Software Acquisition: Reducing Risks

James Jones

The Acquisition Risk Crisis

The GAO has shown that technical, cost, schedule, and performance risks are inherent in delivering software-intensive systems. The GAO also reported that FAA acquisition programs have consistently experienced cost, schedule, and performance problems attributed to systemic management issues. Various authorities and the GAO have concluded that there is insufficient knowledge and skill to effectively manage the life cycle of those systems where software plays a significant role. Software acquisition and development have always presented challenges due to such factors as software estimation, design constraints, acquirer lack of experience, and supplier lack of institutionalized software development capability maturity. Cost overrun is the single biggest risk because it represents time, money, and missed opportunities. The difficulty in estimating costs is attributable to inadequate software size estimates and requirements growth. Poor software size estimation is one of the main reasons major programs fail to meet deadlines. Software size is the critical factor in determining cost, schedule,

Jones *is a DoD software acquisition subject-matter expert at Support Systems Associates, Inc. He has software experience in acquisition, development, and process improvement and holds two patents (4,479,034: 4,451,702).*

and effort. Software sizing is typically driven by the contract and the supplier software development capability maturity.

My Software Acquisition Journey

My software acquisition journey includes providing advisory and assistance services and software subcontract management for major defense acquisition programs (MDAP). These MDAPs experienced cost and schedule overruns. One MDAP experienced three "Nunn-McCurdy" unit cost breaches. Prior to my serving as the software subcontract management lead, thousands of subcontract data requirement deliverables requiring approval by the prime were 6 months to 2 years overdue. Approval was achieved prior to FAA type certification.

FAA procurement comprises over 200 acquisition programs. I provided systems engineering and integration services such as system development manager, software lead, on-site support, and software subject matter expert. Several programs experienced cost and schedule overruns, performance, and terminations for convenience and default. I was deposed by the supplier. One major cornerstone of the FAA experienced cost increases from $3.6 billion to $7.6 billion and required restructuring. However, I was involved with two very successful major programs. As the software lead, the supplier delivered the first production unit 6 months ahead of schedule and received a $9 million incentive award. The other program ($1.3 billion) achieved 100% on-time delivery of all systems.

Five Key and Effective Software Acquisition Elements

Based on previous acquisition experiences, I will discuss five key and effective software acquisition elements that result in reducing risks and improving the acquisition outcomes.

- Software Contract Requirements
- The Acquisition Environment
- Technical Performance Assessment
- Software Acceptance
- Performance Measurements

1. Software Contract Requirements

The degree of interaction between the acquirer and supplier depends on the nature of the development effort and the contract type. Although there are many variations, the two basic compensation schemes are fixed-price and cost-reimbursement. To provide an effective software acquisition environment, appropriate software contract requirements must be communicated to the supplier in the request for proposal (RFP). Success of an acquisition is directly linked to the quality of the RFP. During the RFP preparation, the acquisition team must have software acquisition expertise to ensure that essential software data and data rights are acquired.

Software must be addressed in the following RFP essential elements:

- Section L and Section M
- Statement of Work (SOW)/Statement of Objectives (SOO)
- Contract Data Requirements List (CDRL)
- System Specification
- Data Rights

Section L and Section M

Section L (instructions) should instruct the following software data to be submitted: 1) draft software plans (i.e., software development plan, software configuration management plan, and software quality assurance plan), 2) description of previous software development experience of similar systems, and 3) description of the software process defined in the draft software plans. Section M (evaluation factors) should contain district discriminators for software requested in Section L. For example, the government will consider the offeror's plans for conducting the software development and capability maturity.

Statement of Work/Statement of Objective

The Statement of Work (SOW) or Statement of Objective (SOO) is the "linchpin" of the contract. It defines the tasks required to successfully supply the software. The SOW/SOO must provide sufficient detail to allow the supplier to scope the effort, cost it, and provide a technical solution.

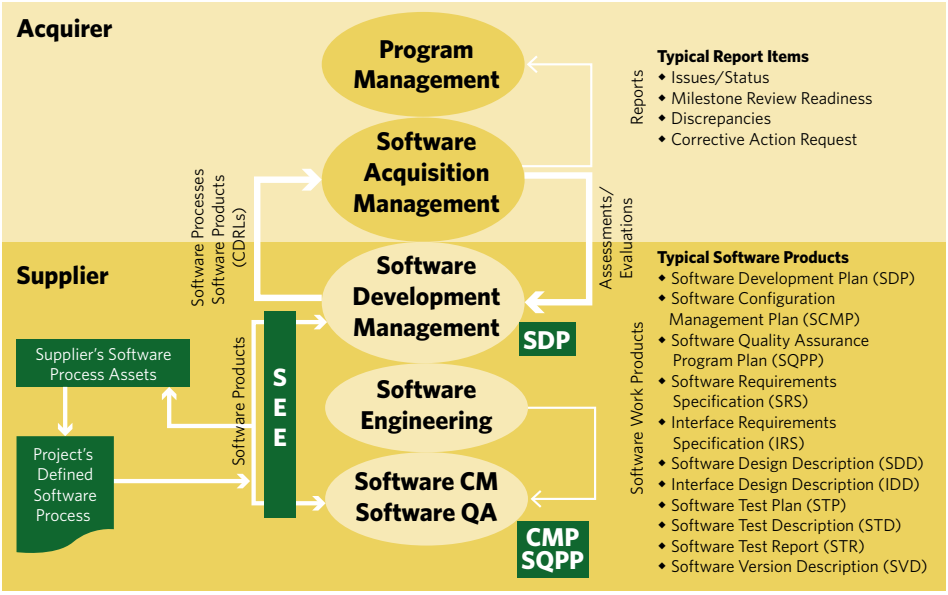
The SOW/SOO must contain software tasking information which should reference any applicable CDRL item that will be delivered by that task. Task should include updates of software plans as CDRLs, subject to acquirer review and approval. While the SOW/SOO states the specific tasks to be performed, it must not tell the supplier how to do the required work.

CDRLs

CDRLs are absolutely essential for managing the development process. Software CDRLs are a natural by-product of the development process to capture results for each software development activity. The *Defense Federal Acquisition Regulation Supplement* requires the use of a CDRL in solicitation when the contract will require delivery of technical data. CDRLs should be used only to acquire software data and rights which are essential to meeting the needs of the requiring organization. CDRLs must be referenced in the SOW paragraphs describing the software effort and preparation of software data. SOW takes precedence over the CDRL in a contract. Therefore, it is essential that the language in the SOW be consistent with and does not conflict with the CDRL in any way. Special data provisions (e.g., data rights and warranty if required) should be identified in the contract via special contract clauses.

Each CDRL identifies a data acquisition document data item description (DID). The DID defines the data the supplier is required to provide, along with delivery instruction. Assist Quick Search should be used to access the current DID. The DID selected should be used as is, or with non-applicable requirements tailored out (i.e., data requirements cannot be added to, only tailored out of a DID). Tailoring instructions (e.g., "BLK: Delete paragraphs...") are entered in the remarks section (Block 16). The DID should be referenced

Figure 1. Acquisition Environment



SDP: Software Development Plan
SEE: Software Engineering Environment
CMP: Configuration Management Plan
SQPP: Software Quality Program Plan

by the exact identifier and title with reference to any issue or revision identifier.

CDRL submission should be associated with events such as technical reviews (Quality Gates) in accordance with the CDRL item blocks. CDRLs should be delivered prior to the event to allow: 1) the acquirer sufficient time to perform a detailed review and provide review comments, 2) the supplier to disposition review comments, and 3) the acquirer and supplier to agree on the disposition. It is absolutely critical that the acquirer time for review and acceptance/rejection be specified in Block 16. All software CDRLs should be prepared by the software acquisition management team, reviewed by all applicable distribution addressee organizations and approved by either the appropriate acquirer program manager or data requirements review board chairperson prior to action by the contracting officer.

With the SOO approach, a list of CDRLs is proposed tailored to their design. The proposed CDRLs are then evaluated by the acquirer during proposal evaluation.

System Specification

The system specification is used to establish top-level technical performance, design, development, integration, and verification requirements. Software development constraints (e.g., methodology and safety-critical constraints) should be included. Sound system requirements are the backbone for accurate performance parameters—essential to the development of effective capabilities.

Data Rights

Data rights are of great importance to both the acquirer and the supplier. The acquirer must have sufficient rights to enable the use, maintenance, and replication of the software data. The supplier wants to ensure that its proprietary rights for software developed at company expense are protected in order to maintain its competitive advantage. Data rights categories include: unlimited rights, acquirer purpose rights, and restricted data rights. The secretary of the Air Force has directed the acquisition of technical data and associated rights to be addressed in all acquisition strategy plans.

2. The Acquisition Environment

Software-intensive system acquisition involves a number of organizations, including the user, the acquirer, and the supplier. Figure 1 depicts the acquirer/supplier relationships. The degree of interaction depends on the development effort and the contract type.

Acquirer

The acquirer program manager (PM) has full authority, responsibility, and resources to execute the acquisition program. The appropriate business function groups—finance, contracts, and legal—should establish and monitor the terms and conditions of the contract. During the establishment of the contract, the acquisition team must consist of a software acquisition management (SAM) integrated product team (IPT). A software acquisition manager should be designated to be responsible for all software acquisition activities and products. The manager should have software acquisition experience and coordinates with other affected parties such as the PM, contracting officer, and finance.

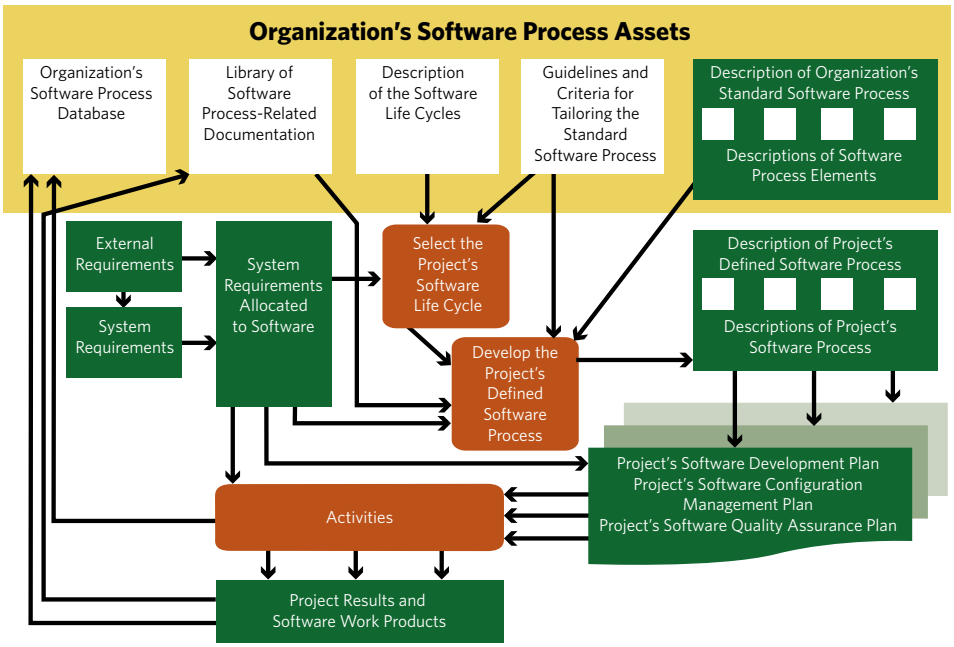
The SAM IPT must have adequate resources and funding. Members should be trained to perform the acquisition activities and receive orientation in the technical aspects of the program. The SAM IPT must recognize quality work before they can require and accept it.

Supplier

The supplier software IPT should be headed by a software manager who is responsible for planning, managing, tracking, and oversight. The manager should be the single point of contact for the acquirer software manager. The manager should provide visibility into actual progress so supplier senior management and the acquirer software manager can take effective actions when the performance deviates significantly from the plans. The software IPT should consist of a software process group, software engineers, software configuration management, and software quality assurance. The software

process group must establish and maintain a set of software process assets. Figure 2 illustrates the supplier process definition. The program software process should be developed by tailoring the organization's standard software process. A software life cycle model should be selected from among those approved by the organization to satisfy the program contract requirements and operational constraints using the guidelines established by the organization. After the program software process is established, the supplier should develop the software plans. Software plans should be updated based on events or phase-dependent.

Figure 2. Typical Software Process Definition



3. Technical Performance Assessments

Technical performance assessments enable the acquirer to determine accuracy and adequacy of the supplier process, progress, and CDRLs. It provides measurable results for determining effectiveness of the process and CDRLs quality.

The technical performance assessments are:

- Process Assessments
- Progress Assessments
- CDRL Review

Process Assessments

The acquirer should conduct process assessments to verify that software management, software configuration management, and software quality assurance activities and products are in compliance with contract requirements, the supplier process, and plans. Results should be analyzed to detect issues and to identify risks. The contract should provide a mechanism allowing the acquirer to access the supplier process and plans.

Progress Assessments

Progress assessments, using reviews, should be conducted to determine status, surface issues, and provide feedback to the supplier. The key focus should be what is done and the product being built. There are two general types of reviews: formal and informal. Formal reviews, such as technical reviews, should be defined in the contract. Informal reviews are conducted by the supplier—peer reviews and walkthroughs, for example.

Formal reviews should be structured around well defined procedures and objectives and coupled with realistic program events. Technical reviews (e.g., software requirements review) should directly support the software development process and provide the acquirer insight into the development status and CDRL quality. Technical reviews should be used as quality gates. The completion of software activity and associated CDRLs should be a prerequisite for the technical reviews. The acquirer and supplier should agree on CDRL maturity (preliminary, draft, final) and the technical review entrance/exit criteria.

CDRL Review

CDRLs are essential for managing the development process and delivery of quality software. Prior to exiting each development phase, the supplier should perform CDRL peer reviews and place the CDRLs under software configuration control prior to delivering to the acquirer. The acquirer should perform a detailed review providing review comments and/or recommendations in accordance with the acquirer CDRL review procedure. It is critical that the acquirer time for review and acceptance/rejection be specified in the CDRL.

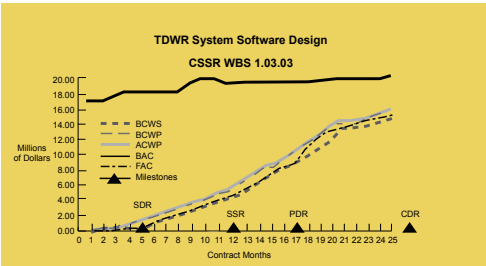
4. Software Acceptance

The development of software involves a series of production activities within which opportunities for human induced software errors are enormous. Because of this likelihood, the development process is accompanied by software testing, a quality assurance activity. There are typically three levels of software testing performed by the supplier: Unit testing, integration, and formal qualification testing (FQT). Unit and integration testing are conducted in accordance with the supplier process, and plans.

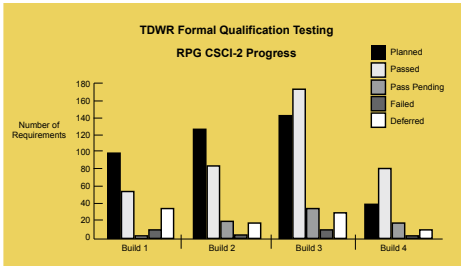
FQT is performed to evaluate the "as-built" software against the software requirements to ensure that the probability of failure due to latent defects is low enough for acceptance. FQT should be specified in the SOW with CDRLs. The supplier

Figure 3. Examples of Performance Measures

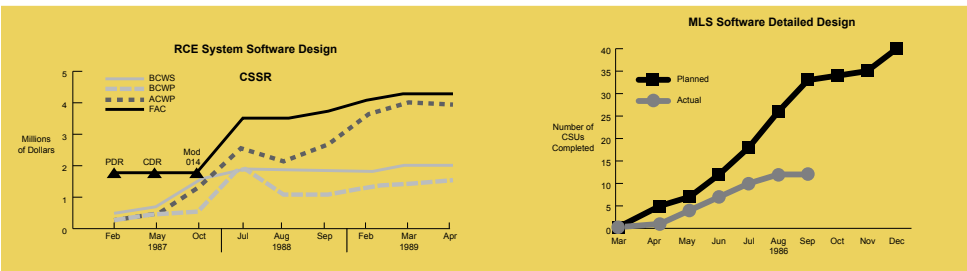
Cost/Schedule Deviation



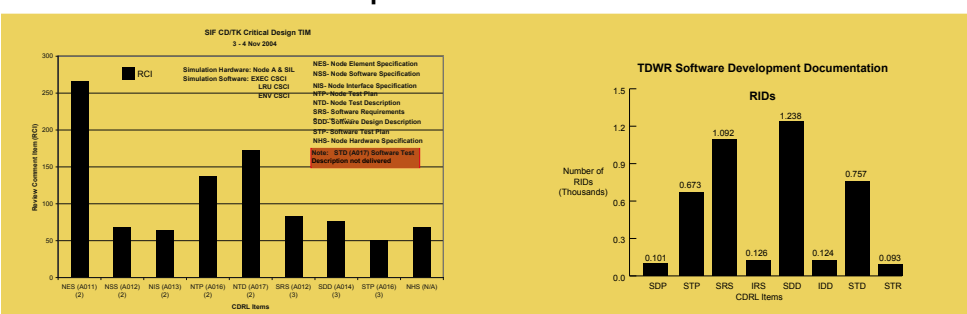
FQT Progress



Development Progress



Document Review Item Discrepancies



should document testing criteria, regression testing strategy, and acceptance criteria. Tests should be traceable to the software requirements. The supplier should document test cases and procedures.

Prior to FQT execution, the supplier should establish test readiness criteria and document the "as-built" software. The acquirer and supplier should agree to proceed to FQT execution. Upon agreement, the supplier should execute the tests in accordance with test procedures and capture the execution activity via test logs. Acquirer and supplier software quality assurance should witness all FQT execution. After FQT execution, the supplier should document test results indicating any problems detected.

Problems identified during FQT should have priority classification and should be tracked to closure. The supplier should establish a change control system to implement software changes identified during FQT, peer reviews, and approved acquirer comments. The change control system should provide how many problems have been reported, how many problems are pending, and how many problems are closed, as well as the progress of each problem.

Considering that complete test coverage is generally not possible, the acquirer and supplier face a difficult question in deciding when to release the software. The acquirer and supplier should agree on completion criteria. Prior to software acceptance, the acquirer should conduct functional and physical configuration audits to establish a product baseline.

5. Performance Measurements

Performance measurement is essential to managing and producing quality software. Software development is often out of control; you cannot control what you cannot measure. Better process management can be achieved if the attainment of cost and schedule targets and the quality of the software can be qualitatively measured. The acquirer and supplier should use performance measurements as quality indicators (metrics) to augment conventional acquisition and development reports. As mandated by Section 804 of the National Defense Acquisition Act, "metrics for performance measurement and continual process improvement" are a requirement.

The acquirer and supplier should mutually agree on and implement selected performance measurements. Examples of performance measurements are shown in Figure 3. As shown, over 4,000 review item discrepancies were identified, which contributed to FAA success.

Reducing Software Acquisition Risks

Studies have shown that technical, cost, and schedule risks are inherent in delivery of software-intensive systems. Five key and effective software acquisition elements can reduce risks, but reducing software acquisition risks requires assiduously detailing software contract requirements and applying knowledge and skill acquirer and supplier with capability maturity, effectively assessing supplier technical performance through process, progress, and CDRL review to measure effectiveness and compliance. Reducing risk also requires ensuring the "as-built" meets requirements, and determining progress toward objectives through performance measures.

The author can be contacted at **jjones@ssai.org.**