provides Program Managers information on deviations from plan (cost and schedule variances), so that Program Management attention can focus on problem areas in order to most effectively finish the program within program objectives. The planning methods and measurements chosen to measure progress are of critical importance to the meaningfulness of the cost and schedule variances. “Measurable” approaches are preferred. Level of effort planning is not considered a “measurable” approach since it assumes progress is accomplished just by having a certain level of manpower over time, with no measurable outcomes. The EVM guidelines specify that level of effort planning should be minimized.

In practice, level of effort work packages are common in EVM systems, probably because this approach is an easy one since managers are required to estimate manpower levels whether they are using an earned value management system or not. Going the step further to think about measurable outcomes or products that might provide a better progress measure is additional work for busy managers. Yet this is an essential step in the development of effective earned value management systems. Program Management teams often track the prevalence of their use in an attempt to increase the use of more meaningful measures. The use of level of effort planning is also typically discussed during the Integrated Baseline Review.

In conclusion, EVM allows for the use of level of effort planning, so it is completely compatible with projects involving level of effort. More importantly, the EVM process encourages a thorough examination of level of effort work classifications to determine if more effective ways of planning and measuring the work are feasible.

Myth 4: EVM is not an effective approach for projects that face rapidly changing requirements.

This misperception typically has its roots in the notion that the EVM Performance Measurement Baseline is a detailed plan from project start to project finish, with little or no opportunity for change. EVM guidelines do require a baseline plan that covers the entire project period of performance, but this is accomplished through detailed planning horizons (work packages) and less detailed planning horizons (planning packages and summary level planning packages). With regard to detail planning, EVM guidelines require a baseline horizon that should not be subject to change – otherwise, measuring cost and schedule variances to that baseline might become meaningless. However, the guidelines are very flexible – only retroactive changes (including those in the current period) are prohibited with some exceptions, in order to maintain the integrity of cost and schedule variance data. Beyond that, the PM and his team must adopt a reasonable planning approach that includes structured change rules and that serve the needs of

---

1Article by David Christensen, Acquisition Review Quarterly, Summer 1999, “Using the Earned Value Cost Management Report to Evaluate the Contractor’s Estimate at Completion.”
the particular program. In the case of a well-defined program, detail planning from beginning to start may be feasible and sensible, while a program with rapidly changing and moving requirements might make liberal use of high level planning packages and shorter (3 month, 6 month?) detail planning windows. The Program Management team should convert the planning packages into detail plans at the earliest opportunity, keeping in mind effective program management needs. The use of management reserves as a risk management tool also adds flexibility into the Program Manager’s toolkit.

Once technical managers understand the flexibility inherent in EVM internal re-planning and change incorporation rules, this myth can be quickly dispelled.

**Dispelling the Myths**

Simple myths like the ones discussed here can become pervasive and undermine effective use of EVM systems and data. Clearly training is a necessary ingredient in dispelling these myths, and the most effective approach in this regard is to embed training in the routine management processes.

Many programs use the Integrated Baseline Review (IBR) process to institutionalize a solid understanding of EVM among the entire program management team, customer and client. The goal of the IBR is to enable the technical management team to gain an in-depth understanding of the schedule, resource, and technical planning including risks and progress measures in the project Performance Measurement Baseline. The IBR team should ideally include members from the technical, financial, scheduling, risk management, and contracting communities thereby providing an opportunity for these critical players in the project management process to better understand the Performance Measurement Baseline. Government has policy that requires IBRs at specific trigger points, but Program Managers may conduct IBRs at any time they need to obtain a better understanding of the PMB.

For more information about Earned Value Management and the IBR process, the reader may reference the National Defense Industrial Association’s guides “The Earned Value Management Systems Intent Guide” and “The Program Manager’s Guide to the IBR.”

---

**About the Author:**

Anita J. Cu Kr has 30 years of experience in financial and project management supporting a variety of organizations in the United States (US) government sector. She retired as a US Air Force officer, and went on to work as a consultant to federal government agencies such as the Federal Aviation Administration, the Office of Personnel Management, and the Missile Defense Agency. She spent her Air Force career in major weapon system acquisition management in the fields of cost estimating, budget, and earned value management. Currently, Ms. Cu Kr is devoting her time to consulting on earned value management implementation.

Ms Cu Kr is a certified Project Management Professional, a former Professor of Financial Management at the Defense Acquisition University, and a former Vice President of Communications for the College of Performance Management. She has lectured to hundreds of students at the Defense Acquisition University and at conferences nationwide. She is also a published author, having written a number of articles on earned value management.

**Author Contact Information:**

Email: Anita Cu Kr [cu kr@evmsconsultants.com]
Satisfy Your Quality Obsession

Software quality and reliability are mission critical. The size, pervasiveness, and complexity of today’s software can push your delivery dates and budgets to the edge. Streamline communication, improve traceability, achieve compliance, and deliver quality products with Seapine Software’s scalable, feature-rich application lifecycle management solutions:

- **TestTrack Pro** — Development workflow and issue management
- **TestTrack TCM** — Test case planning and tracking
- **Surround SCM** — Software configuration management
- **QA Wizard Pro** — Automated functional and regression testing

Designed for the most demanding software development and quality assurance environments, Seapine’s flexible cross-platform solutions adapt to the way your team works, delivering maximum productivity and saving you significant time and money.

Visit [www.seapine.com/gsa](http://www.seapine.com/gsa)

© 2009 Seapine Software, Inc. All rights reserved.

This is a paid advertisement.
Agile Earned Value and the Technical Baseline
AGILE METHODS ARE PRIMARILY USED FOR COMMERCIALY-DEVELOPED SOFTWARE WHEN THERE ARE FREQUENT DELIVERIES OF USABLE SOFTWARE THAT MEET THE CUSTOMER’S HIGHEST AND MOST CURRENT PRIORITIES.
by Paul J. Solomon (Performance-Based Earned Value)

In the April 2009 STN, John Rusk published “Earned Value for Agile Development.” This article augments Rusk’s points by providing guidance for using Agile methods with earned value management (EVM) when the Performance Measurement Baseline (PMB) is not linear. Other topics are measuring progress towards defining and implementing the technical baselines and accounting for deferred functionality.

Agile methods are primarily used for commercially-developed software when there are frequent deliveries of usable software that meet the customer’s highest and most current priorities. This article addresses the usefulness and implementation of Agile methods when using EVM to manage major defense programs during the Engineering and Manufacturing Development (EMD) phase.

Applicable Agile EVM points
First, a recap of John Rusk’s points that apply to major programs:

- We need overall targets for scope, time and cost so that we can track our progress towards them.
- The EV for a task accrues when the task is completed. You get no points for partial completion of a task.
- The burndown chart works best when it covers a period of time that is big enough to feel like the big picture. Cover roughly 3 to 6 months...the whole set of iterations that make up one release
- You will find unforeseen nuances of the previously-identified requirements. These are “derived requirements”—essential to the implementation but not explicit in the original user-facing requirements. Simply implement these as they arise, without reflecting them at all in the EVM system (PMB).

Regarding the last point, during software design and development, features are often defined that support a configuration-managed requirement but do not change the contract scope. For example, if there is a requirement to inform the pilot if an enemy threat has been detected, the customer and contractor may later agree that the threat shall be communicated to a pilot by a flashing light on the display console and/or by an audio signal. The subsequent customer agreement to that feature is normally not a change to the contractual statement of work (SOW) or reason to change the budget.

Points not applicable to major programs
Next, John Rusk’s assumptions and points that are not applicable to major defense programs follow:

- Agile projects are designed to have a linear output over time producing roughly the same amount of output in each iteration; therefore, we don’t need to compute an s-curve to draw our PMB line.
- Only working software features earn value. There is no earned value associated simply with designing something—you only score points when it is designed, built and tested.

During EMD of Major Defense Acquisition Programs (MDAPs) or Major Automated Information System (MAIS) programs, neither the time-phased budget nor the planned physical outputs are linear. The buildup of resources and the mix and timing of measurable output always result in an s-curve.

Although the development of the product baseline does not produce working software, Agile methods should be considered for work products such as validated requirements or features in the technical baseline. The tasks and work products that support the completion of the technical baseline require continual customer approval, long before there are software builds and tests. The phases and work products of an EMD contract for a major program follow.

EMD Phases, Milestones and Baselines
EMD has two phases; Integrated System Design (Design) and System Capability and Manufacturing Process Demonstration
During Design, the technical baseline evolves from the Capability Development Document (CDD) to the Functional Baseline to the Allocated Baseline, and finally, to the Product Baseline.

Table 1 shows the technical baselines, technical reviews, and completion criteria per Operation of the Defense Acquisition System (DODI 5000.02) and Defense Acquisition Program Support Methodology (DAPS).

A list of typical work products that support the product baseline, listed by CMMI process area, is shown in Table 2 shown on following page.

During Design, there is frequent customer interaction until the product baseline is approved. Agile methods may be useful during completion of work products that lead to the approved baseline. For example, the backlog may include the completion of trade studies, test cases, operational scenarios, and validated requirements that require customer involvement and approval. The Agile product backlog may be organized and prioritized by configuration item (CI). The output of the monthly iteration will be the scheduled subset of the CI’s, system functionality, interfaces, or complete hardware and software detailed design.

During Demonstration, the “working software” is produced. However, it is not delivered to the customer for operational use as is done commercially. That will happen after EMD is concluded.

**Table 1:** Technical baselines, technical reviews, and completion criteria

<table>
<thead>
<tr>
<th>Document/Baseline</th>
<th>DoD Source</th>
<th>Description</th>
<th>Review Milestone Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capabilities Development Document (CDD)</td>
<td>DODI 5000.02 (a) performance parameters necessary to complete design of the system</td>
<td>Detailed operational</td>
<td>Technology Development, Milestone B</td>
</tr>
<tr>
<td>Functional Baseline</td>
<td>DAPS Methodology V2.0 (DAPS) (b), 4.3.1.C35</td>
<td>System functional requirements as captured in system specifications (functional baseline). All required system performance is fully decomposed and defined in the functional baseline.</td>
<td>System Functional Review (SFR): System's lower-level performance requirements are fully defined and consistent with the mature system concept and lower-level systems requirements trace to top-level system performance and the CDD.</td>
</tr>
<tr>
<td>Allocated Baseline</td>
<td>DODI 5000.02</td>
<td>Allocated baseline (to hardware, software, human/support systems).</td>
<td>Preliminary Design Review (PDR)</td>
</tr>
<tr>
<td>Allocated Baseline</td>
<td>DAPS, 4.3.1.C38, 4.3.1.Q64</td>
<td>PDR</td>
<td>Assesses the system preliminary design as captured in performance specifications for each configuration item (CI) in the system (allocated baseline), and ensures that each function in the functional baseline has been allocated to one or more system configuration items. The software functionality in the approved allocated baseline is consistent with the updated software metrics and resource-loaded program schedule.</td>
</tr>
<tr>
<td>Product Baseline</td>
<td>DODI 5000.02</td>
<td>Defines system functionality and interfaces; and complete hardware and software detailed design for each CI.</td>
<td>Critical Design Review (CDR): Completion of Integrated System Design Phase: 1. System functionality and interfaces; complete hardware and software detailed design are defined. 2. Product baseline for all CIs is established.</td>
</tr>
<tr>
<td>Product Baseline</td>
<td>DAPS, 4.3.1.C41, 3.2.2.Q76</td>
<td>CDR</td>
<td>Assesses the system final design as captured in product specifications for each CI in the system (product baseline), and ensures that each product in the product baseline has been captured in the detailed design documentation. The software functionality in the approved product baseline is consistent with the updated software metrics and resource-loaded program schedule.</td>
</tr>
<tr>
<td>Product Baseline</td>
<td>DAPS, 4.2.2.C18, 4.3.1.C49</td>
<td>System Verification Review (SVR): Assess the system final product, as evidenced in its production configuration, and to determine if it meets the functional requirements (derived from the CDD and draft Capability Production Document) documented in the Functional, Allocated, and Product Baselines. All system performance specification qualification test requirements have been successfully completed.</td>
<td></td>
</tr>
</tbody>
</table>

(a) Operation of the Defense Acquisition System (DODI 5000.02)
(b) Defense Acquisition Program Support Methodology (DAPS)
near term planning and rewards. Some of the characteristics of Agile methods and EVM and a discussion of their differences follow.

Agile methods have the following characteristics, per Rusk’s article:

- Next iteration of work is detail planned in work package
- Product burndown is a planning package for remaining features
- Features often deferred from the current iteration to the product burndown
- Features and priorities frequently revised

The EVMS Standard includes four principles regarding planning, performance measurement, and variance analysis:

- Plan all work scope for the program to completion
- Integrate program work scope, schedule, and cost objectives into a performance measurement baseline plan against which accomplishments may be measured
- Objectively assess accomplishments at the work performance level
- Analyze significant variances from the plan, forecast impacts, and prepare an estimate at completion based on performance to date and work to be performed

One EVMS Guideline requires maintaining the PMB, the time-phased scope, schedule, and associated budget through the end of the contract. Agile’s focus on meeting near-term customer priorities may lead to a loss of focus on progress towards the next major technical review or software build. During development of the functional, allocated and product baselines, the team may fail to track progress towards meeting the success criteria for the SFR, PDR, and CDR. During Demonstration, the continual reprioritization and revision of the backlog may blur vision of progress towards meeting all the requirements in the baselined blocks and builds. By placing the remaining PBI in a planning package, the team may fail to establish sufficient, interim milestones and fail to perform variance analysis of the impact of schedule and cost variances on downstream tasks and block releases.

In the next sections, methods and examples will describe how to fit EVM to Agile methods while developing working software. The same concepts and techniques may also be applied to the Design phase.

### Working Software

Working software is produced during Demonstration. Also, during Demonstration, the product baseline may also be changed if new customer requirements emerge or if there are tradeoffs to balance cost, schedule, and technical objectives. The following example illustrates how to apply Agile methods with EVM when there is a plan to build a block of software with incremental builds.

### Define Baselines for each Build (Technical, Schedule, Cost)

Once the Product Baseline is approved, establish the schedule in the IMS and the cost baseline in the EVM database. Finally, revise the backlog and burndown or burnup curves to

<table>
<thead>
<tr>
<th>EMD Phase</th>
<th>Review Milestone</th>
<th>Typical Work Products by CMMI Process Area (PA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integrated</td>
<td>CDR</td>
<td>- Decision Analysis and Resolution PA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Trade studies</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Requirements Development PA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Requirements for verification/validation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Test cases and expected results</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Derived requirements</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Product and product-component requirements</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Interface requirements</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Interface requirements</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Product component operational concepts, scenarios and environments</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Technical Performance measures (TPM)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Validated requirements</td>
</tr>
<tr>
<td></td>
<td>SVR</td>
<td>- Technical Solution PA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Product component operational concepts and scenarios</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Technical data package</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Product-component descriptions</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Key product characteristics</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Required physical characteristics and constraints</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Interface requirements</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Material requirements</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Verification criteria to achieve requirements</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Environments and operating/usage scenarios</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Comprehensive interface</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Interface specs.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Interface control documents</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Interface design and documents</td>
</tr>
</tbody>
</table>

Table 2: Typical Work Products per CMMI Process Area
be consistent with the IMS and EVM plans.

When planning incremental builds, allocate the functional requirements in the Product Baseline to each build in each block. Document each build’s technical baseline. Apply budget to each block and build relative to its estimated software development effort. Then allocate budget to each requirement in the backlog. The budget may be equally distributed to the requirements or it may be allocated relative to the estimated software development effort to meet the requirement. The requirement’s business value may also be the allocation basis.

The burndown curve, including all work and planning packages, is a basis of the PMB. Before locking down the PMB, ensure that resources will be available when needed including software engineers with the functional expertise needed to develop and test the scheduled requirements.

Now the first monthly iteration can be planned and executed. Although all requirements in the product baseline must be achieved, one of the early team decisions will be to prioritize the builds in each block, the requirements within the builds, and finally, the content of the next iteration.

**Agile EV Example**

*Note: The following discussions of Agile methods and deferred functionality are extracted from the book, Performance-Based Earned Value®, by Ralph Young and myself. The examples of deferred functionality are from the Proceedings of the 2009 Systems and Software Technology Conference (SSTC).*

EV is often reported as a percent complete of its underlying tasks and work products (base measures). Per A. Cockburn, when selecting a measure upon which to base earned value, the best results are achieved when the measure is directly related to indicating that the desired functionality has been implemented. The product requirements are an excellent measure for use in determining earned value measures since they are directly related to evaluating progress in implementing the functionality required by the system. Each of the functional requirements is decomposed to a set of lower level, derived requirements. The set of higher and lower level requirements facilitate software design. The coded software implements the software design.

Cockburn also states that agile project teams measure progress not according to how many requirements have been gathered but by how much running functionality has been designed, programmed, and implemented (features that run). He also recommends that other units of accomplishment, besides features that run, be used to measure progress. These include use cases, individual steps in use cases, user interface widgets (frames, pull-down lists, buttons), interface calls used by applications, and user documentation.

In order to utilize requirements as the basis for taking earned value, the developer must have a requirements traceability system that provides the capability to track requirements from the level of the system requirements through software requirements, builds, Computer Software Configuration Items (CSCI), design, code and unit test, and to test procedures for all test phases.

**Deferred Functionality**

For valid reporting of project status, earned value should reflect the results of deferring functionality from its baselined iteration, build, or block.

When functionality is deferred from the current iteration to the backlog, even if it has customer concurrence, the deferral has the following major impacts:

1. If all the requirements planned for iteration are not completed, then the earned value for the deferred requirements cannot be earned as part of the iteration. It is behind schedule.
2. The work package which receives the deferred requirements will require additional resources to complete. The unearned budget is transferred to the new work package.
3. Although requirements are deferred to a new work package, the cumulative earned value must continue to show a behind schedule condition. To maintain the schedule variance, schedule the deferred effort in the first month of the new work package.

To illustrate how deferred functionality should be quantified at the work package level, assume that the plan is to develop multiple builds with incremental functionality. The functional requirements are allocated to each build and documented in the requirements traceability matrix. Each build has a separate work package for implementation of code. The completion criteria for each work package include:

- All baseline requirements have been coded, unit tested, and integrated into the build.
- The build has passed peer and customer review
- Documentation for the build has been completed
- The build has been recorded as complete in the configuration management process
The build is released for higher level integration and test.

### SOW: Software Requirements in 2 Builds:

<table>
<thead>
<tr>
<th>Build</th>
<th>Allocated TRs</th>
<th>Budget/TR</th>
<th>Bac</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>100</td>
<td>5</td>
<td>500</td>
</tr>
<tr>
<td>B</td>
<td>60</td>
<td>5</td>
<td>300</td>
</tr>
</tbody>
</table>

**Figure 1: Allocated Testable Requirements**

In this example, assume that there are two builds. The allocated testable requirements (TR) and Budget at Completion (BAC) are shown in Figure 1. The control account for Builds A and B is shown in Table 3.

<table>
<thead>
<tr>
<th>Build</th>
<th>Jan</th>
<th>Feb</th>
<th>Mar</th>
<th>Apr</th>
<th>May</th>
<th>Jun</th>
<th>Jul</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Planned Reqs. met</td>
<td>25</td>
<td>25</td>
<td>25</td>
<td>25</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Budget/Req.: 5 hours</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>500</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BCWS current (cur)</td>
<td>125</td>
<td>125</td>
<td>125</td>
<td>125</td>
<td>500</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BCWS cumulative (cum)</td>
<td>125</td>
<td>250</td>
<td>375</td>
<td>500</td>
<td>500</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Build B |     |     |     |     |     |     |     |       |
| Planned Reqs. Met |     |     |     |     |     |     |     |       |
| BCWS cur |     |     |     |     | 100 |

**Table 3: Control Account for Builds A and B**

Assume that Build A is behind schedule at the end of April with only 90 requirements being met (coded, unit tested, and integrated). At this point, earned value would be 450 hours. The schedule variance (SV) is –50 hours. The April status of Build A is shown in Table 4.

There is a decision to release Build A short of its targeted functionality and baselined requirements. There will be no additional work on Build A subsequent to its release. The requirements that have not been met are deferred into Build B. To report earned value status, close the Build A work package. Open a new work package for the next iteration, Build B, and transfer the deferred requirements (10) and budget (50) to the Build B work package.

Place the budget in the first month of the Build B work package to preserve the schedule variance. Table 5 illustrates the results of the transfer at the beginning of May.

**Table 5: Control Account Status – Beginning of May**

The earned value status at May month end is shown in Table 6. Only 20 requirements were completed in May. So there is still a schedule variance (SV) of – 50 (10 requirements).

**Table 6: Control Account Status – End of May**

Please note that the aggregate PMB (Build A plus Build B) was maintained despite the deferred functionality. During variance analysis, the team should discuss the impact of the behind schedule condition on revised backlog, including the estimated slip, if any, to subsequent releases (of builds and blocks).
Negative EV

Sometimes earned value has been taken for completing a PBI but that item is later returned to the backlog for rework. It is no longer acceptable by the customer. This may result from subsequent tests or other analysis. In order to show true progress against the cumulative plan and towards meeting final objectives, negative earned value should be taken.

Negative earned value is appropriate for accurate status reporting. EVMS Guideline 30 states “Control retroactive changes to …work performed…Adjustments should only be made..to improve the accuracy of performance measurement data.” Clearly, failure to make such adjustments will overstate true technical progress, reported earned value, and the Cost Performance Index.

What About Revised Scope?

In the previous example, there was no change to the Product Baseline or scope, even if the PBIs were reprioritized. However, at the month end review meeting, such as a Sprint review, the customer may decide to revise the backlog by adding or subtracting requirements or features.

If the Product Baseline is revised, there may be cause to revise the contract SOW and program budget. In this case, there will be a contract change that will flow down to a revised PMB.

More frequently, there is only a revision to the derived requirements and features in order to meet cost and schedule objectives. Many features and PBIs are derived requirements. They are derived from higher level functional requirements that are unchanged. After customer approval, the inclusion of additional or fewer features in the backlog may change the estimated completion dates of the remaining tasks, builds, and blocks and the Estimate at Completion. However, there is normally no justification to change the PMB unless the contract SOW and approved Product Baseline are changed.

Summary

The use of Agile methods on a major defense acquisition program may result in earlier completion of the technical baselines and implementation of the Product Baseline. However, the Agile user’s focus on the results of monthly iterations may cloud knowledge of performance towards meeting the program’s technical, schedule, and cost objectives.

Earned value discipline and continual focus on the technical baseline must be maintained.

Agile methods enable a quick response to performance deviations and changing priorities. However, the PMB should be maintained, earned value should reflect true technical performance (after accounting for deferred functionality and rework), and the program should continuously monitor the impact of variances on final cost and schedule objectives.

Note: The author realizes that this topic is relatively new and that programs and organizations may develop best practices for integrating Agile methods with EVM. Please send your practices to me for inclusion in a possible future article.

Endnotes


iiiCockburn, A. A Governance Model for Incremental, Concurrent, or Agile Projects. CrossTalk (February 2006):13-17

About the Author

Paul Solomon, PMP is the co-author of the book, Performance-Based Earned Value®. He is internationally recognized as a leader, teacher, and consultant on Earned Value Management (EVM). He published many articles on EVM, systems engineering, software engineering, and risk management. Most are available on his website. He retired from Northrop Grumman Corporation where he led the use of EVM on programs including the B-2 Stealth Bomber, Global Hawk, and F-35 Joint Strike Fighter. He has taught thousands of professionals and led EVM implementation, compliance reviews, Integrated Baseline Reviews, independent assessment reviews, and process improvement teams. He is qualified to lead EVMS certification reviews.

Author Contact Information

Email: Paul Solomon [paul.solomon@pb-ev.com]
The DACS Gold Practice Initiative:

- Promotes effective selection/use of software acquisition & development practices
- Defines essential activities/benefits of each practice
- Considers the environment in which each practice is used
- Addresses the timeliness of practice benefits
- Recognizes interrelationships between practices that influence success or failure
- Contains quantitative and qualitative information
- A continually evolving resource for the DoD, Government, Industry and Academia
- Free to use/free to join

Learn More About the DACS Gold Practice Initiative:
http://www.goldpractices.com

Current Gold Practices:

- Acquisition Process Improvement
- Architecture-First Approach
- Assess Reuse Risks and Costs
- Binary Quality Gates at the Inch-Pebble Level
- Commercial Specifications and Standards/Open Systems
- Develop and Maintain a Life Cycle Business Case
- Ensure Interoperability
- Formal Inspections
- Formal Risk Management
- Goal-Question-Metric Approach
- Integrated product and Process Development
- Metrics-Based Scheduling
- Model-Based Testing
- Plan for Technology Insertion
- Requirements Management
- Requirements Trade-Off/Negotiations
- Statistical Process Control
- Track Earned Value

Data & Analysis Center for Software
P.O. Box 1400
Rome, NY 13442-1400
http://www.thedacs.com
Common Risk Patterns

THE RISKS THAT WE SEE MOST OFTEN TODAY DURING RISK ASSESSMENTS REVOLVE AROUND A FAILURE TO TAKE THE TIME AND ALLOCATE THE RESOURCES TO GENERATE WHAT MOST PEOPLE WOULD ARGUE IS AN ACCEPTABLE SOLUTION.

by Donald J. Reifer. Reifer Consultants, Inc.

This paper identifies risk patterns that reoccurred over the years as the author conducted risk assessments. These patterns lead to fifteen actions that practitioners should perform during the course of their projects to ward off the evil effects of current and future risks that seem to plague large software developments.

During the past three decades, I served as a member of several greybeard teams whose task was to perform independent risk assessments primarily focusing on projects in trouble. My job as a SME (Subject Matter Expert) was to look over the program and determine root causes of problems. The interesting conclusion was that the risks observed had a lot of commonality. Changes occurred in our risk list, but typically over decades. Problems seem to be held in common across the defense industry.

I then started to look for risk patterns. To do this, I first analyzed the results generated by the assessments that I participated in by decade. This summary is illustrated in Table 1 in what the team considered “greatest impact order.” As a next step, I reviewed the results to determine where the variation was and what the symptoms of the problems were. Finally, I looked at root causes to determine what factors were driving the risks and what we could do to correct them.

The two most pressing problems that appear in the Table revolve with the failure of management to allocate enough time and budget for systems and software engineering tasks on projects. These two problems cross decades and appear pervasive. While such occurrences are not surprising when you think about them, they point to the fact that the defense industry at large still does a terrible job of estimating the resources needed to do the engineering job, whether well-defined or otherwise. The change of emphasis from software to systems engineering as the major risk is also noteworthy to point out as is the move from emphasis on requirements issues to focus on getting testing ready early. Perhaps, this tendency tracks the move in technology from one focused on centralized systems to those that are more distributed and net-ready. I will have more to say about this trend and its impact later in the paper.

Common Risks

When I dig deeper, I find that the risks that we see most often today during risk assessments revolve around a failure to take the time and allocate the resources to generate what most people would argue is an acceptable solution. This more complete list of risks spawned from risk assessments over the past forty years also in “greatest impact order” is summarized in Table 2 shown on the following page.

<table>
<thead>
<tr>
<th>Decade</th>
<th>Number of Assessments</th>
<th>The Perceived Problem</th>
<th>Primary Risks (Top3)</th>
</tr>
</thead>
</table>
| 1970's | 11                    | Software Discipline   | 1. Poor requirements (predominately software)  
2. Impossible schedules and inadequate budgets  
3. Lack of a software engineering discipline |
| 1980's | 21                    | Software Discipline   | 1. Poor requirements (predominately software)  
2. Impossible schedules and inadequate budgets  
3. Lack of an engineering discipline |
| 1990's | 29                    | Software Discipline   | 1. Poor requirements (both systems and software)  
2. Lack of teamwork (with systems and others)  
3. Impossible schedules and inadequate budgets |
| 2000's | 31                    | Systems Engineering   | 1. Poor systems engineering  
2. Failure to pay attention to testing early enough  
3. Impossible schedules and inadequate budgets |

Table 1 – Summary of Risk Assessment Results

Past Issues

As illustrated in Tables 1 and 2, the risks that were prevalent just a decade ago have been replaced with others that seem more pervasive. In the past, software issues dominated and were viewed as having the most impact. We kept hearing from our managers that we had a software crisis. Today, systems engineering issues seem to dominate the list. This trend
tracks the fact that the systems and software engineering disciplines, while still very different in scope and content, are moving closer together. This is driven by the fact that the defense industry is moving from a systems focus to one dominated by systems of systems. In order to understand this transformation better, let’s look at these trends a little more closely.

Because of the potential negative impacts, the defense industry launched efforts in the late 1980’s and 1990’s to address what was labeled the “software crises.” The mantra was that by focusing on process improvement and disciplined engineering methods, firms competing in the industry could be better able to make major improvement in the manner in which they developed large and complex software-intensive systems. For example, by using the Capability Maturity Model (CMM), either the Software (SW-CMM) [1] or the version addressing Integration (CMMI) [2], major improvements were possible because these process frameworks made it possible to repeat positive outcomes. These frameworks put software managers in charge of their own destinies because they were able to predict costs, engineer solutions, manage risks and deliver products on-schedule and within prescribed budgets. Most importantly, these process initiatives enabled firms to control variation across the organization using modern techniques like statistical process control. This movement raised the maturity levels of defense firms and made them more competitive.

In the early part of this decade, defense systems continued to grow larger and more complex as net-centric warfare concepts started to become part of the defense doctrine. Engineering developments became even more difficult to get right as systems like the Future Combat System (FCS) and the Ballistic Missile Defense System (BMDS – Star Wars) began their evolution. To cope with these developments, movements were mounted by organizations like INCOSE (International Council on Systems Engineering) to develop improved systems engineering methods and to make the handover from systems to software more two-way and transparent. The hope is these improvements will help arm those firms working these larger engineering developments with the methods and tools that they need to get the job done properly.

In parallel to these developments, a counter movement was born at the engineer level. The heck with the process and the process police was the rallying cry. They constrain creativity and bog the engineers down with costly and unnecessary administrative burden. Let us instead adopt more agile approaches to engineering development. These techniques emphasize developing product before documentation. They focus on team self-management rather than project management oversight. While having many positive aspects, the leaders from agilest community have stressed what many
view as less disciplined approaches to systems development. However, the community also champions a “test first” notion. As noted in Table 2, use of this approach could have a very positive effect on large engineering developments especially when the test program can take as long or longer to develop as the system being development.

A Look to the Future

During the past six months, I have been leading a study looking at software maintenance. As part of the study, my team has identified two trends that we believe will have a profound impact on the way systems are developed, maintained and sustained in the future. The two factors that are making life more difficult for engineers in the systems of systems world are interoperability and systems assurance. While requirements for these two design considerations continue to grow seemingly unbounded, most organizations that we interviewed that were trying to satisfy them did not know how to properly specify, scope, estimate, measure, mechanize and/or verify them once they were able to stabilize them. The question that we were continuously asked was “how much interoperability (or systems assurance) is enough and how do we know when we have satisfied requirements?” Undoubtedly, the challenges associated with these two factors and probably several others related to them will appear on my risk list in the not so near future.

Conclusions

The risk lists summarized in Tables 1 and 2 provide us with insight into where to focus our energy and place our attention. If I were in charge of developing large systems of systems today, here are some things that I would try to accomplish using these risk lists for guidance at different times during the system development life cycle:

Project Startup

Besides the normal engineering and management things you would do for any project of importance (develop a Work Breakdown Structure-based task plan, define systems requirements, recruit key engineering and management staff [i.e., place a lot of attention on this task in order to avoid problems later on], etc.), I would add the following six tasks to my list of work that need to be completed early in the program:

- Get a team together to define an Operational Concept Document (OCD). As part of the document, I would develop user and usage scenarios and then use them to drive my end-to-end test thread definitions.
- Staff my test organization so that they can develop test concepts and plans. Use these to drive definition of long-lead items that the test group needs to concentrate on in order to be ready to verify and validate the system as it is incrementally developed (and it will be). Ensure that the test group addresses equipment and facility needs as part of this effort.
- Have my systems engineering organization develop a System Engineering Management Plan (SEMP). Ensure that they place emphasis on conducting trades between functional and non-functional (reliability, performance, etc.) requirements. Put in place the processes needed to ensure that requirements specifications address more than functional needs.
- Have outside specialists come in and develop an independent cost and schedule analysis. Based on their estimates, determine what can and cannot be done with the resources at hand. Realize that these resources will always be inadequate. Use the results of this analysis to set realistic expectations on the part of your sponsors. Recognize that you really have to get some engineering work done in order to come up with realizable schedules and budgets. That’s why the three previously mentioned tasks need to be completed first.
- Make sure that my risk management process is working. Put “COTS software” and “licensing” on my “Top 10” risk list and track actions being taken to reduce their negative consequences. Understand that “Open Source” software is just a form of COTS software. The difference revolves around how the license is written and the licensing restrictions.
- Put “interoperability” and “systems assurance” on my “Top 10” risk list to ensure that the engineering group addresses them early and derives answers to questions relative to how much is enough from cost versus benefit point-of-view.
- However, just working issues at the start is not enough. The mid-course and end games associated with delivery are just as important because these are when the risks show themselves and have their major impacts on your project. In response, I have also included additional things that I would do based on risks itemized in the lists provided in Tables 1 and 2 above.
Mid-Course

The five additional tasks that I would add as the engineering work was being accomplished include the following:

- Have the outside specialists come in every quarter and develop a cost- and schedule-to-complete. Using the same people you used during startup is important because they have a history with the project. They would pinpoint the issues and provide you with options as they focus in on whether or not you can get the job done on schedule with the resources that you have left in the budget.

- Place renewed emphasis on getting your test program up and running per the needed schedule especially if you are incrementally developing your software. Often, test scenarios, cases and facilities are delivered late because they become lost as the engineering team concentrates on the requirements and architecture. Putting an independent test team to work in parallel helps you maintain your focus especially as the requirements change.

- Ensure that the metrics and measures that you collect actually provide insight into problems and progress. Just having a bunch of numbers is not what you want. Just having them may provide false confidence. For example, who cares about defect rates if they don’t tell you which components are the most prone to error. In other words, they need to be the right numbers. Because of the costs involved, I would eliminate that part of the measurement program that wasn’t helping me get my job done.

- Ensure that your processes are working well, especially those used for configuration and requirements management. If they aren’t, you will surely suffer the consequences. Often, processes breakdown during development due to schedule pressures. When maintaining discipline, periodic internal process assessments are a necessity.

- Set in place a market watch activity to keep tabs of COTS and Open Source software developments and influence the marketplace (via standards, buying power, incremental funding, etc.). This activity can provide you with early warning of problems (software package migrating to market forces that differ from where you want it to go, vendor going out of business, etc.).

Transition and Turnover

Finally, here are four things that I would do based on the risk lists in the Tables during transition and turnover of the system to maintenance:

- Form a working group and start the transition planning process as early as possible during the process. Get the right people involved. Make sure that the test facilities and regression test baselines are delivered with the system understanding that as much as 75 percent of the work done during maintenance involves testing and retesting [3].

- Conduct an end-product acceptance review to ensure that everything that was promised during development to the user was delivered prior to turnover to maintenance. Often, development hands the system over the wall to the maintenance team and says “here it is, have fun.” Things that should be delivered are left out and the maintenance team has to spend their first year fixing developmental problem. This can be avoided via due diligence.

- Analyze the metrics and measures collected during development to understand what problems occurred, when and why. Based on these determinations and findings, make recommendations aimed at avoiding such problems in the future. Also, probe the findings and develop recommendations on how to improve the metrics and measurement program.

- Analyze the project and develop a set of lessons learned that can be used by others to avoid the mistakes made during execution. Create a lessons learned database that others are instructed to use when they plan projects in the future.

Of course, there are many other things that I would do to reduce risk during development. To summarize, the fifteen items that I have listed are those things that I believe should be considered based on the risk lists provided in Tables 1 and 2. These risk lists were developed based on over ninety risk assessments that were conducted over a period of about forty years. This experience helps us by providing us guidance as to what to focus on in the future.

Acknowledgements

These risk assessments were not conducted alone. Many people participated in their conduct and contributed to their outcomes. I would especially like to acknowledge the
contributions of my departed friend Mike Evans. Mike taught me a lot during these risk assessments. His influence is not forgotten.

References


About the Author

Donald J. Reifer is one of the leaders in the fields of software engineering and management. He has over forty years of progressive experience in government, industry and academia. He has led major projects and initiatives and served on expert panels and review teams. From 1993 to 1995, Mr. Reifer managed the DoD Software Initiatives Office under an Intergovernmental Personnel Act (IPA) assignment with the Defense Information Systems Agency (DISA). As part of this Senior Executive Service (SES) assignment, he served as the Director of the DoD Software Reuse Initiative (SRI) and Chief of the Ada Joint Program Office (AJPO). Previously, while with TRW, Mr. Reifer served as Deputy Program Manager for their Global Positioning Satellite efforts. While with the Aerospace Corporation, Mr. Reifer managed all of the software efforts related to the Space Transportation System (Space Shuttle). Currently, as President of Reifer Consultants, Inc. (RCI), Mr. Reifer advises executives in Fortune 500 firms worldwide in the areas of software investment and improvement strategies. He has published over one hundred software engineering and management papers and six books including his popular 6th edition of the “Software Management Tutorial (IEEE Computer Society).”

Author Contact Information

Email: Donald J. Reifer [don@reifer.com]
Donald J. Reifer, President
Reifer Consultants, Inc.
14820 N. Dragons Breath Lane
Prescott, AZ 86305

Project Management Tools

[Disclaimer: The appearance of an item in this list does not constitute DACS endorsement of the product. DACS has not evaluated these items. This list is simply a glimpse into what is available in the market today beyond the tools available from Microsoft. This is not a comprehensive list. For further details about PM tools visit the Tools section under the DACS Project Management topic area. https://www.thedacs.com/databases/url/key/5606/5655]

Microsoft Office Project, by Microsoft, is a tool (licensed for fee) for understanding and controlling project schedules and finances. It can be bundled with the Microsoft Office suite of tools or purchased separately and there are several versions available. [http://office.microsoft.com/en-us/project/HA101656381033.aspx](http://office.microsoft.com/en-us/project/HA101656381033.aspx)

activeCollab is a project management and collaboration tool that you can set up on your own server or local network. It provides a platform for planning, progress tracking and communication. [http://www.activecollab.com/](http://www.activecollab.com/)

OpenProj is an open source desktop alternative to Microsoft Project. OpenProj has equivalent functionality, a familiar user interface and even opens existing MSProject files. OpenProj is interoperable with Project, with a Gantt chart and PERT chart etc. [http://www.openproj.org/](http://www.openproj.org/)

Open Workbench is an open source desktop application that provides robust project scheduling and management functionality. It is a free and powerful alternative to Microsoft Project. [http://www.openworkbench.org/](http://www.openworkbench.org/)

Open Source Project Management Tools is a web site (blog) that maintains a directory of tools through volunteer efforts. Tools suggested by the community are evaluated by editors for relevance before being posted. Information provided includes: description, screenshots, features list, reviews and links. [http://open-source-project-management-tools.blogspot.com/](http://open-source-project-management-tools.blogspot.com/)

Continued on pg 26
Online Learning Center

Technical Training On Demand

Cost effective way for organizations to provide continuous learning opportunities for their employees

Accessible
Classes available 24/7/52
From:
• Home
• Office
• Travel

Accreditation Certification
• CEUs granted
• IACET accreditation supported

Affordable
• Pay one annual subscription fee ($475)
• Take as many classes as you want

Comprehensive
• 450+ classes
• 12,000 topics
• Programming & Web Development
• Course Catalog

Latest Technologies
• Java XML
• Oracle
• Server technologies
• .NET Framework

Flexible
• Fit course work into your schedule
• Self Paced
• Refresh Knowledge
• Study during lunch or after work

To view the catalog visit:
www.thedacs.com/training
For details call: 1.800.214.7921

Data & Analysis Center for Software (DACS)
Achieving Olympic Success on Software Projects with Scope Management

WHILE PROJECT MANAGEMENT HAS PROVIDED RIGOR AND STRUCTURE TO SUCH SOFTWARE INTENSIVE SYSTEMS PROJECTS, PROJECT MANAGEMENT ALONE IS OFTEN NOT SUFFICIENT TO BRIDGE THE GAP.

by Carol Dekkers, PMP, President, Quality Plus Technologies, Inc.

Analogous approaches from other industries often provide illumination of key concepts to benefit software intensive systems projects. The latter are some of the most complex, monolithic, and costly of all project management challenges today, not only because of technology novelty but because of the sheer amount of focused customer and developer communication that must occur. While project management has provided rigor and structure to such software intensive systems projects, project management alone is often not sufficient to bridge the gap. Customers (who can find it difficult to articulate their requirements to get the results that they need) and suppliers (who are the technical brains behind the project solution delivery) have few problems communicating internally amongst themselves, yet communication across the groups can range from a minor gap to a deep crevasse. In both Australia and Finland, researchers explored this issue and came up with remarkably similar results in approaches called southernSCOPE and northernSCOPE.

Both approaches provide relief in their jurisdictions that tackle head-on the top root causes for software intensive systems project failures. An interesting point is that while geographically diverse, Australia and Finland arrived at remarkably similar approaches, both building on known best-practices available to the software industry. These multi-disciplinary approaches can be compared to how triathlon athletes train for the big event - and this is the topic of this paper. Both northernSCOPE and southernSCOPE emphasize solid training and planning to solve more than six out of the ten top reasons for project failure. This paper presents an analogous look at how the formal scope management concepts mirror triathlon athlete performance in training and racing – and how they can bring success to your next software project.

1. Why does one need training to compete in the Olympics?

Any athlete who wants to achieve success in a multi-disciplinary sport such as the Triathlon, knows that a good deal of preparation goes into all three events and, additionally, the transition phase between each event. To achieve Olympic level success requires dedication, discipline, practice, and attention to minute detail. Triathlon is one of the newest and most difficult of all Olympic sports, and the state of one’s mental condition is as important as the physical conditioning required to become a serious competitor. Training for it presents a series of inter-disciplinary challenges because of the diversity of the three different stages (swimming, biking, and running), each of which entails different but rigorous preparation and mental stamina. Each component is as important as the next, and a serious athlete knows that mastery of all skills and the transitions between each component are critical to overall success.

2. Formalized Scope Management: Triathlon Style Coaching

At first glance it may seem that there is no comparison between the typical customer’s experience with software intensive systems and preparing to compete in a triathlon. The comparison is that, just as in a triathlon, software intensive systems development consists of three major events - preliminary requirements/RFP (request for proposal), developing the software system, and managing change. In between each is a transition: 1. Awarding the contract to a supplier, and 2. Identifying change. Additionally, at the end of the project or finish line, it is important to document lessons learned. Noting that the Standish Group’s CHAOS reports continually cite that over 1/2 of software projects fail, it is clear that customers needing to complete a successful project need a coach to train them through the events and transitions. Just as an athlete wants to accomplish a triathlon, most customers want to score a positive outcome with software intensive projects. A certified scope manager (CSM) is an ideal coach for customers with software intensive systems projects!

Software intensive programs are customer driven initiatives designed to meet a business objective. Suppliers of such systems are typically well-versed and skilled to deliver technology based solutions; but not so are the customers who must
specify early requirements and engage such solution providers. At the earliest stages of a project (before requirements are complete and before a contract is awarded), customers often demand fixed price contracts thinking that such action will guarantee adherence to budget. The only problem is that it is impossible to determine a budget before it is known what will be built. Wary customer groups want to curtail costs of their investment by demanding an upper fixed price estimate; while at the same time suppliers want to be paid for the work they perform on behalf of developing the customer driven solution to requirements yet to be defined. This situation poses a multi-disciplinary challenge to the success of technology projects. What we need is a coach to properly position the customer organization for productive participation and trust? Most mid- to large-scale systems that encompass the delivery of software are candidates for scope management “coaching”. In fact, any software project, where the customer needs to ensure that they receive the software they need, is well suited for formalized scope management. The “coach” is called a scope manager and he/she acts as a supportive, knowledgeable coach for customer team members.

2.1 Approaches to Scope Management

Two formal scope management approaches exist today:

1. SouthernSCOPE from the Victorian State Government in Australia, and
2. NorthernSCOPE™ established by the Finnish Software Measurement Association (FiSMA).

Both approaches focus on scope management as the core principle, and use best-practice approaches throughout. Such concepts include early program and project estimating; unit pricing; baselining the software size; formal change management; objective measurement; project management; communication; and the use of an experience repository for collecting lessons learned.

2.2 Definition of Success

Just as a successful triathlon is a function of rigorous training, coaching, and well planned competition support, similarly, professional and systematic scope management support throughout an initiative is necessary for successful software development programs. Systems customers (acquirers) need a support team to deliver targeted coaching and skills training during the development, and at its heart should be an enthusiastic, and experienced “scope manager”. In addition, to be successful customers must be receptive to that training and coaching. Success means that all of the component projects are finished to the satisfaction of the customer organization, thereby meeting or exceeding their specified requirements, incorporating mutually agreed upon scope changes, and finishing within the agreed upon schedule and budget.

2.3 Concepts of Scope Management

The 2003 CHAOS report by the Standish Group attributes four major factors to software project success including: Senior Management Support; Scope Management; User Involvement; and Formal Basic Requirements. Other studies support similar factors, and failed projects (67%) can become successful by focusing on these traits. Einstein says that insanity is doing the same thing over and over and expecting different results - and it follows then that repeating successful practices should result in continued success.

Both Southern and northernSCOPE™ approaches divide software intensive systems projects into phases (in triathlon terms: “Events”) as outlined in Table 1 shown on the following page.

2.4 The Scope Manager Role

An experienced and knowledgeable scope manager can be as critical to software intensive systems customers as an olympic coach is to new triathletes. Scope managers ideally possess skills including:

- Business analysis
- Formal project management
- Knowledge of measurement industry productivity databases
- Software measurement (including functional size measurement; and assessment of non-functional quality requirements)
- Understanding of northernSCOPE™ processes
- Senior advisory skills (negotiation, communication, progress reporting)

The scope manager acts as a customer advocate and is typically retained by (and paid by) the customer. The corporate pain, expense and overall frustration with prior program failure often provide solidarity. In Finland, (as throughout the world), at least five major national government departments have retained the skills of a certified scope manager since 2004. As
additional organizations discover the value to cost ratio that is provided by a qualified scope manager, the demand for Certified Scope Managers will increase. Currently all proof of concept data is anecdotal, however, several research studies are proposed to unequivocally prove the value of the concept.

3. Proven scope management “downunder”

In 1995, the Victorian State Government in Australia introduced southernSCOPE on its custom software development projects. According to the originator of the method, Mr Terry Wright (Australia), “In 2000 a study of the extent and impact of its use was undertaken and resulted in the release of a revised approach. Although the number of projects that had used the approach was small, their nature was diverse. The results were outstanding:

- All projects completed within 10% of the initial project budget
- They all had a high customer satisfaction in that the software met the intended business need
- Their cost per unit was in the lowest 25% of comparative industry benchmarks.” (Wright)

Continuing in the same journal, why then is the southernSCOPE method so effective? The method successfully enabled the following problems inherent in the software engineering process to be addressed:

- Realistic cost estimates are provided at project inception
- The functional requirements developed and agreed to are sound and unambiguous
- The customer is able to make objective decisions in language he understands as to what functionality should be provided within the agreed budget

3.1 The Future of Scope Management

When we examine the similarities between a triathlon training program, the athletes involved, and the importance of a dedicated coaching staff, one can easily see how the formal scope management can provide preclusions to a successful finish for software intensive systems. For customers involved in software development programs, their success too depends on ultimate trust in the scope manager, and a focus on the following aspects of preparation: understanding of the core business and program objectives; subdivision of program components into discrete projects; ability to articulate and explore preliminary functional and non-functional requirements; clarity of purpose to prepare RFP’s for supplier

<table>
<thead>
<tr>
<th>Phase</th>
<th>Characteristics</th>
<th>Skills Needed</th>
<th>Hazards</th>
<th>Keys to success in Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-contract award (several weeks to several years)</td>
<td>Feasibility study; preliminary requirements; Request for Proposal (RFP) preparation</td>
<td>Understanding the different types of development work; knowledge of best practices of requirements specification (functional and non-functional) and software sizing (functional size measurement) skills</td>
<td>Monolithic development programs with hybrid mixtures of unknown requirements; incomplete RFP’s with ambiguous or missing requirements; overzealous suppliers; demand for fixed price based on imperfect and incomplete requirements</td>
<td>Early identification of need for and engagement of skilled scope manager; division of acquisition program into discrete projects; identification and early analysis of functional and non-functional requirements; early software size estimates; solid RFP preparation by project</td>
</tr>
<tr>
<td>Transition: Engage supplier (1-6 months)</td>
<td>Receive, open, evaluate and compare RFP responses; supplier interview, negotiation and selection</td>
<td>Analysis, and negotiating skills; knowledge of contract and unit pricing models</td>
<td>Inconsistent proposals; selection of inappropriate pricing model; incomplete or unreasonable RFP responses</td>
<td>Trust and understanding established with chosen supplier(s)</td>
</tr>
<tr>
<td>Requirements specification (1-6 months)</td>
<td>Formalize and articulate complete set of functional and non-functional requirements for each project</td>
<td>Understanding requirements types and levels; excellent documentation, writing, and reviewing skills</td>
<td>Misunderstanding or impatience regarding the definition of requirements types and levels; scope creep and gold-plating based on supplier needs</td>
<td>Complete set of product requirements; Baseline size for functional user requirements of each subsystem and project</td>
</tr>
<tr>
<td>Transition: Sign contract(s) and formally baseline project plans (days to months)</td>
<td>Contract(s) with suppliers signed for construction</td>
<td>Contract and legal experience; risk management skills</td>
<td>Overly complex or obscure legal agreements; incorrect pricing models; lack of legal advice; emphasis on sanctions instead of scope</td>
<td>Reality check validation for tendered prices (based on industry $/Function Point (FP) whenever applicable)</td>
</tr>
<tr>
<td>Software intensive system construction (6 months to 3 years)</td>
<td>Systems and software design, construction, testing and pre-installation. Program closure occurs after delivery of all subsystems; Results are evaluated and experience data collected and recorded</td>
<td>Knowledge of customer business and objectives of program and subsystems; detailed eye to gauge progress and identify issues; project management skills; ability to formulate changes and identify errors; measurement skills.</td>
<td>Uncontrolled scope creep; lack of senior management commitment; ad hoc or informal change management; lack of quality control; artificially imposed deadlines; risk ignorance</td>
<td>Measured and managed formal change control; earned value reporting and monitoring by subsystem (based on FP as applicable); progress monitoring; post-delivery payment based on delivered product size (unit pricing); data collection in an experience repository</td>
</tr>
</tbody>
</table>

Table 1: Scope Management Program Phases (and Pre-requisite Customer Skills)
unit pricing responses; ability to objectively evaluate and select most suitable supplier(s); stamina to remain committed to the program despite false starts; devotion to participate and review aspects of the ongoing program development; commitment to scope management processes; functional size measurement; earned value / function point progress reporting; formal change management; program closure and lessons learned review and identification. The jury is in based on limited results already in Finland that reinforce the success of the northernSCOPE™ concepts: projects managed formally with northernSCOPE™ principles are remarkably on budget and on scope as attested to by the various national ministries employing a certified scope manager. Documented proof of such real-life cases are forthcoming as the projects complete in Finland in fall 2009.

In summary, a scope manager is similar to a triathlon coach in the following analogous ways:

- When requirements are vague or preliminary, this is similar to the difficulty triathlon swimmers experience in seeing the direction ahead. The scope manager provides guidance similar to a coach in a boat providing navigational advice about how to turn and whether to slow down and pace oneself in the event.

- When a supplier has been chosen based on unit pricing and the RFP response, the requirements phase goes fast but can be laden with incomplete or incorrect requirements. The scope manager in this case is like the coach driving in a vehicle alongside a group of bicycle riders with extra tires, and able to provide assistance should there be an accident during the event.

- During the main event (running for the triathlon, and the actual program development from design through to coding and final testing), the scope manager provides regular roadside reporting concerning potential hazards and challenges on the road ahead, and provides an up-to-date report on expenditure of resources compared to the progress thus far (earned value management reporting), and any upcoming changes to the course.

- At the completion of the competition, the coach and the athlete review the race components and strategize on future training goals for continued success. At the completion of the software program, the scope manager collects data and reviews with the customer team the successful delivery plus what can be better done next time.

It took almost 30 years for the Triathlon competition to be become officially recognized as an Olympic Event. This author is confident that scope management concepts will soon become accepted principles in the mainstream of our industry - and that it will take less than an Olympian 30 years to do so!

References

Dekkers, Carol and Pekka Forselius, presentation and proceedings of the International Project Management Association global congress, September 2008, Rome Italy.


About the Author

Carol Dekkers is an international technology expert and speaker who provides expertise and advice to global corporations in the areas of globalization, communication and culture, software and systems measurement, quality and process improvement. Her keynote addresses are renowned for being able to present technical information in a manner highly digestible and understandable by the non-technology masses and technology professionals alike. Ms. Dekkers is the founder and president of Quality Plus Technologies, Inc., and a partner in 4SUM Partners, both of which work with a) customer companies who want to improve their success in acquiring the software they need, and b) with supplier companies who want to improve their on-time and on-budget software delivery through professional SCOPE Management. Carol is a past president of the International Function Point Users Group (IFPUG), a technical advisor to the International Software Benchmarking Standards Group (ISBSG), and a delegate to ISO’s SC7 (Software and Systems Engineering) committee. Carol’s professional designations include Project Management Professional (PMP), Certified Management Consultant (CMC), Certified Function Point Specialist (CFPS), and Professional Engineer (P.Eng.-Canada). Carol’s international speaking experience spans more than 25 countries, where she has spoken about software measurement, global software development issues, and more recently, the value of SCOPE
Management. She is co-author of five books, two of which were released in 2008: *The IT Measurement Compendium: Benchmarking and Estimating Success with Functional Size Measurement*; and *Program Management TOOLKIT for Software and Systems Development*; and three others *Practical Software Measurement: Advice from the Experts; Practical Project Estimation, 2nd Edition;* and *Fundamental Concepts for the Software Quality Engineer, Volume 2*. She has presented in more than 25 countries and is part of the PMForum Ambassadors speakers bureau, a professional member of the National Speakers Association, and the International Federation For Professional Speakers (IFFPS).

For further information and to book Ms. Dekkers to present at your next conference or corporate event, please email her at Dekkers@qualityplustech.com or visit www.caroldekkers.com or www.qualityplustech.com.

### Project Management Tools (cont. from pg. 20)

**Open Source Project Management Tools in Java** is a tool category maintained on the Java-Source.net site. Information provided for each tool includes: a brief description of its purpose, the applicable license, and a link to the source site. [http://java-source.net/open-source/project-management](http://java-source.net/open-source/project-management)

**Lighthouse** by Artifact Software: combines project management with requirements management, change management, test management, defect tracking and issue management providing a single web-based repository for project plans, resources and artifacts. [http://www.artifactsoftware.com/](http://www.artifactsoftware.com/)

**Hosted Services**

**Git and GitHub:** Git is a fast, efficient, distributed version control system ideal for the collaborative development of software; GitHub is the easiest way to participate in that collaboration. GitHub was written for hosting public, open source and private, proprietary codes — if you use Git, GitHub is for you. [http://github.com/](http://github.com/)

**Launchpad** is a an open source code hosting and collaboration platform featuring: shared bug tracking, distributed version control, translations, package building and hosting, specification tracking, building help tools and FAQ libraries. [https://launchpad.net/](https://launchpad.net/)

**GForge®Advanced Server** is an extensible platform with an intuitive interface that ties together a huge toolset, from Source Code Management (SCM) to extremely customizable Trackers, Task Managers, Document Managers, Forums, and Mailing Lists. All of these are controlled by a centralized permission system and maintained automatically by the system. [http://gforge.org/gf/](http://gforge.org/gf/)

**Basecamp** is a web-based hosted (fee for service) project collaboration tool that focuses on project management through communication and collaboration rather than tracking. Features include: customized colors and logo, a dashboard, to-do lists, file sharing, message boards, milestones, time tracking, project overview, comments on messages. [http://basecamphq.com/](http://basecamphq.com/)
Performance Based Acquisitions – What’s Out There?
A Look at Defense Acquisition University’s Learning Assets

This article focuses on the PBA training, tools, and resources that are available to the acquisition community through a myriad of Defense Acquisition University’s (DAU) training and knowledge sharing assets.

by Pamela Gouldsberry, Leslie Deneault, and Bruce Hatlem

(Here’s a brief introduction to the concept of PBA and its importance in today’s acquisition environment.)

Performance Based Acquisitions (PBA) isn’t new. Since at least 1991, acquisition professionals have been discussing the concepts, goals, training, roles, and responsibilities that support PBA. As defined in the Federal Acquisition Regulation, PBA “means an acquisition structured around the results to be achieved as opposed to the manner by which the work is to be performed”. This type of acquisition is not a process just for contracting specialists. It is a process that applies to logisticians, quality assurance evaluators, requirements generators, systems engineers, contracting, and anyone who is assigned as a contracting officer’s representative (COR).

So why, some sixteen plus years later, does there continue to be confusion on what it is, when to use it, what are the team member’s roles and responsibilities, and what measurements do we use to achieve the intended results during contract performance, etc?

One of the answers that repeatedly emerge in many reports and studies, including GAOi, Inspector Generalii, and the Services Acquisitions Advisory Panel (SARA)iii, is lack of training and experience of the acquisition workforce. If their lack of training isn’t bad enough, many people who are significantly involved in PBAs are not in the acquisition workforce and receive absolutely no training on this topic.

So, what’s out there? This article focuses on the PBA training, tools, and resources that are available to the acquisition community through a myriad of Defense Acquisition University’s (DAU) training and knowledge sharing assets. These resources are especially critical at a time of shrinking budgets, increased complex acquisitions, and limited number of experienced PBA professionals.

To assist the acquisition community, DAU has adopted a Performance Learning Model (PLM) - http://www.dau.mil/plm/plm.asp to promote career-long learning at the point of need. This model depicts quality assets such as continuous learning modules, tools, classroom training, performance support, and knowledge sharing systems.

Acquisition, Technology, and Logistics Performance Learning Model

So, let’s take a look at these learning assets…

On-Line Resources

DAU offers on-line self-paced classes, free to anyone and available 24/7. You can take the PBA related classes listed below for credit (which means that you get a certificate of completion and continuous learning points); OR you can just browse through the course and learn something new. These on-line courses come in two types: Continuous Learning (CL) modules of 2-6 hours estimated effort, and Distance Learning (DL) courses of 30-60 hours of effort. DL courses are part of DAWIA (Defense Acquisition Workforce Improvement Act) certification requirements.
CLC 013 (Contracting) Performance Based Services Acquisition: reviews all elements of a performance based service.

CLC 106 (Contracting) COR with a Mission Focus: provides the basic skills needed to be a Contracting Officer’s Representative (COR).

CLM 013 (Program Management) Work Breakdown Structure: addresses how to prepare a work breakdown structure when analyzing work to be put on contract.

CLL 011 (Logistics) Performance Based Logistics: provides an overview of the process for developing a performance based sustainment strategy.

CLL 015 (Logistics) Business Case Analysis: presents the basic considerations and objectives for conducting a BCA.

LOG235A (DL) Performance Based Logistics: provides in-depth introduction of the 12 Step process for developing a Performance Based sustainment strategy.

In addition, there are over 150 other acquisitions CLMs covering various topics in these functional areas:
- Acquisition management
- Business
- Contracting
- Engineering and technology
- Logistics
- Program management
- Harvard ManageMentor Plus

A complete list of these CLMs can be accessed at http://clc.dau.mil/.

AT&L Knowledge Management System

DAU provides access to on-line acquisition resources and training materials via the AT&L Knowledge Management System (AKMS). The AKMS consists of the AT&L Knowledge Sharing System (AKSS), the Acquisition Community Connection (ACC), and a virtual library. The AKMS is available 24/7 to fully support the AT&L workforce at the point of need. Visit these repositories and resources at http://www.dau.mil/basedocs/knowledgesharing.asp.

The Acquisition Community Connection (ACC) consists of 13 Communities of Practice (CoP), 24 Special Interest Areas, and Workspaces. Through partnerships with the Services, federal agencies, and industry, this one-stop shopping provides direct links to training, policy, guidebooks, lessons learned, templates, and other references across the federal government https://acc.dau.mil/CommunityBrowser.aspx

Within the ACC, there is a Contracting Community of Practice (CoP) – https://acc.dau.mil/con and a Logistics Management CoP – https://acc.dau.mil/log with similar PBA training, tools, and references. There is also a COR CoP – https://acc.dau.mil/cor which includes the COR Area Forum that is specifically designed to connect you with CORs performing duties in similar areas, such as contingency, construction, R & D, etc.

Contracting Officer’s Representative (COR)

Within the “Special Interest Areas” of the ACC, there is specific resources and information on Services contracting under “ACE for Services” (Acquisition Center of Excellence) with a direct link to “Performance Based Acquisition” https://acc.dau.mil/ace. This central clearinghouse for Service acquisition learning assets supports both the public and private sectors.
New DAU Services Acquisition Course

This fiscal year, DAU piloted and is offering a new course specifically addressing performance based acquisition of services. It is called ACQ 265, Mission-Focused Services Acquisition.

This multi-functional 4-day course provides members of the acquisition team the tools necessary to analyze and apply performance-based principles. The emphasis is on developing performance requirements documents and effective business strategies for contractor provided services. It is specifically designed for both contracting and non-contracting team members needing to improve their planning, executing, and performance assessing skills related to contracted services.

This course uses the 7-Step Performance-Based Acquisition (PBA) process, a team oriented approach, and several case-based activities designed to provide students with practical hands-on experience. Successful completion of this course will enable students to apply a life-cycle approach by using results-driven techniques when acquiring the acquisition of services in an integrated process team environment. The case study used for the course involves an agency-wide food service requirement valued at $1B. We start with an old, very prescriptive Statement Of Work (SOW) for food services and then walk-through the market research, strategy, contract type/incentives, risk analysis, quality assurance, performance standards, and monitoring of a new PBA Service contract for Marine Corps food service.

To apply for this course, please view “I Need Training” located on the DAU homepage http://www.dau.mil/.

Targeted Training (classes offered for a fee and if instructors are available)

DAU offers a 3-day PBSA targeted training workshop that is provided on-site of the requesting agency. The mini-course provides an overview of performance-based methods and how to determine when they are appropriate. It is designed for personnel who must work with program officials to plan, award, and administer performance-based contracts.

In addition, there is a three-day Contracting Officer’s Representative (COR) course designed to provide a full picture of what this position requires, such as duties, responsibilities, limitations, nature, and scope of personal interactions.

For more information on these courses or to request a course for your organization, visit http://www.dau.mil/performance_support/targeted_training.asp.

Performance Support

Performance Support is tailored to the customer’s needs. In addition to targeted training, other performance support activities include consulting, group facilitation, and rapid deployment training (RDT). Faculty from all disciplines and regions can consult or provide training with government acquisition organizations on either short or long term basis. For more information, visit the Performance Support/Rapid Deployment Training web site at http://www.dau.mil/performance_support/RDT.asp.

DAWIA Certification Courses

Contracting -- After a three year effort, in 2007 DAU finished rolling out the last of completely new and updated certification courses for the contracting professional. Three of the resident courses focus large portions of the curriculum on performance-based acquisition with a focus on services contracting.

CON 120 Mission Support Contracting: includes formal instruction and team exercises on FAR Part 37, Services Contracting, and the basic elements of a performance based service contract including:

- Defining a performance requirements summary
- Elements of a quality assurance surveillance plan
- Description of a performance work statement
- Identification of roles and responsibilities of the contracting officer and contracting officer representative
New DAU Services Acquisition Course

This fiscal year, DAU piloted and is offering a new course specifically addressing performance based acquisition of services. It is called ACQ 265, Mission-Focused Services Acquisition.

This multi-functional 4-day course provides members of the acquisition team the tools necessary to analyze and apply performance-based principles. The emphasis is on developing performance requirements documents and effective business strategies for contractor provided services. It is specifically designed for both contracting and non-contracting team members needing to improve their planning, executing, and performance assessing skills related to contracted services.

This course uses the 7-Step Performance-Based Acquisition (PBA) process, a team oriented approach, and several case-based activities designed to provide students with practical hands-on experience. Successful completion of this course will enable students to apply a life-cycle approach by using results-driven techniques when acquiring the acquisition of services in an integrated process team environment. The case study used for the course involves an agency-wide food service requirement valued at $1B. We start with an old, very prescriptive Statement Of Work (SOW) for food services and then walk-through the market research, strategy, contract type/incentives, risk analysis, quality assurance, performance standards, and monitoring of a new PBA Service contract for Marine Corps food service.

To apply for this course, please view “I Need Training” located on the DAU homepage http://www.dau.mil/.

Performance Support

Performance Support is tailored to the customer’s needs. In addition to targeted training, other performance support activities include consulting, group facilitation, and rapid deployment training (RDT). Faculty from all disciplines and regions can consult or provide training with government acquisition organizations on either short or long term basis. For more information, visit the Performance Support/Rapid Deployment Training web site at http://www.dau.mil/performance_support/RDT.asp

DAWIA Certification Courses

Contracting -- After a three year effort, in 2007 DAU finished rolling out the last of completely new and updated certification courses for the contracting professional. Three of the resident courses focus large portions of the curriculum on performance-based acquisition with a focus on services contracting.

CON 120 Mission Support Contracting: includes formal instruction and team exercises on FAR Part 37, Services Contracting, and the basic elements of a performance based service contract including:

- Defining a performance requirements summary
- Elements of a quality assurance surveillance plan
- Description of a performance work statement
- Identification of roles and responsibilities of the contracting officer and contracting officer representative

CON 215 Intermediate Contracting for Mission Support: is an application course where students process a service contract requirement. Performance based services acquisitions are integrated through readings, references, exercises and class discussions. Additional emphasis on the Seven Steps Guide is incorporated throughout the resident course. Specific areas addressed include:

- Establishing an integrated solutions team
- Examining private-sector and public-sector solutions
- Executing a PWS or SOO
- Developing meaningful measures
- Selecting the right contractor
- Managing performance

Targeted Training (classes offered for a fee and if instructors are available)

DAU offers a 3-day PBSA targeted training workshop that is provided on-site of the requesting agency. The mini-course provides an overview of performance-based methods and how to determine when they are appropriate. It is designed for personnel who must work with program officials to plan, award, and administer performance-based contracts.

In addition, there is a 3-day Contracting Officer’s Representative (COR) course designed to provide a full picture of what this position requires, such as duties, responsibilities, limitations, nature, and scope of personal interactions.

For more information on these courses or to request a course for your organization, visit http://www.dau.mil/performance_support/targeted_training.asp.
**CON 353** Advanced Business Solutions for Mission Support: requires the students to do critical thinking and problem solving for complex contracting issues. PBA is specifically addressed in both team and individual projects and through our module on AT&L Hot Topics.

Logistics -- In addition to the LOG235A DL course there is the one week LOG235B resident course. It provides members of weapons systems acquisition teams and other acquisition professionals the expanded tools necessary to analyze and apply performance-based principles to the development of a sustainment strategy. The emphasis is on identifying the factors that need to be addressed when defining the scope and responsibilities for implementing PBL. This course uses a case based, IPT oriented approach to provide students with practical hands-on experience. Successful completion of this course will enable students to apply a life-cycle approach by using results-driven techniques when defining a sustainment strategy to support warfighters performance requirements.

**Summary**

Formal training is only part of the answer. It must be augmented with on-the-job training. Learning and retention studies clearly support that 80% of what we learn is by “doing”. So, what can we do to gain more exposure to performance based acquisitions? Consider adding cross-functional rotational assignments, direct involvement or reassignments, or obtaining or shadowing a mentor to your Individual Development Plan (IDP). Of course, leadership support for these professional developmental opportunities cannot be over emphasized. It takes all of us -- to address these challenges.

One final note…, we can’t impress enough how many times we’ve used and appreciated samples from our co–workers. Although it may take a few extra minutes in your day, sharing your PBA Performance Work Statements (PWSs), Statement of Objectives (SOOs), and lessons learned, etc. is critical to our professional development and achieving our organizational missions. So, take time to share…both internally and externally. Either through your organizational training websites, leading a cross-functional roundtable to share lessons learned, or sharing your materials on our DAU Acquisition Community Connection, etc.

**End Notes**


iii Acquisition Advisory Panel (1423) Draft Report, Chapter 2, Improving Implementation of Performance-Based Service Acquisition (PBSA) in the Federal Government” December 2006

iv A joint industry and Government website with vetted PBSA examples and samples http://www.acqnet.gov/comp/seven_steps/index.html


---

**About the Authors:**

**Pamela Gouldsberry** is the Program Director for Contracting, Level III, at DAU. She teaches and is responsible for the curriculum development for CON 353, Advance Business Solutions for Mission Support. Ms. Gouldsberry has over 22 years of operational and managerial experience in contracting. She is Certified Professional Contract Manager (CPCM) and Level III DAWIA certified in contracting. Pamela is the Program Director for Acquisition Services at DAU. She regularly teaches DAWIA contracting classes and is responsible for the new ACQ265, Mission-Focused Services Acquisition course.

**Leslie Deneault** is Level III DAWIA certified in contracting and program management and a CPCM. She is also a retired Air Force Officer who spent most of her military career in contracting.

**Bruce Hatlem** was the Program Director for Performance Based Logistics at DAU when this article was written. He was Level 3 DAWIA certified in logistics and regularly taught DAWIA certification logistics classes. He was responsible for the LOG235 Performance Based Logistics courses and supported PBL policy activities for the DUSD (L&MR). He is a retired Marine Corps Officer who has supported multiple ACAT 1 weapon systems development program offices.

**Author Contact Information:**

Pamela Gouldsberry can be contacted at 703-805-4893 or [pamela.gouldsberry@dau.mil]

Leslie Deneault can be contacted at 703-805-4477 or [leslie.deneault@dau.mil]

Contact information is not available for Mr. Hatlem.
The STN is a theme-based quarterly journal. In the past DACS has typically solicited specific authors to participate in developing each theme, but we recognize that it is not possible for us to know about all the experts, programs, and work being done and we may be missing some important contributions. Therefore, beginning in 2009 DACS is adopting a policy of accepting articles submitted by the software professional community for consideration.

DACS will review articles and assist candidate authors in creating the final draft if the article is selected for publication. Note that DACS does not pay for articles published. Note also that submittal of an article constitutes a transfer of ownership from the author to DACS with DACS holding the copyright.

Although the STN is theme-based, we do not limit the content of the issue strictly to that theme. If you submit an article that DACS deems to be worthy of sharing with the community, DACS will find a way to get it published. However, we cannot guarantee publication within a fixed time frame in that situation. Consult the theme selection page and the Author Guidelines located on the STN web site (see https://www.softwaretechnews.com/) for further details.

To submit material (or ask questions) contact news-editor@thedacs.com

Recent and upcoming themes include:
- Earned Value
- Software Testing
- Project Management
- Model Driven Development
- Software Quality and Reliability
The first 50 people to send in a completed survey will receive a FREE DoD/IT Acronym CD from the DACS.

This valuable CD-ROM contains over 9,000 Department of Defense and Information Technology acronyms. There are hundreds of acronym lists available but none are as well done as this CD and specifically targeted towards DoD and Information Technology. This unique-shaped CD-ROM plays in your computer's regular, hub-mounted, CD drive. You'll use this great resource over and over again. It's FREE, just for filling out our brief survey on the next page!

Data & Analysis Center for Software (DACS)

http://thedacs.com
1. Which volume of the Software Tech News did you receive?  

2. When did you receive the newsletter? (month/year)

3. How satisfied were you with the CONTENT of the newsletter? (Article Quality)
   - Very Satisfied
   - Satisfied
   - Neither Satisfied nor Dissatisfied
   - Dissatisfied
   - Very Dissatisfied

4. How satisfied were you with the APPEARANCE of the newsletter?
   - Very Satisfied
   - Satisfied
   - Neither Satisfied nor Dissatisfied
   - Dissatisfied
   - Very Dissatisfied

5. How satisfied were you with the OVERALL QUALITY of the newsletter?
   - Very Satisfied
   - Satisfied
   - Neither Satisfied nor Dissatisfied
   - Dissatisfied
   - Very Dissatisfied

6. How satisfied were you with the ACCURACY of the address on the newsletter?
   - Very Satisfied
   - Satisfied
   - Neither Satisfied nor Dissatisfied
   - Dissatisfied
   - Very Dissatisfied

7. Approximately how much of the newsletter do you read?
   - The entire issue
   - Most of the content
   - About half the content
   - Briefly Skimmed
   - Didn’t Read

8. Would you read this newsletter in an E-mail newsletter format?
   - Definitely
   - Probably
   - Not Sure
   - Probably Not
   - Definitely Not

9. How did you request the product or service?
   - Phone Call
   - E-mail
   - DACS Website
   - Subscription Form
   - Other ______________________

10. Would you recommend the Software Tech News to a colleague?
    - Definitely
    - Probably
    - Not Sure
    - Probably Not
    - Definitely Not

11. What topics would you like to see this newsletter devoted to? ________________________________

    _______________________________________________________________________________________
    _______________________________________________________________________________________
    Comments (optional) _______________________________________________________________________
    _______________________________________________________________________________________

REGISTER FOR THE FIRST TIME | UPDATE CURRENT SUBSCRIPTION

Name: __________________________ Position/Title: __________________________

Organization: __________________________ Office Symbol: __________________________

Address: __________________________ City: __________________________

State: __________________ Zip: ___________ Country: __________________________

Telephone: _____ - _____ - ______ Fax: _____ - _____ - ______ Email: _______________________

Functional Role: __________________________________________________________________________

Organization Type:  
   - Air Force
   - Army
   - Navy
   - Other DoD
   - Commercial
   - Non-Profit
   - Non-US
   - US Government
   - FFR&D
   - Other ______________________

*Note: You must give us your address to receive the CD.
ABOUT THE SOFTWARE TECH NEWS

STN EDITORIAL BOARD

Ellen Walker
Managing Editor
ITT Corporation, DACS

Dan Ferens
Co-Editor
ITT Corporation, DACS

Philip King
Production Editor
ITT Corporation, DACS

Wendy Butcher
Graphic Designer
ITT Corporation, DACS

Paul Engelhart
DACS COTR
Air Force Research Lab

Morton A. Hirschberg
Editorial Board Chairman
Army Research Lab (retired)

Dr. Kenneth E. Nidiffer
Software Engineering Institute

Dennis Goldenson
Software Engineering Institute

John Scott
Mercury Federal Systems

ABOUT THIS PUBLICATION

The Software Tech News is published quarterly by the Data & Analysis Center for Software (DACS). The DACS is a DoD sponsored Information Analysis Center (IAC), administratively managed by the Defense Technical Information Center (DTIC). The DACS is technically managed by Air Force Research Laboratory, Rome, NY and operated by ITT, Advanced Engineering and Sciences Division.

ADVERTISEMENTS

The Software Tech News is now accepting advertisements for future newsletters. In addition to being seen by the thousands of people who subscribe to a paper copy, an electronic version is available at the DACS website, exposing your product, organization, or service to hundreds of thousands of additional eyes every month.

For rates and layout information contact: news-editor@thedacs.com

ARTICLE REPRODUCTION

Images and information presented in these articles may be reproduced as long as the following message is noted:

“This article was originally published in the Software Tech News, Vol.12, No.3 September 2009.”

Requests for copies of the referenced newsletter may be submitted to the following address:

Philip King, Production Editor
Data & Analysis Center for Software
P.O. Box 1400
Rome, NY 13442-1400

Phone: 800-214-7921
Fax: 315-838-7130
E-mail: news-editor@thedacs.com


In addition to this print message, we ask that you notify DACS regarding any document that references any article appearing in the Software Tech News.

COVER DESIGN

Wendy Butcher
Graphic Designer
ITT Corporation, DACS

DISTRIBUTION STATEMENT:
Unclassified and Unlimited
IN THIS ISSUE

Tech Views
by Dan Ferens, Co Editor for DACS ................................................................. 3-4

Dispelling Some Myths about Earned Value Management (EVM)
by Anita Cukr ........................................................................................................ 5-7

Agile Earned Value and the Technical Baseline
by Paul J. Solomon (Performance-Based Earned Value) ........................................ 9-14

Common Risk Patterns
by Donald J. Reifer, Reifer Consultants, Inc ..................................................... 16-20

Project Management Tools
by Ellen Walker, STN Managing Editor ............................................................... 20, 26

Scope Management
by Carol Dekkers, PMP, President, Quality Plus Technologies, Inc .................... 22-26

Performance Based Acquisitions – What’s Out There?
by Pamela Gouldsberry, Leslie Deneault, and Bruce Hatlem ................................. 27-31