Off-the-Shelf Software: A Broader Picture
By Bryan Chojnowski, Reglera Director of Quality

In the past decade, there has been a sea change in the business software domain. Many companies are no longer expending significant internal resources in order to develop software solutions from scratch and are instead opting for software that meets most or all of the business requirements as delivered ‘off the shelf’ by a third party. Off-the-shelf (OTS) software is an extremely broad category that encompasses software that can be purchased and utilized with minimal or no configuration. It comprises spreadsheet and word processing applications; statistical packages; resources planning applications (e.g. ERPs and MRPs); Customer Relationship Management (CRM) solutions; Quality System databases for CAPA, Complaint Handling, Auditing, and Document Control; Laboratory Information Systems; accounting software; software embedded in medical devices and equipment; etc. There are virtually unlimited types of off-the-shelf software.

There are many benefits to using off-the-shelf software. Foremost, it is the vendor who expends the resources to design, develop, test, and support the software on an ongoing basis. Often, the software vendor also has extensive expertise in the target market for the software and thus is able to incorporate functionality to support best practice methodologies into the software. In some cases, for certain types of OTS software, the vendor will also provide a hosted solution, which eliminates the need for your company to purchase and maintain servers and supporting hardware. Altogether, this can allow a company to implement a tailored software solution more quickly and cost effectively; however, there are potential downsides related to OTS software as well.

One of the fundamental limitations of implementing OTS software is that your company typically will not have direct control over the OTS software feature set, including what functionality is added, changed, or removed with each release. This could result in a situation where your company is not able to dictate the schedule for incorporation or persistence of business- or compliance-critical functionality in the system. Next, the release schedule of the software is also typically dictated by the software vendor. This could lead to a forced upgrade of the software, either because the prior version is no longer supported or because the latest version will be pushed to all clients simultaneously, such as in a multi-tenant, hosted environment. Lastly, your company will be reliant on the vendor for the software system technical details that can be critical to successful application integration and interfacing as well as the troubleshooting and resolution of software issues that arise.

Furthermore, while the use of OTS software can virtually eliminate internal software development activities for a business, it presents a unique set of challenges and doesn’t obviate the need to ensure compliance through software validation and procedural controls. One of the more common compliance challenges for businesses in the medical industry is in the validation of off-the-shelf software. This article will focus on the validation of business and Quality System OTS software and discuss other aspects of implementing such OTS systems that can have a significant impact on your business.
**OTS Validation Decision**
If someone asks Reglera if off-the-shelf software needs to be validated, our answer is ‘It depends.’ It depends on how the software is being utilized, aka, the intended use of the software. Per the FDA General Principles of Software Validation, software must be validated if it is used to achieve compliance with predicate rule (e.g., 21 CFR 820, 21 CFR 1271). According to the FDA General Principles of Software Validation Guidance, ‘All production and/or quality system software, even if purchased off-the-shelf, should have documented requirements that fully define its intended use, and information against which testing results and other evidence can be compared, to show that the software is validated for its intended use.’

The FDA further states in the guidance that ‘Off-the-shelf software may have many capabilities, only a few of which are needed by the device manufacturer...When device manufacturers purchase "off-the-shelf" software, they must ensure that it will perform as intended in their chosen application.’ This means that regardless of how the software vendor defines the intended utilization of the system, each business should independently define its specific use of the system and ensure this use is validated. Therefore, the first step in deciding whether to validate an OTS software system is to understand its intended use. If a software system or functions within the system are not utilized for compliance with predicate rule, validation of the whole or parts of the system may not need to be performed. In other words, the breadth of the validation will be driven by the functionality used to comply with predicate rule. An example is an accounting package, which is used only to capture business financial transactions. While the system is important to the business, the tracking of finances is not required by FDA regulations; consequently, the FDA does not require it to be validated. Even so, some companies may choose to validate all software systems because it makes good business sense to ensure systems operate per specifications and/or to set a consistent company expectation regarding software validation activities.

**OTS Validation Approach**
Once the high-level decision has been made regarding validation, there are two specific elements that need to be addressed: One, the system’s actual compliance with applicable predicate rule, standards, and general regulations, and two, the system’s formal validation.

While the software vendor may strongly assert that its application is compliant with FDA regulations and other standards, it is prudent to verify compliance, ideally before making the decision to purchase, but absolutely before moving the system to production use. Again, the system should be compliant with applicable elements of the predicate rule and should also adhere to requirements as set forth in 21CFR Part 11, which applies to all records in electronic form that are created, modified, maintained, archived, retrieved, or transmitted under any records requirements set forth in FDA regulations. In addition, a company may decide to demonstrate adherence to other industry standards such as ISO 13485. A compliance assessment should be performed
for each applicable regulation and standard. Relevant elements of the applicable regulations and standards will also be incorporated into the system requirements so that they are validated. Upon moving the system to production use, it would render an entire validation effort meaningless if the system is not also compliant with applicable regulations and standards. Stated another way, validation establishes that the intended use of the system is satisfied, and the intended use inherently must include compliance considerations.

As an example, a Complaint System used by a medical device company must be compliant with 21 CFR Part 820.198, which covers Complaint files. The use of the system to approve and store the official Complaint records would also necessitate compliance with 21 CFR Part 11, which governs electronic records and electronic signatures. The specific subsections of Part 820.198 that are related to functions performed within the software are captured as requirements against which validation is performed. In order to limit downstream risk, a compliance assessment is performed at the time of software selection. If gaps are found during the compliance assessment, the findings could be used to direct additional vendor development or configuration activities, or a different, more compliant software solution could be sought.

The basic methodology for conducting validation of off-the-shelf software involves the following activities, which are performed in concert with implementation activities such as system configuration.

- Define business’ use of the system, ideally including use cases and explicit clarification of in-scope and out-of-scope functionality
- Determine validation deliverables set based on system type, system risk, project scope, and degree of system modification
- Review existing vendor system and validation documentation
- Devise strategy for validation that leverages vendor documentation/systems as applicable
- Create applicable system requirements specification and design documentation
- Generate requirements-traceable validation protocol and execute validation
- Put in place system use, administration, and maintenance procedures to ensure the system is used as intended and remains in a validated state

Now that the top-level aspects of OTS software selection and validation have been covered, it is essential to cover vendor-related topics that often go overlooked until after the contract is signed and the initial OTS implementation is well underway.

If the vendor has documented system requirements, you will need to assess the requirements to ensure your company’s needs are met. If there are requirements beyond those documented by the vendor, you will need to create a separate requirements document to capture your company’s specific requirements. The validation testing will then need to cover and trace to these requirements. If this scenario applies, you will need to determine whether any of the vendor requirements and testing can be leveraged toward your validation. Remember that even if all the
requirements are covered in the vendor and company documentation, it must be easy to trace and explain or you will have difficulties demonstrating compliance.

**Vendor As Critical Business Partner**

In addition to vetting the software to ensure it meets your company’s requirements, it is almost equally important to assess the software vendor’s supporting processes to determine its capabilities to ensure long-term satisfaction with the system. Unless you plan to purchase a single version of the software and never update it, your company will be relying on the vendor’s operations on a continual basis. Ninety percent or more of the total cost of ownership of a software system can be associated with production use and updates of the system after its initial implementation, so you shouldn’t discount the importance of a vendor’s overall support capabilities and software update practices. For this reason, it is crucial to thoroughly evaluate the vendor, both qualitatively and quantitatively, and preferably when it is still prospective. Use the selection process to thoroughly qualify the vendor. In addition to following standard supplier qualification processes, which take into account the risk and criticality of the software for your company, the below areas should also be considered.

**General Responsiveness**: Submit several requests for data during the selection process and meet with as many people as possible at the vendor in order to ascertain the level of engagement, commitment, and expertise. Does the vendor do what it says it will do? Be wary of vendors who don’t admit to any deficiencies in their systems or operations. Be on the lookout for instances during the sales process where a prospective vendor incorrectly communicates the capabilities of the software or the support functions or doesn’t follow through on commitments to provide more information. Communication problems during the sales and qualification process will likely not go away after a contract is signed and often serve as a warning for future challenges.

**OTS ‘Degree’**: A business’ relationship with the vendor will be somewhat dependent upon the degree to which the software is ‘off-the-shelf’. For instance, word processing software purchased at the local computer store will have a different level of vendor support than will an off-the-shelf Quality System software solution. In other words, the target consumer of the system will dictate the level of personalized customer service that will be provided and recognizing the ‘degree of OTS’ will allow the implementation, validation, and maintenance paths to be charted appropriately.

**Client Mix**: Consider the vendor’s mix of software clients. How many clients do they have? Is the vendor’s typical client large or small? What markets do they serve and what is the percentage of clients in each market? How unique is your company’s use of the system? At the end of the day, businesses exist to make money; in order to make money, they need to make their clients happy so they will logically shape their software to satisfy as much of their client base as possible. The mix of clients and markets served will very much dictate the future path of the software and support processes. Therefore, the best case is that your company is similar to the majority of
the vendor’s clients in terms of functional and regulatory requirements that are satisfied by the system. If this is not the case, your company’s relative contribution to the vendor’s revenue stream and the vendor’s commitment to serve your market segment should be understood. If your company is small and has unique needs in a market segment to which the vendor is not committed, it would be prudent to assess competitive offerings from other vendors who may better match your company’s size and specific needs.

**Hosted Application Sharing:** If the software system is web-based and hosted, it is important to understand how many clients access the same hosted deployment of the application. The number of clients on a hosted deployment will influence the flexibility the vendor has regarding software release dates and times. If it is important to your company to be able to control the production release of each new software version, you should pursue arrangements to have a dedicated instance of the hosted application so that no other companies will be able to influence the release schedule.

**Software Update Frequency:** Try to determine how often the vendor releases new versions of the software as well as if you will be forced to upgrade the software every time a new release is available. The typical release schedule for the software will dictate ongoing validation/re-validation activities, which can significantly contribute to the total cost of ownership and should be taken into account for internal resource planning purposes. The best case is therefore that your company can control if and when system updates occur. It is good to understand the visibility to Bug Fixes and Hot Patches. Ideally, the client will be made aware of Bug Fixes and Hot Patches so that impact on functionality used can be assessed. This is important as what the vendor deems an insignificant bug fix may be more than that to your company.

**Software Update Previews:** It is also important to understand whether or not the vendor provides the opportunity to preview each new release before it is moved to production. This is a key concern if the software is a hosted, web-based solution as the client often has little control over the timing of the production release of the software. Pre-production availability of the software will allow your company to become familiar with the new software functionality, evaluate its impact on the established intended use, and perform any necessary validation activities before production release.

**Software Release Notes:** The key mechanism for assessing if and how a functional change to the software affects your company’s use of the software is a set of detailed software release notes. If there are multiple modules in the software and your company doesn’t use and/or validate all of the modules, it is particularly important that the release notes break out changes by module. In addition, the vendor should provide user and administrative documentation (i.e., Manuals or Help Files) that walk through all aspects of the use of the system as well as administrative functions to be performed by your company.
Key Software Controls: In addition to the basic quality system elements that allow the software vendor to ensure the ongoing quality of its product and service, pay particular attention to the following processes:

- Change Control
- Configuration Management
- Environment Deployment

Change Control ensures updates to the software application are assessed and integrated in a systematic fashion. The vendor should be able to demonstrate how system change requests are logged, reviewed, linked with a release, and decisions communicated back to the requestor. Change Control participation should ensure representation from business functions that can speak to technical, regulatory, and business impacts of each proposed change.

Configuration Management controls the elements of the software system and its documentation so that they are identifiable and synchronized, and so that only approved versions are moved to production. If the vendor is performing technical configuration or customization to the system on behalf of your company, it is essential that it be able to tightly control and deploy updates to your system.

Besides reviewing procedures, there are other practical ways to assess how effective Change Control and Configuration Management are being performed. For instance, you can review the system change requests submitted by internal and external resources. The requests should be thoroughly documented and there should be evidence of review, disposition, and feedback for each request.

Environment Deployments are often the most difficult aspect of implementing and updating a database application. It is beneficial to use discrete environments for development, test, and production use. Typically, configuration changes will be made in development and promoted to the test and production environments. The promotion process needs to take into account, either automatically or manually, that there may be existing master data and records that need to be unaffected when updating the production environment. Using the test environment solely for acceptance and testing will ensure that test or sample data are not included in an environment promotion.

Technical Troubleshooting and Root Cause Analysis: Inevitably, there will be issues with the software system that occur over time. As noted previously, your company will be reliant on the OTS software vendor for the software system technical details that can be critical to successful troubleshooting, root cause analysis, and resolution of software issues that arise. During the assessment, review past reported issues and how they were resolved. The current software technology requires the coordinated performance of hardware, operating systems, vendor software, web
browsers, printer drivers, etc., so it is essential that the vendor not only demonstrate technical expertise across these interconnected domains but also a commitment to truly resolving technical issues that affect its customer base.

**Part 11 Knowledge:** You should also consider if the vendor is aware of and well-versed in 21 CFR Part 11. This is important as a vendor who is not familiar with Part 11, which covers electronic records and signatures, may make changes to the software over time that unknowingly affect its Part 11 compliance.

**Software Configuration vs. Customization**
Off-the-shelf software is typically configured by the client company. Configuration often involves activities such as adding system users, populating field dropdown lists, adding entities, etc. but can also involve renaming data fields to allow the process to better match an existing client process. Software customizations, which are sometimes required to ensure compliance or match a client’s defined business process, involve branching the code from the base set. Customizations therefore frequently require additional development coding each time a new release of the software is installed, which can result in the expenditure of development resource time with each release and increase the time required to perform validation. Limiting client software changes to configuration only allows the client to more easily upgrade the software with each new release. Therefore, it is beneficial to utilize software that meets your company’s needs without having to be customized.

**Final Thoughts**
Off-the-Shelf software solutions are a great way to tap into established industry best practices while avoiding the need to design and develop bespoke software from scratch. The up-front costs of implementation can also be reduced by selecting a hosted, web-based solution. However, it is important to recognize that using an outside software vendor includes giving up some level of control of the solution. In order to mitigate this risk, it is prudent to perform up-front vendor assessment activities, including both qualitative and quantitative aspects. In concert with a standard supplier quality assessment, your company should also understand key parameters such as the degree to which the software is ‘OTS’, the vendor’s client mix and demographics, the vendor’s approach to hosted solutions as applicable, the frequency of and communications related to software updates by the vendor, the vendor’s key software controls, and its knowledge of 21 CFR Part 11. Your company will still have the responsibility to ensure compliance through software validation and procedural controls, so all of the above factors should be taken into account when calculating the true cost of ownership of a software solution.

**References**
General Principles of Software Validation; Final Guidance for Industry and FDA Staff, January 11, 2002

*Bryan Chojnowski is a Director of Quality for Reglera Corporation, which provides regulatory consulting, outsourcing, and resources to the medical device and human tissue industries.*