

# Software Acquisition & Supplier Management: Part 1 – Product Definition & Supplier Selection

Theresa Hunt  
Senior Consultant  
The Westfall Team  
Orlando, FL 32750  
theresahunt@earthlink.net  
[www.westfallteam.com](http://www.westfallteam.com)

Linda Westfall  
President  
The Westfall Team  
Plano, TX 75075  
lwestfall@westfallteam.com  
[www.westfallteam.com](http://www.westfallteam.com)

## SUMMARY

Today's software intensive products and business practices require companies to approach software purchases with a well-informed and strategic method. This paper discusses the first seven steps in the Software Acquisition and Supplier Management Model illustrated in figure 1, including:

- Step 1: Planning the acquisition
- Step 2: Defining the software product's requirements
- Step 3: Determining the acquisition approach
- Step 4: Identifying and evaluating potential suppliers (and their software products)
- Step 5: Defining the contract requirements
- Step 6: Selecting a supplier
- Step 7: Negotiating and awarding the contract

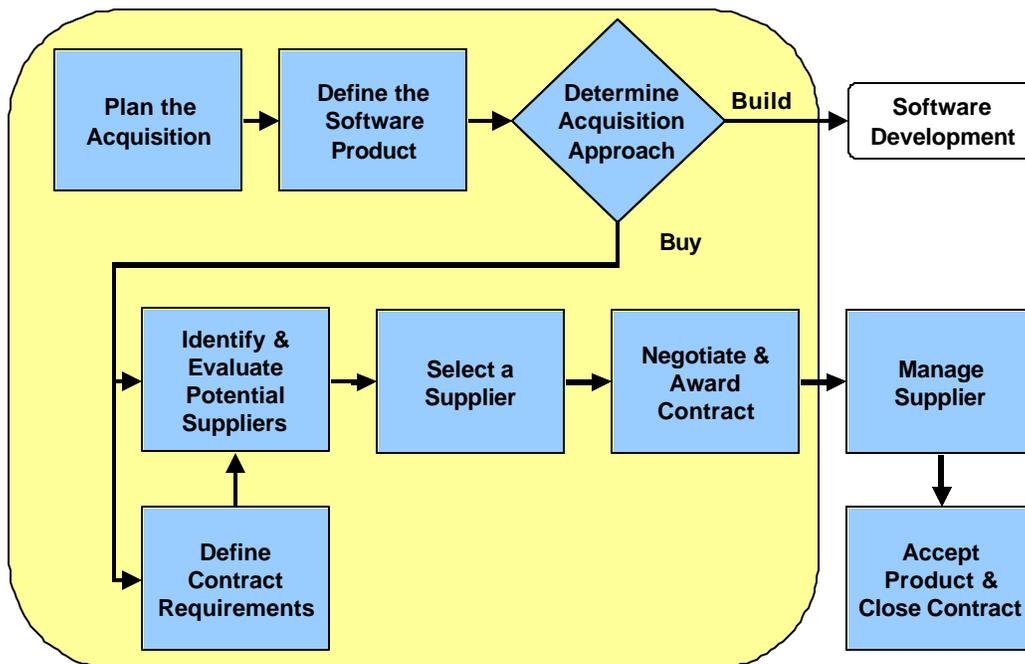


Figure 1 - Software Acquisition and Supplier Management Model

## INTRODUCTION

Before the task of purchasing commercially available software, contracting a supplier to create a custom software package, or developing a "homegrown" software application can commence, adequate planning must take place. Having a defined software acquisition and supplier management process helps insure that important steps in the acquisition process are not forgotten. Predefined goals and objectives align the business needs with the mission of the acquisition. The software

industry is now several decades old, however problems including schedule slippage, budget overruns, missing functionality and poor quality and reliability are still prevalent. These continuing issues make the need for good software acquisition and supplier management practices essential to the success of our organizations.

“Those who cannot remember the past are condemned to repeat it.” George Santayana. [Santayana-05]

The reason we keep making the same mistakes over and over is that we are not learning from our own past. We either fail to examine the past because we are afraid of what we will find, we make an assumption that the past is not relevant because “this project is different”, or we don’t have the mechanisms in place to capture the data and information we need to adequately analyze our past efforts. However, as software becomes an integral part of our businesses and our lives, we can no longer afford not to learn from our past mistakes. Having a defined software acquisition and supplier management process facilitates the propagation of lessons learned from one acquisition project to the next so we can repeat our successes and stop repeating actions that lead to problems.

## STEP 1 - PLAN THE ACQUISITION

Planning for the software acquisition begins when the idea or need is established for acquiring a software product. In this first step of the acquisition process:

- Key acquisition roles are assigned
- The business need for the software is described in terms of technical and functional needs, quality attributes, project constraints, and acceptance and completion criteria
- The acquisition plan is documented

**Assigning Key Roles:** Responsibilities for key acquisition roles must be assigned. This includes the acquisition sponsor, acquisition manager and members of the acquisition team. It also includes representatives from the customer and/or user community and other key stakeholders that will work with the acquisition team to define the business needs and software requirements.

The role of the acquisition sponsor is to provide organizational influence to help justify and sell the acquisition within the acquirer’s organization. The sponsor is responsible for:

- Communicating the vision of the proposed product to executive management
- Establishing credibility to the acquisition effort
- Obtaining approval to proceed
- Supplying a solid business case and strong support for the acquisition
- Championing the cause of the acquisition
- Ensuring ongoing funding, staffing and resources for the acquisition project

The role of the acquisition manager is to:

- Lead and manage the acquisition team
- Direct ongoing acquisition project activities
- Control cost and schedule
- Track and control the acquisition project against the plan
- Report acquisition status to the sponsor and other management

The role of each individual Acquisition Team member is to adequately represent their stakeholder organization and to perform assigned tasks.

**Describe the Business Need:** The increasingly high rate of change in new technology makes it difficult for companies to keep up with current trends and remain competitive. The current state of the company has to be continually examined to determine any need for change or update. A business need could be a problem that needs to be addressed or it could be an opportunity for the business to improve in some area. It is important to precisely specify the business need so that people will clearly understand what action is to be taken and avoid addressing the wrong need. Quantitative measures consisting of data related to the stated business need should be used to support the facts upon which the need is based. For example, the need to obtain a software configuration management tool could be supported with quantitative measures including the number of source modules, lines of code, fielded and in-process versions of the software that must be maintained. The business need also includes the description of any assumptions upon which the project is based or any constraints on the project factors such as schedule, budget, resources, software to be reused, acquired software to be incorporated, technology to be employed, and product interfaces to other products. For example, any estimates, expectations, statements or proposed solutions that lack supporting data should be considered as assumptions. Business sponsors and stakeholders indicate their buy-in and support of the business need through the approval of the business need description.

The identity of the stakeholders is not always obvious. Different stakeholders can also give conflicting direction. This is especially difficult if one organization is paying for the software and another is going to actually use it and they disagree about the basic business need. Despite all of these difficulties, agreement needs to be reached about what the software will do. The Acquisition Team must understand the individual stakeholder motives. How might the stakeholders benefit from the acquisition and how might the acquisition be a threat to them? A recommended method for accomplishing this is to meet with each stakeholder individually to encourage open communication and sharing of both positive and negative perspectives of the acquisition as it relates to them.

Stakeholders typically have commonly acknowledged business needs that the Acquisition Team can easily identify. However, stakeholders may also have unacknowledged motives (hidden agendas) that might drive the direction of their influence on the acquisition effort. The Acquisition Team needs to uncover and understand both the acknowledged business needs and these less obvious motives to effectively plan the acquisition. Table 1 includes an example of business needs/motives for the acquisition of a payroll software package:

Table 1 – Example of Business Needs/Motives for a Payroll Software Package Acquisition

Key Stakeholder	Business Needs/Motives
Accounting	<ul style="list-style-type: none"> <li>• Convenient mechanisms for capturing time worked, vacations, etc. for each employee</li> <li>• Track, reconcile, manage and report payroll</li> <li>• Eliminate the need to “chase” employees and supervisors for completed/ approved timesheets each pay period</li> <li>• Keep jobs (don’t automate people out of employment)</li> </ul>
HR	<ul style="list-style-type: none"> <li>• Automate and track employee elected deductions, contributions, savings, etc.</li> <li>• Eliminate the tedious tasks of dealing with paper forms from their workload</li> </ul>
Employees	<ul style="list-style-type: none"> <li>• Convenient mechanisms for reporting time worked, vacations, etc.</li> <li>• Detailed statement of earnings, deductions, contributions, savings, etc.</li> <li>• On time delivery of accurate paychecks and automatic deposit of payroll check to banks</li> <li>• Elimination of many different, complicated (sometimes hard to obtain) forms for requesting/changing deductions, contributions, insurance benefits, savings plans, etc.</li> </ul>
IRS, Social Security and State Tax Offices	<ul style="list-style-type: none"> <li>• Ability to automate wage garnishment, reporting and fund transfers for taxes owed</li> <li>• Ability to collect taxes owed on or before due date</li> <li>• Eliminate labor intensive paperwork</li> </ul>
Insurance Companies	<ul style="list-style-type: none"> <li>• Ability to automate premium payments through payroll deduction</li> <li>• Ability to sell more policies and options to employees</li> </ul>
Charities	<ul style="list-style-type: none"> <li>• Ability to establish and automate long term contributions through payroll deduction</li> </ul>
Employee’s Bank	<ul style="list-style-type: none"> <li>• Automated transfer of payroll into employee bank accounts</li> <li>• Get entire payroll check into employee bank account sooner</li> <li>• Eliminate labor intensive paperwork and need for human interaction with paycheck</li> </ul>
Company’s Bank	<ul style="list-style-type: none"> <li>• Automated transfer of payroll into employee bank accounts</li> <li>• Automated transfer of withheld taxes from company account</li> <li>• Keep company’s money in their accounts as long as possible</li> <li>• Eliminate labor intensive paperwork and need for human interaction with paycheck</li> </ul>

**Document the Acquisition Plan:** An Acquisition Plan is established to detail the methods to be employed throughout the acquisition project’s life cycle. The time spent defining the acquisition strategy early on will go a long way in assuring stability throughout the acquisition process and the life of the software. The acquisition planning process should link acquisition objectives and tasks to resources (time, people, funds, and technology). It must organize these resources and define a process for achieving the approval of all stakeholders to guarantee the adoption of the acquisition plan. It should then guide the acquisition activities and provide for the integration of the effort. Software acquisition activities are sensitive to the

same risks that occur in any project and require the same level of project and risk planning and management concerning decisions, budget, schedule, quality, etc.

## STEP 2 - DEFINE THE SOFTWARE PRODUCT

In the Define the Software Product step of the acquisition process, the software requirements are elicited, analyzed, specified and validated. This step drives the direction of the acquisition. The desired product must be adequately analyzed and its individual features and quality attributes decided on and documented. The Acquisition Team should prioritize the requirements and separate needs from wants. Successful acquisition projects are dependent on clearly defined requirements. We cannot stress enough the importance of good requirements documentation. As identified in a Standish Group survey, IT executive managers believe that the three major reasons that a project will succeed are user involvement, executive management support, and a *clear statement of requirements*. Opinions about why projects are impaired and ultimately canceled ranked *incomplete requirements* and lack of user involvement at the top of the list. [Standish-95]

In order to write good requirements documentation we need to understand the basic categories of requirements. Requirements are classically referred to in two categories: “Functional” and “Non-Functional” requirements and design constraints. The functional category specifically addresses tangible features, capabilities or functions of the desired software. This includes requirements for the various modes of operation, operational scenarios, data input/generation, data transformations and data outputs/storage. The non-functional category addresses requirements that must be inherent throughout the design and are not tied directly to any single feature or function. Non-functional requirements include requirements for usability, reliability, performance, supportability, safety, security and other product attributes. Design constraints include any predefined limitations on how the system can be designed and implemented. Constraints should include:

- Any required standards or policies that must be used
- Any external interfaces, communication protocols or operational limitations
- Any existing budget or schedule limitations
- Any platform limitations (e.g., hardware, operating system, language, tools)

The software requirements should be verified to ensure that they are:

- **Unambiguous:** Every requirements statement should have one and only one interpretation.
- **Complete:** Includes all the functions and functional attributes of the software and all constraints that must be satisfied.
- **Testable:** There exists a reasonably cost-effective way to determine that the software satisfies the requirements.
- **Traceable:** Each requirement should be traceable back to its source (e.g., system level requirements, standard, enhancement request). It should also be specified in a manner that allows traceability forward into the design, implementation and tests.
- **Consistent:** Internal conflicts do not exist between requirements.
- **Modifiable:** Each requirement should be specified in a coherent, easy-to-understand manner. The requirements should be non-redundant (i.e., each requirement is stated in only one place). Each requirement should be specified in a manner that allows it to be changed without excessive impact on other requirements.
- **Reasonable:** The requirements can be implemented using available technologies, techniques, tools, resources and personnel within the specified cost and schedule constraints.

## STEP 3 - DETERMINE THE ACQUISITION APPROACH

Once we understand the product requirements, we must determine the mechanism for acquiring the software. Our options include:

**COTS Software:** Things to consider if we purchase COTS or supplier developed software include:

- How fast do we need it delivered?
- How many users will there be (licenses needed)? What is the expected growth rate?
- Is there commercially available software and what is its cost?
- What is our projected budget for purchase and for ownership?
- What licensing restrictions can we live with?
- What are our integration needs with our existing systems? With our business processes?
- What is our risk tolerance?
- What are the rollout costs?

- What is the state of the technology (sunrise or sunset)?
- What are our competitors doing?

**Internally Developed Software:** We may have the need for software that is not in mainstream usage. This may necessitate the development of a customized software product to meet our unique needs. One way of accomplishing this is to create a “home grown” software product through developing it internally. (Note that if we decide to build the entire product in house, it becomes a software development project, which is outside the scope of the rest of this paper.) Things to consider if we develop the software internally include:

- How fast do we need to complete it?
- What are our core competencies?
- Do we have the needed infrastructure (tools, equipment, methodologies)?
- What are our development costs? What are the rollout costs?
- What is our risk tolerance?
- What is the state of the technology (sunrise or sunset)?
- What are our competitors doing?
- Can we market it?
- Can we maintain it?

**Software Developed Through Contract/Subcontract:** If a customized package is needed but the organization lacks the internal capabilities or resources to produce homegrown software, another option is to contract or subcontract the development to another company.

**Combination:** Finally, the organization may select to combine two or more of the above options. The acquirer may develop most of the software in-house but subcontract one or more functions to suppliers with specialized skills. For example, they might subcontract the GUI interface or security functions to a supplier with specific expertise in those areas. Another option would be to integrate COTS software that performs a specific function into a customized software product. For example a COTS database manager could be integrated into an inventory tracking software that is being internally developed.

For each viable option, a cost/benefit analysis should be performed that explores:

- Costs:
  - o acquisition and supplier management
  - o development
  - o purchase/licensing
  - o new equipment
  - o acceptance and roll-out
  - o operating and support costs
  - o maintenance costs including enhancements, corrections, and adaptation
- Benefits:
  - o increased efficiency/ effectiveness
  - o increased profits/revenues
  - o marketplace advantages
  - o taking advantage of innovation
  - o response to competition
  - o adaptation to change (e.g., environment, technology, business rules, regulations)
  - o increased customer/ employee satisfaction

#### **STEP 4 - IDENTIFY AND EVALUATE POTENTIAL SUPPLIERS**

In the Identify and Evaluate Potential Suppliers step a market search is performed to make sure that we are considering the available candidate suppliers and their software products. Sufficient preliminary research should be performed to narrow the list to the few potential suppliers that best match our business needs in order to target our evaluation and keep evaluation costs to a minimum. The data collected during the market search can be used as feedback to reassess our original requirements and to determine whether modification to those requirements will result in greater overall value in terms of cost, performance, availability, reliability, etc. The market analysis should also cover maintenance and support data, test results, and user satisfaction analyses.

In this step we evaluate selected potential suppliers by performing an in-depth examination of their capabilities, quality systems and products. Care should be taken to obtain enough information to compare the qualifications of each supplier in order to make an informed decision. Methods for doing this include:

- Formal Request-For-Proposal (RFP)
- Supplier Demos and Conferences
- Prototypes and Evaluation Copies
- Supplier Evaluations, References and Past Performance

**Formal Request-For-Proposal (RFP):** Advertising of an upcoming Request for Proposal (RFP) may not only identify unknown suppliers but it may also encourage suppliers to offer technological input and business advice. They may be able to suggest new technologies and capabilities not previously known about or considered by the acquirer as possible alternatives. The Request-For-Proposal (RFP) process is typically used only for customized software to be developed by the supplier. It is a very formal process where specific proposal requirements and questions are outlined by the acquirer in the RFP and responded to by the supplier in a proposal.

Since supplier's funds for bids and proposals are often limited it is important that the acquirer realize that they are competing for the attention of qualified suppliers. The time before formal RFP release is important for both the acquirer and the potential suppliers to examine potential alternatives and solutions.

IEEE/EIA 12207.2 contains an example outline of the contents for an RFP, however the SAM includes a much more detailed outline for the RFP contents including: [GASM V3.0]

- Specific tasks (quantifiable, measurable, and testable)
- Specifications and standards to be used for the program (relying on commercial standards and practices, whenever possible)
- Planned use of government-furnished equipment (GFE), government-furnished information (GFI), and/or government-furnished software (GFS)
- Requirements for:
  - the supplier to provide a comprehensive layout of program schedules (internal reviews, formal peer inspections, testing, technical meetings, etc.)
  - relevant and pertinent domain experience
  - a thorough Software Development Plan (SDP) and plans for its implementation and updates, to include a proposed test process plan
  - a description of the use of the chosen programming language, commercial-off-the-shelf (COTS), and reuse
  - appropriate software documentation
  - an open systems architecture and architecture performance verification
  - resources growth/margin verification
  - a total life cycle/total systems perspective
  - prototyping and/or demonstrations (ideally, a demonstration of an executable architecture as part of the proposal)
  - a progress, process, and quality measurement program, including a specific Metrics Usage Plan
  - a software quality assurance (SQA) program
  - supportability planning
  - a Risk Management Plan and its implementation
  - a Process Improvement Plan and its implementation
  - a process control mechanism (e.g., software development tools, management tools, etc.)
  - developing interface software with other system software and/or hardware
  - assessing software development maturity/capability
  - planned communications with any Independent Verification and Validation (IV&V) suppliers or agencies
  - delivery of the life cycle support environment

**Supplier Demos and Conferences:** Holding a supplier demonstration provides an excellent opportunity to see the product first hand and ask questions. Demos provide a mechanism for the Acquisition Team to evaluate the product and hear other team member's questions and comments in context with the demonstrations. Actually seeing the features of different software in a live demonstration provides a context for comparison. Perhaps the basic functionality is comparable but one tool has a more intuitive user interface. We may like the user-interface of one tool but are convinced of the structural integrity of another. Supplier conferences are back-to-back meetings, which are held with all of the suppliers demonstrating their products in succession. Conferences can foster competition and they can make it easier to make comparison.

The level of product knowledge and confidence of the supplier representative and their willingness to answer tough questions can be an indicator of our future relationship with them. Another indicator of future support is whether the supplier presents a “canned” demo or has spent the time to customize their demo to specifically address our business needs and requirements.

**Prototypes and Evaluation Copies:** A prototype can be used to determine the supplier’s understanding of the requirement. However, do not mistake a prototype for more than what it is — a shortcut for demonstrating proof-of-concept. Prototypes are typically not developed with supportability, readability, and usability in mind, and bypass normal configuration management, interface controls, technical documentation, and supportability requirements.

For many COTS products, evaluation copies are available as mechanisms for demonstrating the software functions/capabilities and for eliciting user buy-in. Check sheets, like the following example for a software configuration management tool, can be created for use by the acquisition team when evaluating the software product.

**CHECK SHEET - SOFTWARE CONFIGURATION MANAGEMENT TOOL(S) EVALUATION**

EVALUATOR \_\_\_\_\_ GROUP \_\_\_\_\_

EVALUATION CRITERIA	Supplier 1	Supplier 2	Supplier 3
1. Does the tool require purchase of any additional hardware or software?			
2. On what platforms is the tool available?			
3. How is interface across platforms? (i.e., how easy is the interface to setup, what is the response time, how is the user interface across platforms, etc.)			
4. Is the user interface consistent across available (and applicable) platforms?			
5. What is the best system for the central repository?			
6. How easy is the necessary administration? (i.e., is it convenient or time consuming)			
7. How easy is it to create the project/repository?			
8. How much space does the repository take on the disk?			
9. Does the tool allow different permissions to be allocated for different users?			
10. How secure are the different project domains?			
11. How easy is it to check-in/check-out the archived/configured files?			
12. Do you like the way the tool handles multiple checkouts?			
13. How are files checked out and where are they kept?			
14. How are versions of the different files stored in the archive/repository.			
15. What is the merge file capability of the tool? (Advantages and disadvantages)			
16. Does the tool limit which editors you can use?			
17. Does the tool provide a comprehensive audit trail for builds across platforms?			
18. Is the check in/out feature tied to the change tracking tool and visa versa?			
19. What audit trails are there for the software? (i.e., version history, what makes up a build, tracking tool information, etc.)			
20. How (using which tools) are the audit trails created?			
21. Does the build tool have limitations based on system?			
22. How are executables released/promoted? How much time does it take? (i.e., how easy is the process and how much time does it take, do you have to release/promote each file in the build/executable or just the executable, etc.)			
23. How much time does it take to label a revision/version of a file?			
24. Which Web Browsers does the tool use? How good are they?			
25. How customizable are the change tracking forms?			
26. How customizable are the change tracking reports?			
27. Can we keep separate databases for the same project?			
28. Can the compatible requirements traceability tool interface with other tools we are using? How?			
29. Can the tool track files, defect write-ups, etc. back to the requirements, not just back to the code?			
30. What metrics does the tool keep?			
31. Can external tools (e.g., evaluation/analysis type tools, etc.) be easily integrated with an event trigger?			
32. How does the tool hold up under REAL concurrent use (load/volume/stress)? (e.g., selecting a deficiency number or checking out/in files)			
33. How suited overall is the tool to the tasks we intend to use it for?			

34. How stable was the tool during the evaluation period?			
35. What kind of technical support is provided?			
36. How was the technical support during the evaluation period?			
37. Does the vendor plan to continue support for the tool indefinitely (or for how long)?			
38. How difficult will upgrades be for the areas of the tool that have been customized?			
39. What, if any, changes are planned for the tool in the next two years?			

**Past Performance, Supplier Evaluations and References:** A supplier who has had a consistent history of successfully providing software is more likely to perform effectively in the future. Past performance is a strong indicator of whether the supplier has the capability and ability to successfully complete delivery within schedule, on budget, and with the required level of functionality and quality. Like past performance, previous experience is a credible indicator of the likelihood that a supplier can successfully perform in the future. For instance, if a Software Engineering Institute (SEI) Capability Maturity Model (CMM<sup>®</sup>) Level 1 supplier struggled to complete a previous software project for us, that supplier still might be preferable to other Level 1 suppliers for a comparable project merely because the supplier probably learned many lessons that will benefit it on a subsequent project. Ideally, the supplier has both previous experience and an excellent record of past performance.

Evaluation criteria lists can be utilized when evaluating multiple suppliers to ensure complete coverage of all the important factors and promote consistency in the evaluation process. Evaluation criteria lists are made up of questions that help organize and structure the evaluation. These lists can be comprehensive and include all of the criteria being considered or they can be directed to a specific part of the evaluation (e.g., product specific criteria for a set of features or process specific like a software quality evaluation criteria). Standardized evaluation criteria lists can be created for use across multiple acquisition projects to promote efficiency and the sharing of lessons learned. However, these standardized lists should be tailored to the specific needs of each individual project.

Completed evaluation criteria list provide documented input into the supplier selection process. They also provide historical information and evidence that a comprehensive evaluation was performed in case there are any questions about the evaluation in the future.

**Example – Software Quality Process: Supplier Evaluation Criteria List**

Supplier: \_\_\_\_\_ Date Evaluated: \_\_\_\_\_ Evaluator: \_\_\_\_\_

<b>Evaluation Criteria</b>	<b>Strengths</b>	<b>Weaknesses</b>	<b>Improvement Activities</b>
What type of quality systems does supplier have?			
Has supplier been evaluated against SEI CMM <sup>®</sup> or CMMI <sup>SM</sup> ? When? What results?			
How does supplier independently evaluate/assess software engineering and related activities?			
Are the supplier's SQA processes documented? Controlled? Maintained? How is compliance verified?			
How are software deliverables and other work products verified and validated?			
What software work products do supplier's SQA organization audit/review? What criteria are used?			
What are the supplier's procedures for handling non-compliances and corrective action?			
To whom does the supplier's SQA organization report?			
How does the supplier's SQA organization support sub-contractor management activities?			

## STEP 5 – DEFINE CONTRACT REQUIREMENTS

In the Define Contract Requirements step of the acquisition process the contract or agreement type is selected and the contents of the contract or agreement is defined. A well-written contract minimizes the probability of misunderstandings and is a major contributor to a congenial relationship between the acquirer and the supplier. Experience has shown that when the contract is unambiguous and clearly defines the duties and responsibilities of the parties, the animosity that arises from quibbling over performance obligations usually can be avoided. The contract should be customized to consider the supplier's strengths and weaknesses. For example, if the supplier has achieved a high level of maturity, we may decide that online access to the supplier's development environment and management status reports (e.g., cost, schedule, risk management and metrics data) is an effective alternative to the traditional oversight mechanisms of formal reviews and submission/approval of data items. Alternatively, if a supplier's process for coordinating the efforts of different engineering disciplines and stake holders is relatively weak, we may add a requirement for an on-site liaison to support coordination with users and the suppliers developing interfacing systems. The following items should be considered for inclusion in the contract:

- Cost
- Schedule
- Performance
- Obligations
- Statement of Work
- Requirements
- Acceptance criteria

Acquirers should select contract types that are most likely to minimize risks for both parties and motivate suppliers to perform optimally. Type of contracts or agreements include:

**Fixed-Price Contracts:** Fixed-price type contracts are appropriate for software that can be objectively defined in the requirements specification and for which the risks of performance are manageable. For such acquisitions, performance-based statements of work, measurable performance standards and surveillance plans are ideally suited. The supplier is motivated to find improved methods of performance and cost control since this contract type places all costs and resulting profit or loss directly on them.

Advantages	Disadvantages
<ul style="list-style-type: none"> <li>• Provides firm assurance of ultimate cost</li> <li>• Insures prompt notification to acquirer of delays and extra costs resulting from changes</li> <li>• Provides maximum incentive for quickest completion at lowest cost</li> <li>• Involves minimal auditing by acquirer</li> </ul>	<ul style="list-style-type: none"> <li>• Requires exact knowledge of what is wanted before contract award</li> <li>• Supplier views all requirement changes as out-of-scope</li> <li>• Requires substantial time and cost to develop inquiry specs, solicit, and evaluate bids.</li> <li>• High bidding costs and risks may turn away qualified bidders</li> <li>• Cost may be increased by inflated bids to cover high-risk work</li> </ul>

**Cost-reimbursement contracts:** Cost-reimbursement type contracts are used only when costs cannot be estimated with sufficient accuracy to use any of the fixed-price type contracts. They establish an estimate of total cost for the purpose of obligating funds and establishing a ceiling that the supplier may not exceed without the approval of the acquirer. They are appropriate for software that can only be defined in general terms or for which the risks of performance are uncertain. Where possible, they should include specific incentive provisions in addition to the award fee to insure that suppliers are rewarded for good performance.

Advantages	Disadvantages
<ul style="list-style-type: none"> <li>• Does not require exact knowledge of what is wanted before contract award</li> <li>• Requires acquirer approval for costs beyond established ceiling</li> </ul>	<ul style="list-style-type: none"> <li>• No assurance of actual final cost</li> <li>• No financial incentive to minimize time and cost</li> <li>• Permits excessive design changes by acquirer increasing time and costs</li> <li>• Requires on-going audits of financial records</li> </ul>

**Incentive Contracts:** Incentive type contracts relate the amount of profit or the fee to the supplier's performance. Incentive increases or decreases are applied to performance targets rather than minimum performance requirements.

Advantages	Disadvantages
<ul style="list-style-type: none"> <li>• Motivates the supplier to effectively manage costs</li> <li>• Discourages inefficiency</li> </ul>	<ul style="list-style-type: none"> <li>• In attempts to increase efficiency shortcuts may be taken that compromise quality</li> </ul>

**Indefinite-Delivery Contracts:** Indefinite-delivery type contracts do not specify a firm quantity of supplies or services (other than a minimum or maximum quantity). Instead, they provide for the issuance of orders for the delivery of supplies or the performance of tasks during the period of the contract.

Advantages	Disadvantages
<ul style="list-style-type: none"> <li>• Allows the contract to be issued without knowledge of exact times/quantities of future deliveries</li> <li>• Allows the contract to be issued without commitment to execute it in its entirety</li> </ul>	<ul style="list-style-type: none"> <li>• By the time an order on the contract is issued, the supplier may need to be re-qualified</li> <li>• Production lead-time may delay delivery</li> </ul>

**Time-and-Materials, Labor-Hour and Letter Contracts:** Time-and-materials, labor-hour, and letter type contracts provide for acquiring supplies or services on the basis of direct labor hours at specified fixed hourly rates that include wages, overhead, general and administrative expenses, and profit. When the use of time-and-material, labor-hour, and letter contracts is appropriate, acquirers should employ performance-based methods to the maximum extent feasible.

Advantages	Disadvantages
<ul style="list-style-type: none"> <li>• Provides for acquiring products or services on the basis of direct labor hours and materials at cost</li> <li>• Allows the contract to be issued when it is not possible to accurately estimate the extent, duration or cost of the work</li> </ul>	<ul style="list-style-type: none"> <li>• No assurance of actual final cost</li> <li>• No financial incentive to minimize time and cost</li> <li>• Surveillance of supplier's performance is necessary to ensure efficiency and cost control</li> </ul>

**Agreements – Basic and Basic Ordering:** A basic agreement is not a contract. It is a written instrument of understanding, negotiated between the acquirer and supplier that contain contract clauses that might apply to future contracts between the parties during its term. It is normally used when a substantial number of separate contracts may be awarded to a supplier during a specified period and the acquirer has experienced previous problems with that supplier. Basic agreements are included into the appropriate contract by specific reference. A basic ordering agreement is also not a contract. It is a written instrument of understanding, negotiated between the acquirer and supplier that contains contract clauses that might apply to future contracts between the parties during its term. It is normally used to expedite contracting for uncertain requirements for supplies or services when specific items, quantities, and prices are not known at the time of the agreement. The goal is to reduce administrative lead-time, inventory investment, and inventory aging due to design changes. They are included in the appropriate contract by specific reference.

Advantages	Disadvantages
<ul style="list-style-type: none"> <li>• Allows for pre-documentation of understanding and contract clauses that can be referenced by the contract</li> <li>• Expedites the contracting activity</li> <li>• Reduces administrative lead-time</li> <li>• Reduces inventory investment and aging</li> </ul>	<ul style="list-style-type: none"> <li>• Is not a contract</li> <li>• A contract is still required to execute the agreement</li> </ul>

## STEP 6 - SELECT A SUPPLIER

In the Select a Supplier step of the acquisition process the results of the supplier evaluation are judged against established selection criteria, risk associated with each supplier are identified and analyzed, and a cost/benefit analysis is conducted. Based on this information, the final supplier selection is made.

**Cost/Benefit Analysis:** Cost/benefit analysis now needs to be revisited. This time however, we are looking at the costs and the benefits associated with each individual supplier. In this cost/benefit analysis we will also factor in the costs of risk management activities for each supplier. These costs include the cost of implementing the containment plans, tracking risks and the expected costs if the risks turn into problems. For example:

Costs	Benefits
<ul style="list-style-type: none"> <li>• Acquisition cost <ul style="list-style-type: none"> <li>- Acquisition and supplier management</li> <li>- <b>Risk management</b></li> <li>- Development</li> <li>- Purchase/licensing</li> <li>- New equipment</li> <li>- Acceptance</li> <li>- Roll-out</li> </ul> </li> <li>• Operating and support costs</li> <li>• Maintenance costs <ul style="list-style-type: none"> <li>- Enhancements</li> <li>- Corrections</li> <li>- Adaptation</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Increased efficiency/ effectiveness</li> <li>• Increased profits/revenues</li> <li>• Marketplace advantages</li> <li>• Take advantage of innovation</li> <li>• Response to competition</li> <li>• Adaptation to change (e.g., environment, technology, business rules, regulations)</li> <li>• Increased customer/employee satisfaction</li> </ul>

**Supplier Scoring:** A final scoring matrix can be created to summarize all of the evaluation criteria information and lower level scores for individual cost, schedule, product and process attributes.

Care should be taken that all information is gathered and the suppliers are scored at the same time in each of the respective criteria to eliminate variances in the scoring. This is best performed in a group-scoring meeting where participants have all of the gathered information available to assist in their scoring decisions and the scoring activity can maintain momentum and consistency.

Different techniques can be used for assigning a score to each attribute or criteria. Figure 2 illustrates the calculated scoring method where scores are assigned to each attribute based upon predefined formulas. Figure 3 illustrates the indexed scoring methods where specific predefined criteria are used to select each score. Figure 4 illustrates a weighted scoring system. A weight system is a method for quantifying qualitative data in order to minimize the effect of personal prejudice on supplier selection. The steps to implementing a weighting system include:

1. Assigning a numerical weight to each of the evaluation criteria
2. Grading the prospective suppliers on each criterion
3. Multiplying the weight by the grade to obtain a score for the individual criterion
4. Totaling the resultant products to compute an overall score

### Examples - Calculated scores:

Attribute	Max Score	Supplier 1	Supplier 2	Supplier 3
Ability to deliver by date needed	10	10	7	8
Purchase price / licensing costs	10	7	5	10
Licensing restrictions	5	5	4	5
Operating costs	15	12	15	5
Maintenance costs	10	5	10	7
Process capability	10	10	8	5
Product functionality matches needs	20	18	16	8
Product quality	20	20	15	15
Ease of integration with existing systems	5	3	5	3
Ease of integration with our business processes	10	10	7	10
Ability to customize product	5	5	4	5
Technical support	5	5	3	2
Training availability	10	10	5	5
<b>Total Score</b>	<b>135</b>	<b>120</b>	<b>104</b>	<b>88</b>

Ability to deliver by date needed = 10 points minus one point for each week past needed date

Product functionality meets needs = (# requirements met / Total requirements) x 20

Figure 2 – Calculated Scoring Method Example

### Example - Indexed scores:

Attribute	Max Score	Supplier 1	Supplier 2	Supplier 3
Ability to deliver by date needed	10	10	7	8
Purchase price / licensing costs	10	7	5	10
Licensing restrictions	5	5	4	5
Operating costs	15	12	15	5
Maintenance costs	10	5	10	7
Process capability	10	10	8	5
Product functionality matches needs	20	18	16	8
Product quality	20	15	15	10
Ease of integration with existing systems	5	3	5	3
Ease of integration with our business processes	10	10	7	10
Ability to customize product	5	5	4	5
Technical support	5	5	3	2
Training availability	10	10	5	5
<b>Total Score</b>	<b>135</b>	<b>120</b>	<b>104</b>	<b>88</b>

Ease of integration with existing systems =

5: Seamless integration expected

4: Minimal work required for integration (<= 1 week)

3: Average work required for integration (1-3 weeks)

2: Major work will be required for integration (>3 weeks)

1: Existing systems will have to be replaced to accommodate integration

0: Integration expected to fail

Figure 3 – Indexed Scoring Method Example

### Example - Weighted scoring of evaluation criteria:

Attribute	Max Score	Supplier 1	Supplier 2	Supplier 3
Ability to deliver by date needed	10	10	7	8
Purchase price / licensing costs	10	7	5	10
Licensing restrictions	5	5	4	5
Operating costs	15	12	15	5
Maintenance costs	10	5	10	7
Process capability	10	10	8	5
Product functionality matches needs	20	18	16	8
Product quality	20	20	15	15
Ease of integration with existing systems	5	3	5	3
Ease of integration with our business processes	10	10	7	10
Ability to customize product	5	5	4	5
Technical support	5	5	3	2
Training availability	10	10	5	5
<b>Total Score</b>	<b>135</b>	<b>120</b>	<b>104</b>	<b>88</b>

Process Capability	Grade	Weight	Score
Software quality	1	x1	1
Project management	1	x2	2
Configuration management	1	x1	1
Requirements management	0	x1	0
Systems & software engineering	1	x2	2
Verification & Validation	1	x2	2
Risk Management	0	x1	0
<b>Total Score:</b>			<b>8</b>

Grade: 1 = meets or exceeds requirements  
0 = does not meet requirements

Figure 4 – Weighted Scoring Method Example

**Supplier Qualification Audits:** Once the primary supplier candidate has been selected; a supplier qualification audit may be used as a final in-depth evaluation of that supplier’s quality system and capability to produce the required software prior to final selection. An audit is a planned, independent and documented assessment to determine whether agreed-upon requirements are being met. [Russell-97] A quality system audit evaluates an organization’s existing quality program’s adequacy and conformance to company policies, contractual commitments, industry standards and/or regulatory requirements.

The Acquisition Team selects an auditing organization (e.g., in-house audit team or external audit team) and defines the scope and objectives of the audit. The supplier provides the audit organization with the requested audit inputs (e.g., process and product documentation, plans, quality records). The lead auditor prepares the audit plan. Auditors prepare for the audit by studying the relevant documentation prior to the on-site visit. This allows the auditor to determine if the quality system is properly designed and/or documented. The auditors should evaluate the documentation to ensure that it is appropriate, adequate and current. The execution step of the audit is the actual on-site visit where data and information are gathered. The job of the auditors is to collect factual information, analyze and evaluate it in terms of the specified requirements, draw conclusions from this comparison and report results to management. After the completion of the on-site visit, the lead

auditor prepares an audit report detailing the findings of the audit. If a decision is made to actually select the supplier, any non-conformances or noncompliances found during the audit should be followed up by the auditee's management with corrective action plans. The audit team is then responsible for validating that the corrective actions were appropriately implemented and that they were effective in resolving the non-conformances or noncompliances. This step may impact the final negotiations and contract award step of the acquisition process.

## STEP 7 - NEGOTIATE AND AWARD CONTRACT

Once the supplier has been selected the contract or agreement terms are negotiated and the contract is awarded. Now is the time to do the final negotiation with the preferred supplier. In this step, the final terms of the contract are negotiated and the contract is awarded.

**Leverage:** The information we have acquired through researching supplier candidates can now be used to our advantage in negotiating the final terms. We should also consider areas where we have leverage including:

- Potential future sales including support agreements and additional software
- Information we have gained about the strengths of the supplier's competitors and the supplier's own weaknesses
- Use of our corporate identity as reference (future marketing)
- Ability to walk away from the table and select another supplier

We should also consider the leverage that the supplier has, including:

- Market place share, reputation, business stability and product recognition
- Superior process capability
- Domain knowledge and experience
- Superior product capabilities including functions or features that no other supplier can provide
- Ability to walk away from the table and not do business with us

**Risk Sharing:** The contract should equitably allocate the risks of the acquisition project between the supplier and the acquirer. This allocation should be based on which party is best able to manage the risk. Risks essentially fall within three categories: cost, schedule and performance. Each of these three categories of risk deserves a separate discussion in the contract.

## CONCLUSION

An acquisition project has a greater chance of a successful outcome if the acquirer:

- Properly plans the acquisition
- Gives appropriate attention to defining the software products business needs
- Decides on an acquisition approach only after consideration of each viable option
- Explores available sources to identify and evaluate potential suppliers
- Ensures that the software requirements are defined and fully understood
- Works to secure a capable supplier
- Formalizes the acquisition agreement in an appropriate contract vehicle

Once the supplier is selected and the agreement established, the acquisition project must be executed, tracked and controlled through supplier management activities and the complete software product must be accepted and the project must be closed. We explore these activities in the companion paper, *Software Acquisition and Supplier Management: Part 2 – Acquisition Project Execution & Supplier Management*.

## REFERENCES

- GASM V3.0 Department of the Air Force, Software Technology Support Center, *Guidelines for Successful Acquisition and Management of Software-Intensive Systems: Weapon Systems, Command and Control Systems, Management Information Systems (GSAM) Version 3.0*, May 2000.
- IEEE/EIA -12207 IEEE Standards Software Engineering, Volume 1, *Industry Implementation of International Standard ISO/IEC 12207: 1995 – Standard for Information Technology – Software Life Cycle Processes*, IEEE/EIA 12207.0-1996, The Institute of Electrical and Electronics Engineers, 1999, ISBN 0-7381-1559-2.
- Russell-97 ASQ Audit Division, J. P. Russell editing director, *The Quality Audit Handbook*, ASQ Quality Press, Milwaukee, WI, 1997, ISBN 0-87389-374-3.

*Santayana-05*

*George Santayana, Chapter 12, "Reason in Common Sense," *The Life of Reason, Volume 1*, 1905.*

*Standish-95*

*The Standish Group Report, <http://137.94.177.65/cficse/2000/resources/chaos.pdf>.*