How to Purchase COTS Software

Because perspectives of software vendors and users are different, there is no guarantee that commercial off-the-shelf software will meet the user's requirements

Software has become part of almost every business process, and the scope and complexity of software is increasing. Most software is purchased from a vendor and is referred to as commercial off-the-shelf software (COTS). How a company purchases COTS software is a major component for ensuring quality.

There are several key elements in purchasing COTS. First, a potential purchaser needs to understand that developers and users of software have different perspectives, and each must accept responsibility for different aspects of the relationship. Users must feel confident that their partner (vendor) is suited to be in a long-term relationship. And of course the product itself must be assessed.

A Difference of Perspective

In dealing with a vendor, it's important to know where your interests and the vendor's coincide and diverge. The principle may be obvious, but in practice, purchasers of COTS often seem confused about the roles of users and developers of software. The term "developer" identifies people who create software and for the purposes of this article is equivalent to the vendor. The term "user" identifies people who use software to perform their jobs.

From the start, developers and users have different objectives. Developers strive to create products that are desirable to their customers. Once a product is made it can be easily copied and distributed to many customers. Each customer has a different specific use or *application* for the software, and as the number of customers multiply, the number of applications increases.

The developer must develop the software with good quality attributes and perform validation. But validation of the software in context of its application is the job of the user. In addition, the user must ensure proper maintenance and control of the software on an ongoing basis.

The requirements of validation from the developer and user perspectives have many steps in common, but the focus of each step is different because their perspectives are different. Some of the differences in perspective are summed up in Table 1.

Software's Life Cycle

Both developer and user validation are part of the software development life cycle (SDLC). The life cycle consists of a series of stages that start with the birth of a concept and end with the retirement of the software. The developer looks to potential clients to determine what software is required. The developer interprets the requirements to develop a software product, then validates the software to ensure it is working as intended. At that point, the product is made available for sale.

Users determine a need for software that will satisfy their application and define user requirements. They evaluate the products available for sale and refine their user requirements. Users select a vendor and purchase the software. Next, they validate the software to ensure it is working for their application. Remember that the perspectives of the developer and user are different so there is no guarantee that the software will fulfill the requirements of the user's application. Once the validation process has shown that the software is capable of meeting the application requirements, users release the software for production use. Users must control upgrades of the software provided by the vendor and control changes to the configuration of the software made by users.

Eventually the application will be retired, either because the vendor has discontinued the software product or because the user has decided to replace the application. Retirement of the software and retirement of the application are separate and may or may not occur at the same time.

Once a company has purchased software, it is dependent on the developer. This relationship is much like a marriage. The keys to a successful developer–user relationship are selecting the right partner and fostering good communication.

Steps in Purchasing COTS

User requirements. The process of purchasing COTS starts with the user writing user requirements that explain in general terms what the software needs to do (but not how the software needs to do it). The users then

David Nettleton *is a 21 CFR Part 11 specialist and project management consultant at Computer System Validation, 916.928.1470, computer systemvalidation.com*

Developer	User
What functionality will the software have to be useful to the users?	What does the application need to do to implement the specific business processes?
How will the functions be developed into a software product?	How will the functions be used in business processes?
What testing demonstrates that the software is capable of performing the predefined functions?	What testing demonstrates that the application and associated business processes function as intended?
Not applicable	What testing demonstrates that the application is installed correctly?
Do the training materials explain how the software can be used?	Do the users know how to use the application to perform their jobs?
What are the potential hazards associated with the software as related to its interaction with the operating environment?	What are the potential hazards (risks) associated with the use of the application in the associated business processes?
How are changes to the software functions managed?	How are changes to the application, configuration, and business processes managed?
Is the product ready for sale (release)?	Is the system ready for production use (release)?

Table 1. Software developers (vendors) have one set of priorities, which may not coincide with those of the user (purchaser).

research the market to learn about available products, drawing on trade shows, periodicals, colleagues in other companies, and the Internet. As a result of this research users can update their user requirements to include features that they desire because they are available.

RFI/RFP. Users send the updated user requirements to a "long" list of target vendors in the form of a Request for Information (RFI). The vendors reply to the RFI by explaining how their products and services meet the user requirements. Users compile the RFI responses to form a product and vendor comparison matrix. The matrix allows the users to compare competing products and uniformly assess the vendors.

The most promising vendors are put on a "short" list. Together, users and Purchasing send out a Request for Proposal (RFP) to all vendors on the short list. The RFP is a formal request for information on products and services. It includes information about associated costs, software licensing parameters, and additional components. Costs can vary greatly depending on the licensing model chosen. Are costs dependent on total number of users, number of concurrent users, number of client installations, or number of server installations as required for production, test, training, and development environments? The RFP also identifies other software and hardware components required for the software to function. Often database software, operating systems, server hardware, backup software, and other utility software are needed but aren't provided by the target software vendor.

From the RFP replies, the users gain an understanding of the actual costs to be incurred. Typically, the users will narrow down the desired vendors to one or two.

References. Next, users request references to determine how well the vendor's products and services meet existing customer expectations. The ideal reference is a relatively new client in the same business or of the same size as your company. The goal is to learn the vendor's strengths and weaknesses. A good technique is to ask, "What would you do differently?" If possible, talk to people involved in different aspects of the software implementation at the same company.

Audit. When a vendor is tentatively chosen, users and Quality representatives (typically consultants) audit the developer and create a vendor evaluation report. A vendor audit should address the following.

- How well does the vendor develop software (SDLC)?
- What are the quality standards for the software?
- What is the history of the product?
- What is the history of the vendor?
- What is the financial health of the vendor?

If the audit results are favorable the procurement process continues with the creation of a contract. If the audit results are not favorable, the users should consider additional vendors. The user and vendor relationship is essential for success. If the relationship doesn't appear positive, move on to another vendor.

Developers that are not open to an audit show that they aren't interested in a long-term relationship. Developers that are willing to change their quality systems to meet customer quality expectations demonstrate that they are interested in building relationships that will support future changes in software use.

Contract. Users and Purchasing create a contract. A purchasing requisition or purchase order may accompany the contract. The contract should address:

- *Products.* Provided by the vendor and other third parties.
- *Installation services* include responsibilities of the vendor and the users' company. Typically, instead of having the vendor perform all installation work, it is best to have the vendor teach the users' IT staff the installation procedure. This education is invaluable for maintenance activities such as upgrades.
- *Implementation services* include training, configuration assistance, and project management. Again, it is usually best to have the vendor train in-house trainers. The vendors don't know the specific application of the software and can only provide generic training.
- *Maintenance fees*, including technical support and upgrades.
- *Payment schedule*. Payments should be divided with the first payment up front, the next after successful installation, and final payment after user validation. Remember that all vendors are primarily in business to make a profit. Vendors tend to be much more accommodating before full payment is made.

After the contract is approved by the vendor and users' company, the user validation activities begin. A product user group is an important communications vehicle. It promotes communication between the users, and between the users and the developer. This communication is essential through the retirement of the software application. **BPI**