JOURNAL OF
MultiState
TAXATION AND INCENTIVES

FEBRUARY 2012

Best Practices in Transaction Tax Systems Implementation

Diane L. Yetter

With more than 7,000 different authorities imposing their own taxes in the U.S. alone,

it is critical for businesses that sell items in multiple jurisdictions to implement technology solutions to accurately manage their sales and use tax functions. A wide variety of software systems are available today that can embed the tax calculation within the systems that process a company's sales and purchases. Choosing the right system for a company is half the battle. Implementing the system is the other half.

Implementing an indirect tax automation system—whether a new system, an upgrade, or a platform transfer—is no small task, but the results are well worth the investment of time, money, and effort when an effective system is finally in place. After all, the right system will help companies reduce errors, increase productivity, and avoid the heavy fines and penalties of a negative audit, not to mention avoiding the labor costs associated with correcting the mistakes that were found in the audit. Given the quantity and variety of users involved in making sales and use tax decisions within a company, automation is critical.

Nevertheless, despite the sophisticated technology available for managing sales tax systems, many companies are still using spreadsheets and manual entry to manage their sales and use taxes. Or, they have an automated tax system but it is not providing the desired results. Things are still falling through the cracks, eventually coming to the company's attention in a government audit after it is too late to collect taxes from customers.

Especially in today's economy, when companies are forced to do more with less, the right tax system, coupled with the right implementation process, can bring about greater efficiency and a substantial return on investment. Also, given the current economic conditions, states are getting more aggressive in collecting sales taxes; therefore, companies must have their sales tax systems in perfect working order to avoid hefty fines. In other words, this is the perfect time for companies to evaluate their systems and decide whether improvements need to be made.

This article will explore the best practices needed to implement an effective sales tax system. These best practices apply to companies that are in one of the following situations:

1. Implementing an automated system for the first time.
2. Changing their systems—either the financial system or the tax engine.
3. Upgrading their existing tax system to a different version or platform.
4. Merging their data to another system.
5. Expanding into a new market and/or new geographic area.

Fortunately, there are experts who can guide companies through this transition. Since these projects are not part of the day-to-day responsibilities of a transaction tax or information technology (IT) person, attempting to accomplish a project of this magnitude and importance without someone with significant experience can add risk to the project. A seasoned consultant should be able to supplement in this situation, either as a coach and advisor or as a critical member of the team. Finding the right partner, one who knows the business, the industry, the systems, and is willing to provide the needed assistance can help an implementation project succeed.

The following discussion may serve as a guide to companies during the transition of a new tax system, from planning to execution to maintenance.

Types of Tax Automation Systems

Before selecting a software system, one must understand the types of systems that are available on the market today and their capabilities. The decision as to which option is most appropriate will depend on a company's tax and business requirements, the capabilities of the system, and, in large part, the capabilities of the users. The more sophisticated the system, the less reliance on users for making tax decisions. The less functionality in the system, the more decisions must be made by the users.

Host systems. Most financial systems include some level of sales tax processing capability. Whether the capability is sufficient to meet a business's requirements depends on the complexity of the taxation of its products and the jurisdictions where it does business.

Determining whether to rely on the standard delivered functionality entails an evaluation of the user's proficiency in making tax determinations, the effort required to maintain the tax rules, and the flexibility in the system. Product taxability indicators are usually limited to a single indicator and do not allow for state specific taxability indicators. For products that have different taxability by jurisdiction, this functionality will be insufficient to meet tax calculation requirements.

Customer taxability indicators are often tied to an address. As long as unique address records are created for each delivery address and all items sold to the given customer qualify for an exemption, these indicators may be sufficient.

DIANE L. YETTER, CPA, MST, is a strategist, advisor, speaker, and author in the field of sales and use tax. She is president and founder of YETTER, a sales tax consulting and tax technology firm in Chicago, Illinois. She is also the founder of The Sales Tax Institute, a premier think tank that offers live and online courses to educate business professionals about sales and use tax. She serves on the Executive Committee of the Board of Advisors of the University of Kansas School of Business, where she is an adjunct professor.

Procurement systems, including purchasing and accounts payable functions, also may include some level of tax processing. This is generally limited, however, and in some cases only a taxable or exempt indicator is permitted.

Host systems typically rely heavily on the users' making tax decisions as they select the correct tax codes that likely indicate taxability as well as rates. Companies with complex taxability rules or significant business in many states are likely not good candidates for this type of solution.

Bolt-on products. For more sophisticated tax processing, several third-party "bolt-on" products are available that will interface with financial systems. In order to communicate with the financial system, an interface must be established between the bolt-on package and the financial system. Some financial systems have standard interfaces available to select tax packages. The interface, however, may not utilize all functionality available in the tax package.

To obtain full functionality as required for tax management, enhancements to either the financial system or interface may be required. User training may influence the effectiveness of the tax package.

These solutions are best suited for companies with more-complex tax profiles and those that want to control the taxability decisions among a limited number of individuals—likely within the tax department.

Custom designed systems. In companies that have created a custom billing system, custom tax determination may be incorporated into the billing calculation. This situation often exists for companies in industries (e.g., food, medicine, telecommunications and utilities) with unusual tax requirements or nonstandard sales tax rates. These customized systems, although more difficult to maintain, often have been enhanced over time and can be very sophisticated and accurate. In some instances, a third-party rate package is incorporated into the custom calculation in lieu of manual rate updating. As custom systems are converted to standard billing systems, the custom tax functionality must be replaced with canned programs.

Application service provider (ASP)/ cloud-hosted systems. As an alternative to installing and maintaining software on the company's hardware, "application service providers" are available to host and maintain some tax and financial applications. ASP-hosted applications can be cost-efficient, as the ASP will manage all the infrastructure including system and, usually, content updates. If an ASP hosts the tax application, the hosting agreement should included arrangements for access to the tax data.

These solutions are best suited for smaller to mid-sized companies that have fairly standard tax profiles. The systems usually do not facilitate complex rules that the tax engine may offer in an on-premises version. These products/services are ideal for companies that have grown to such a size that they have tax-collection responsibilities in multiple states but lack the full tax staff necessary to support an on-premises solution.

Non-interactive solutions. As an alternative to integrated sales and use tax systems, there are non-interactive calculating solutions. These solutions are simplistic in nature and require manual updating. Standard office applications such as Excel and Access can be used to build templates that can incorporate company-specific tax rules for calculating tax on outbound invoices or on procurement transactions. Customized web applications also can be created that provide for more flexibility and capabilities at reasonable costs. Maintenance of rules and rates is required to comply with tax changes as they occur.

These systematic solutions may be best suited for companies that have limited billing systems that cannot interface with a standard third-party tax package or that do not facilitate passing enough detailed information to the tax package, or users that are not connected to an interactive system at the time they are calculating tax. In some situations, a combination of a non-interactive solution with an interactive solution may be necessary.

Many companies use hand-held units to generate invoices upon delivery or service at a customer's location. In some cases, the employee will not have connectivity, thus requiring a non-interactive solution. In these cases, an off-line version of an interactive or bolt-on solution may be appropriate. This will depend on the approach being used for the hand-held units from an overall accounting perspective.

Implementation Plan

Before implementing a new tax system, companies must first formulate a plan to address how the change will affect the company as a whole, who will be involved in the change, who will be affected by the change, and how long the changeover will take. The impact that a transaction tax system has on the company as a whole is often underestimated. Since sales and use taxes affect virtually all financial and business operations, a decision should not be entered into without a thorough plan. To derive answers to these questions, company officials must ask themselves a series of other questions.

Which processes require tax calculation? There are many aspects of a business that are involved with sales and use taxes, including:

1. Sales order processing.
2. Sales invoice billing.
3. Accounts receivable.
4. Credit processes.
5. Purchase order processing.
6. Accounts payable.
7. Purchasing cards.
8. Capital asset acquisitions.
9. Goods movement.
10. E-commerce.
11. E-procurement.

Who will be involved? A tax system implementation affects many departments of a company. Too often, companies assign the task to one department without seeking the input of the other departments involved. The team and points of contact should be well-established at the project's onset. Both the IT and tax departments, as well as third-party implementation consultants, must work together as a team. Under a best practices approach, the tax department, rather than IT, should lead a sales and use tax systems implementation project. The most critical decisions of the project should be made by the group that has the business technical knowledge to do so—the tax department. When the project sponsor is from the IT department, decisions tend to be based on infrastructure or timing rather than business requirements.

The following departments are integral to any sales and use tax process and should be involved in all aspects of the implementation process:

1. Tax.
2. Accounting, including fixed assets, general ledger, and financial reporting.
3. Accounts payable.
4. Accounts receivable.
5. Customer service.
6. Order entry.
7. Master data maintenance, including customer, vendor, item, and plant (the company's own locations).
8. Information systems.

Effective communication, cooperation, and knowledge-sharing among the entire team are crucial to the success of the implementation. Any decisions that have been made and processes that have been agreed upon should be documented and distributed to the implementation team members.

A kick-off meeting at the beginning of a project will help all team members understand the overall project and how their work will affect it. Members of the third-party consulting team should be educated on the client's business and processes and participate in the initial meetings. Similarly, it is critical that IT, business, and tax participants, as well as key members of the consulting and software team, are present at the kick-off meetings to properly understanding the business workflow.

The team should understand the time commitment required to ensure the project's success and timeliness, especially during the testing phase. Team members will need to be flexible and willing to rearrange priorities and schedules in order to keep the project on track for the go-live date. Additionally, "enterprise resource planning" (ERP) resources should be readily available to the team for troubleshooting and integration issues if the need arises. For any significant effort, a best practices approach would be to temporarily reassign key team members to the project team and relieve them of their normal duties.

Create an implementation plan that describes each phase and the required steps so that the entire team understands the timeline and responsibilities of the project. This should include a thorough testing plan and a detailed go-live plan, with defined checkpoints and responsibilities. The implementation plan should be used as a guide to ensure that all phases are followed through to completion.

Project phases. Each phase in an implementation project is critical to the project's success. Skipping or minimizing any of the steps will impact the success of not only the project as a whole, but each of the subsequent phases. Shortcuts in the requirements phase will result in challenges during the functional design phase, which will then impact the development, research and configuration, and testing phases. Listed below are the most common phases of an implementation project

with the relative portion of time that should be expended on each phase. It is difficult to estimate how long each phase and project typically takes since each project is unique. Nevertheless, focusing on the relative percentage of time that each phase should take will help a project team understand the relationship of each phase to the entire project.

Each phase relies on team members' focusing on certain responsibilities within their area of expertise. The team members who usually bear the primary responsibilities within each phase are included below along with the size of each phase (as to time or cost) relative to the total project.

1. **Requirements definition:** Usually led by the tax consultant or tax representative. All team members should participate in their respective component discussions. Typically around 15% of project budget.
2. **Tax software selection:** Usually led by IT, tax, and procurement. May be coordinated by tax consultant or ERP consultant. Typically around 7% of project budget.
3. **Functional design:** Usually led by tax consultant or tax representative but involves all team members. Typically around 5% of project budget.
4. **Interface development:** Led by either ERP, tax engine technical consultant, or IT. Typically around 5% of project budget unless significant customization or development is required.
5. **Tax research and configuration:** Led by tax representative and tax consultant. Will vary depending on complexity and ability to use tax content of tax engine. Configuration is typically around 10% of project time.
6. **Testing:** Usually led by ERP or tax consultant, with appropriate team members assisting. Typically around 22% of project time.
7. **Reporting:** Usually led by tax representative or general ledger/accounting team member. Typically around 4% of project time.
8. **Training:** Usually performed by either the tax engine company or as knowledge transfer sessions with all consultants. Typically around 4% of the project budget.
9. **Production cutover:** Usually led by the ERP conversion team member. Most of this effort is related to master data conversion. Typically, the time to convert to production is about 2% of the project time.
10. **Project management:** Usually led by overall project manager or tax consultant. Typically around 12% of project time.
11. **Post-implementation maintenance:** If provided by a tax consultant, the consultant would work with either the help desk or tax representative. This will vary depending on the level of support required. Using an estimate of two weeks of support after production cutover, this phase is typically around 14% of project budget.

Following best practices, for each of these phases there are key questions and issues that should be addressed in order to ensure a successful project. The responses here can result in adjustments to the plan. As each phase is executed, flexibility is key, given that sales and use taxes are complex, far reaching, and can be impacted by the business operations.

Requirements Definition

The first phase of any transaction tax systems project should be the evaluation and gathering of all the business and tax requirements. Ideally, this phase should be conducted before a solution is selected, but this rarely occurs. Since the optimal solution depends on the requirements, selecting a product before the requirements are known can lead to the need to either compromise or incur more costs than necessary.

Of all the phases in the project, this is by far the most important and the one where the efforts of the entire team are needed. When this phase is limited by either time, resources, or participation of key stakeholders, requirements that are undiscovered or not fully developed create challenges for the remainder of the project. Ideally, this phase should be led by someone knowledgeable about the business, the potential tax systems being considered, the financial system to which the tax engine will be connected, and the tax issues facing the business and the industry. Using someone who is familiar with all of these components is a best practices approach, as this person will be able to identify the issues related to the requirements, and understand and propose solutions on how they can be handled.

The following questions will help direct the project and shape the business, tax, and systems requirements.

Evaluation of system functionality.

1. Is the project part of an overall ERP replacement or upgrade, or is it a standalone tax system project?
2. Is the project an upgrade of an existing tax engine, a replacement of an existing tax engine, or a new solution?
3. Is the desired platform/architecture "enterprise/on-premise" or "software as a solution" (SaaS)?
4. Which processes currently have automated sales and use tax functionality?
5. Are there limitations in the current systems that need to be enhanced?
6. Will all current functionality remain in the new system?
7. What are the efforts to dismantle the current taxability process?
8. What systems will need to integrate with the selected tax system?
9. Is some of the tax logic currently handled in the interface instead of the tax system?

Evaluation of tax requirements.

1. Which jurisdictions are included in the project?
2. If jurisdictions outside the U.S. are included, is local country tax expertise available to assist with defining the tax requirements?
3. How is taxation handled currently?
4. What audit issues have been identified?
5. Will there be changes in taxation that may affect customers?
6. Is taxation based on customer exceptions?
7. Is taxation based on product exceptions?

8. Can tax groups be established for products or customers?
9. Are there situations where information other than customer, product, and jurisdiction is needed to determine a tax calculation?
10. Are there special rates or jurisdiction issues that need to be handled based on product or customer?
11. Are there maximum tax rules based on quantity?
12. How should tax be displayed on the invoice, order, and the screen display? Should it be one line with a generic description, one line with the state name in the description, multiple lines for each level of tax with generic descriptions, or multiple lines for each level with the exact jurisdiction name?

Tax Software Selection

All businesses are unique and host an array of different requirements. Once the type of solution is identified, research the various providers within that arena. Different software packages will offer different sets of possibilities. The team should be aware of all the benefits/disadvantages of the different software packages before making the final selection. Working with a consultant familiar with many different options can help the team narrow down the viable contenders. Not conducting a full evaluation of at least a reasonable portion or even all of the potential solutions prior to selection puts fulfillment of all requirements at risk.

In making the software selection, consider the following steps:

1. Determine the selection process (informal evaluation, on-site vendor demos, or rigorous "request for proposal" (RFP)). The larger the project and the more complex the tax requirements, the more rigorous the evaluation should be.
2. Create evaluation criteria and scoring approaches.
3. Develop RFPs and select vendors for evaluation.
4. Evaluate technical and functional capabilities of tax software.
5. Schedule tax software vendor presentations.
6. Inform vendors of requirements for demo scenarios and other critical information for their presentation. Under a best practice approach, the vendors should establish that the software can meet your specific requirements, rather than providing merely a general sales demo or showing their standard functionality.
7. Determine whether the tax software offers standard interfaces to the selected financial systems. Is there an additional charge for the interface? Can enhancements be easily incorporated into the interface?
8. Evaluate implementation specialists' experience with tax software and financial software.
9. Evaluate in-house implementation experience and availability of tax and information systems staff.

Functional Design

Once the software is selected and the requirements are defined, the functional design phase can be completed. In this phase, the information gathered in those prior phases is translated into the functional requirements within the tax engine, interface, and financial system. This phase also will help identify any business process changes or additions that may be required to ensure success of the project within the production environment.

The following questions will be used to refine the requirements and translate them appropriately into more systematic requirements.

1. Will additional data elements be needed to be passed from the financial system to the tax engine in order to handle tax requirements?
2. Will the financial system's screen/table displays need to be altered to accommodate the additional data elements, or are the required data elements already a part of the financial system structure?
3. Will required fields have to be altered for different types of information?
4. Will data values within existing data fields need to be adjusted or new values added to the existing options?
5. How will freight and discounts be handled within the financial system?
6. Will the new system handle tax variances on vendor-charged tax in the accounts payable system?
7. Will additional warning/error messages be necessary?
8. Will customized reports be required, or do the standard reports available within either the financial system or the tax engine meet the needs of the end user?

Interface Design

Depending on the tax engine selected and the financial system with which it is interfacing, this phase can be either significant or limited. If a tax engine is selected that offers a standard interface to the financial system, the efforts will be limited to evaluating any enhancements or custom data attributes that are needed to meet the tax requirements. If a standard interface is not available, development of the interface will be required. Additionally, if the financial system does not have standard integration points for third-party tax engines, the integration points must be created within the financial system. In these instances, a key member of the project team should be a developer familiar with the financial system and able to understand not only how to code the requirements but also the business processes within the financial system.

Evaluation of standard interface to partner software.

1. If a third-party tax software product is being considered, determine if a standard interface has been written to the financial software package.
2. Evaluate release information for tax software and interface to financial software.
3. Contact other users to determine if the interface works as described.
4. Contact other users to determine how the tax engine provider and financial system company coordinate support and work to resolve issues.

5. Investigate user groups supported by either the tax or financial software provider specific to the tax interface.
6. Evaluate all the functionality of the tax software and compare it to the functionality provided in the standard interface.
7. Determine if the standard interface will meet your needs or will require modifications.
8. If modifications are necessary, evaluate impact on tax software and financial system upgrades.

Design of interface if standard interface to partner software is unavailable. If a third-party software product is being considered and there is not a standard interface available to the financial system, a custom interface will be required

1. Evaluate all the functionality of the tax software and determine your requirements.
2. Do not shortchange your design—build the required and desired functionality into the design, as it may not be implemented later.
3. Determine if internal or external programmers will provide the interface.
4. Define the scope of the project and make sure the tax department is involved.
5. Coordinate with all the impacted business user functional areas to see what changes they would like built into the system and review proposed changes with them.

Development of custom taxation functionality. If a third-party tax package will not be used, evaluate the standard sales tax functionality in the financial software.

1. Many major financial systems may be able to handle value added taxes (VAT) but not U.S. sales taxes.
2. If custom tax functionality will be used, evaluate how tax rates will be obtained, interfaced, and maintained in the custom system.
3. Evaluate all tax requirements to verify that they will be programmed correctly.
4. Allow for variability for taxability determination

Tax Research and Configuration

A critical phase in the project is defining the tax rules. If the tax rules are not accurate, tax calculations will not meet the requirements regardless of all other efforts. Based on the business and tax requirements established in the "requirements definition" phase and the information gathered in the "functional design" phase, which will determine how the data elements will be defined and what data will be available for a given transaction, the tax rules need to be researched and configured in the tax system.

In many cases, the tax software may contain taxability content as well as tax rates. When this applies, the content categories must be analyzed to select the appropriate ones for each data value combination being sent from the financial system. In a best practice scenario, this content should be reviewed and evaluated to determine if it meets the company tax profile.

Depending on the tax software engine selected, additional configuration may be necessary to define the company's rules regarding nexus, customer exemptions, product rules, special rates, and authorities, as well as any other company-specific rules. Understanding the various methods available within the tax system and how all the functionality works together will help determine the best practice for configuration. Keeping in mind the efforts not only to initially configure the system but also to maintain the rules and troubleshoot errors, will help determine the best approach.

The following issues should be evaluated during the configuration planning process.

Customer taxability issues.

1. Are customers fully exempt on all items purchased?
2. Are customers taxed differently based on product purchased?
3. Are customers taxed based on purchase order indication of usage?
4. Will customer groups be used to define exempt status?
5. How will exemption certificates be obtained, reviewed, and entered into the tax system?
6. What is the timing between when a new customer is acquired and set up in the financial system and when the first order is taken?
7. Who will maintain exemption certificate data?
8. In which system will exemption certificate data be maintained—financial system, tax engine, or exemption certificate system, and how will the data be integrated and kept in sync?
9. Will certificate images be stored?
10. How and when will the taxability determinations be researched and entered into the tax system?

Product taxability issues.

1. Are there items that are taxed differently regardless of the customer, such as freight and labor, and statutory exemptions that do not require exemption certificates?
2. Are there products that are exempt based on customer usage, but all customers that purchase the product are eligible for the exemption?

A wide variety of software systems are available today. Choosing the right system is half the battle; implementing the system is the other half.

3. Are there special tax rates or maximum taxes based on the product, such as machinery and equipment rates?
4. How will taxable invoice additions such as freight be handled when the item purchased is exempt?
5. How and when will the taxability determinations be researched and entered into the tax system?

Jurisdiction issues.

1. How will the system handle interstate vs. intrastate transactions and the determination of the correct sales/use tax type?
2. How will the system determine the correct local jurisdiction for intrastate transactions (origin or destination)?
3. Are there home-rule local tax jurisdiction issues?
4. Are there special jurisdiction rates that apply because, e.g., a location is in an enterprise zone?

Tax type issues.

1. Do you need the ability to determine whether the transaction is a sale or a rental or involves a service?
2. Are you subject to additional industry-specific taxes that may require separate calculation rules and rates, such as telecommunications, meals, lodging, medical, or motor fuel?
3. Are there different tax rates or reporting requirements for the different tax types (sales, use, or rental)?

Credits and debits.

1. Will the system calculate taxes on credit and debit memos?
2. What date will be used as the transaction date for determining the correct tax rate for credits and debits?
3. Do you need the ability to process tax-only credits and debits?
4. How will the system properly account for tax credits and debits for accurate return reporting?
5. How will the taxability of pricing adjustments be handled if the original invoice item is not identified?
6. Are there pricing and customer accommodation adjustments? If so, how should these be taxed and how will they be entered into the customer account?

Procurement issues.

1. How will items be identified if a material group is not assigned in master data?
2. Will it be necessary to know accurate ship-from information?
3. How will transactions be handled when the location of where items will be used is unknown?
4. Is it necessary to track and document taxes paid to vendors?
5. Is it necessary to perform vendor tax validation on accounts payable invoices? If so, what will be done with the differences—reject the invoice, pay as charged, or short-pay the vendor?
6. Is there a need to determine use tax or validate tax on procurement card transactions?
7. Will the company be using any sort of e-procurement processing such as "electronic data interchange" (EDI)[1] or "evaluated receipts settlement" (ERS)[2] or web store? If so, how will the tax systems interact with these processes?

Foreign transactions.

1. Is there a need to calculate any foreign transaction taxes?
2. What impact does the different tax structure have on the tax calculation program?
3. What country-specific rules are there either for language, reporting, or invoicing?
4. Does the tax engine fully support the foreign jurisdiction for rates, rules, and content?

Testing

Within a tax system implementation, various different types of testing are required for a successful project. The tests should simulate real business transactions and processes, and the affected business stakeholders should provide input on the development of the test scenarios. Ideally, super-users from the business should execute the tests. A detailed test plan for each test phase should be developed that includes the type of tests needed, the data values to be used within the test scenario, the expected results, and the functionality that is being tested. The results of each test scenario should be documented. As development changes occur, test scenarios should be re-executed to ensure there is no impact on the results.

Tax system unit testing. Once the tax rules are configured within the tax engine, unit testing should occur using the tax engine internal testing functionality. This can be done by creating test scenarios that simulate all complex transactions that employ the different types of tax configuration settings.

Once the tax system unit testing is successful, it is easier to troubleshoot testing errors in the integration testing. If the test was successful during this phase but fails when it is executed from the financial system, the source of the error can be isolated to either the financial system or the integration.

Basic system functionality testing. Once the basic tax and financial system functionality is available, testing should be conducted to determine if all the types of business transactions will be handled appropriately. The initial test should validate that the connection is operating as expected between the tax engine and the financial system. Once the connection is validated, all interface logic should be validated to ensure that the correct data fields are passing from the financial system into the correct tax engine fields.

Be sure that the formatting of data values is as expected. All types of order entry methods, including all fields that can be used, should be tested to ensure that no functionality has been compromised. All modules (order entry, invoice billing, credit processing, purchasing, accounts payable) should be tested to make sure all functionality is appropriate. The company should confirm that the transactions are correctly posting to the financial systems and general ledger. This test should be conducted in each environment to ensure there are no changes in the system environments. If a financial system conversion is in process, it

[1] EDI is the computer-to-computer exchange of business transactions in a standardized, structured electronic format without human intervention, resulting in transactions where the only record is electronic.

[2] ERS involves a formal agreement between a buyer and a seller that places directly on the buyer the calculation burden for not only the sales/use tax but also the purchase itself. Rather than having the seller issue an invoice, the buyer relies on pre-established terms, presented in a purchase order or similar agreement, to calculate its obligations and remit payment directly to the seller, based on the acceptance of goods or services delivered. Up-to-date knowledge of the seller's tax status (i.e., where it has nexus) must be maintained by the buyer in order to assess the proper tax (sales or use) at the proper rate.

is important that all tax system testing is done in the same release and same environment as the live production system.

Integration testing. Once basic system functionality is approved, full integration testing, which will include tax-specific functionality, should be conducted. Test the various combinations of special tax rules to ensure that all results are as expected. Also, correct interpretation of parameters that trigger special tax rules, such as customer number or product identification, should be tested.

A best practice is to develop a list of integration test transactions and expected results that can be used repeatedly to test the system when patches or upgrades are applied. This practice will minimize efforts to keep the software current and also can be used during troubleshooting.

Reporting

A key output of any tax calculation system is the ability to provide reports for tax compliance, financial reporting, and management. Most third-party bolt-on tax engines include functionality to capture tax-calculation results within a database for use in reporting. Standard reports provided by the tax engines may be sufficient for some businesses, while others will require more sophisticated reporting options. Enhanced reporting components are included in the more sophisticated tax engines. It will be necessary, however, to generate tax reports from the financial systems for reconciliation purposes. If the tax engine reporting functionality is not sufficient for the business requirements, other business reporting tools should be evaluated. If the tax calculation tool selected does not capture tax results in a reporting or results database, then all reporting will be done from the financial system.

Reconciliation. As part of the testing of the reports, the results in the tax system and in the financial system general ledger should be reconciled. Issues that can create differences include transaction dates, adjustments, reversals, and voids. Understanding how these are processed between the financial system and tax engine, and confirming that they are handled in the same manner, will ensure that the systems will reconcile. If there are differences with processes that cannot be resolved, a manual process should be created so that the necessary adjustments can be made to the appropriate system.

Audit documentation. All companies should ensure that the data used to prepare their tax returns is sufficient and available for use during an audit. In evaluating the reporting approach that will impact the ease of audit defense, some key issues to consider include:

1. What information will be available from either the tax or financial system to support a taxing authority's sales and use tax audits?
2. Does either the tax or financial system facilitate computerized audit techniques?
3. Will the tax department be able to access and generate reports as required for audits when needed from historical records?
4. How will data be generated from prior tax systems or earlier versions?
5. If a SaaS solution is implemented, will access to the tax data be maintained if the contract is terminated?
6. How much data can be accessed live vs. archived access? Who has access to the archived data and how will it be accessed?

Return reporting functionality. Since one of the critical needs of the tax calculation data is the preparation and filing of the periodic tax returns, the functionality of the reports module related to summarization and export of data for compliance packages should be evaluated and tested. Consider the following issues not only in the report testing but also in the evaluation of the tax calculation and reporting solutions.

1. If a third-party tax software partner is selected, does it offer an integrated returns package that is consistent with your filing needs?
2. Is there an electronic import process available to enter the tax information from the tax calculation system into the returns package, or does an interface need to be developed?
3. Does the tax return package include all the jurisdictions for which you are required to file?
4. What is the acceptance rate of the jurisdictions for the computer-generated returns?
5. If you currently report using EDI or web filing, does the tax return package support these file-generation formats?
6. Does the tax return package have any limitations that should be considered in terms of reports or payment processes?

Training/Knowledge Transfer

Depending on the project team members and the implementation approach, the determination of what training is required for which team members, and the timing, will vary. If a consultant familiar with the tax engine will be directing the project and handling the development and configuration of the system, the in-house team may not require training at the project outset. It is advisable for the consultant to provide at least overview training of the functionality of the tax system during the requirements phase to level-set the understanding of the entire project team. If the project will be run by in-house tax and/or IT staff, however, these individuals should attend the tax systems training prior to the initiation of the project.

At the end of the project, appropriate knowledge-sharing sessions should be conducted to transition the information gathered, business process changes, and technical knowledge to the individuals who will have responsibility for maintaining the system in production. Different types of training will be beneficial for different team members as described below.

Tax system training for programmers. If a third-party tax package is used, evaluate the need for internal programmers or support staff to attend training. Even if internal resources will not be handling the implementation, it is nevertheless advisable for them to be familiar with the tax system for maintenance tasks. Training should occur in a timely fashion—generally near the beginning of the implementation process.

Tax system training for tax professionals. If a third-party tax package is used, tax department personnel should attend training. If possible, also train a back-up person. If tax personnel will not be performing the tax-system configuration, it is likely better to delay training until the end of the project in order to minimize knowledge-loss over time.

Financial system training for users. If there are any changes in the financial system to handle new tax requirements, be sure business users obtain the appropriate training in a timely fashion. If users will be maintaining any tax software settings, provide appropriate training.

Financial system training for tax professionals. In order to conduct testing, tax department personnel will need training, access, and support from other users in most aspects of the financial system. This will include order entry, invoicing, billing, credits, purchasing, and accounts payable.

It will also be beneficial—and a best practice—for the primary tax department personnel working on the tax-system implementation to be very familiar with the financial systems. Their understanding of the business transaction processes and data available will facilitate the identification of business and tax requirements, configuration needs, and testing requirements. Other processes to which the tax department will need access in the financial system include testing taxability transactions, issuing tax credits/debits, and generating reports for reconciliation with the tax system. Tax department personnel will need additional access if the tax calculation is not a third-party product but is built into the financial system.

Project knowledge transfer. If the project team includes individuals other than those who will be maintaining the system after it moves to production or who have responsibility for tax functions, a full knowledge transfer session should occur. The project team should provide comprehensive documentation of all configurations in the financial and tax systems, design requirements, interface logic, and test results. The individuals responsible for ongoing maintenance should have a complete understanding of all the configuration and update processes.

Project Management

Key to the success of any project is strong project management. A tax system implementation spans many departments, involves internal and external resources (even if only the tax engine and financial system providers) and has significant dependent activities. Without strong project management and oversight, including the involvement of an executive sponsor and steering committee, risks to the project increase.

Someone with project management experience should serve in this role. A best practice is to fill the role with an individual familiar with tax systems implementation projects as well as having knowledge of the specific systems (financial and tax) being implemented.

Despite the sophisticated technology available, many companies are still using spreadsheets and manual entry to manage their sales and use taxes.

Production Cutover

Once the system project is fully tested, migration to production can begin. There are some key considerations and steps necessary to ensure a successful move to production.

Transition date. The company must select an appropriate transition date for the new tax system. If the project is part of an overall financial system project, the selection of the transition date is often tied to minimizing the impact on all affected users. Calendar or fiscal month-end is desirable in order to minimize the reporting issues that come with merging multiple systems in the same reporting period. From a financial reporting perspective, however, this may provide challenges with month-end accounting activities. Determine if there are any "lock-out" time frames during the cutover process that would impact tax compliance. Coordinate the transition date to ensure that regular compliance activities are not compromised with system or personnel resources.

Address cleanup. The accuracy of any tax calculation depends on accurate and valid address data. As part of the cutover process, the company must decide whether a customer/vendor/facility address record cleanup is necessary. Issues to consider include whether multi-county or outside-city-limits issues apply, whether the use of "ZIP+4" or street-level address validation within the tax or financial system should occur and, if necessary, whether a separate address-validation package should be used.

Allow sufficient time for customer/vendor contact and the updating of system records. Establish a contingent address-handling plan for nonresponsive customers/vendors. A best practice approach is to include an exemption certificate update request at the same time.

Pre-implementation transactions conversion. As part of the cutover plan, open transactions need to be migrated to the new system. Develop a conversion plan for these items, such as orders entered but not yet billed. Establish conversion rules for all parameters that may have changed (e.g., customer number, product number, jurisdiction identification, and taxability indicators), and set up a method for adding data elements to these transactions that may not have existed in the prior system. Also, establish a method for handling credits in the new system for original transactions from the prior system.

It is also necessary to include in the plan whether closed transactions will be migrated to the new system. This is primarily an issue in a financial system change but could apply also to a tax engine change. If the archived transactions will not be migrated to the new system, it is important to ensure that access to the prior system is maintained or that data is exported in such a way that all elements and data integrity are maintained, and that appropriate backup and access are available to the tax department. Conversion tables that are necessary to translate data should be maintained. If new exemption certificates are obtained, prior certificates should be maintained until the periods covered by the certificates either expire or are audited.

Customer relation issues. If changes to taxation policy could impact customer invoices, customers should be so notified. Regardless of whether customers are notified, all customer-service and sales representatives should be made aware of any changes in taxation policy so they can be responsive to inquiries. The company should determine a conversion policy for customers that have been exempt but did not respond to exemption certificate update requests.

Post-Implementation Maintenance

Once the system has moved into production, efforts shift to maintenance. These activities will vary depending on the design, configuration, and business processes. At a minimum, tax content updates for rates and rules will be required. Other maintenance activities will vary.

Customer taxability maintenance. Maintenance of exempt customer data will be required on an ongoing basis after implementation. Determine which department will have responsibility for this function. Monitor and maintain any special taxation rules based on customers in the tax system.

Product taxability maintenance. As the business adds new products or product groups, it is necessary to maintain product tax mapping and rules in the tax system. A tax department representative should be part of the creation and approval process for new tax groups. In addition, someone from the tax department should be assigned to monitor the assignment of tax product groups to new products.

Jurisdiction maintenance. The tax department must monitor business activities to determine when new jurisdictions are entered that require registration. Once a new jurisdiction is entered, the tax system should be updated accordingly. This process will include not only the nexus tables but also a review of customer and product tax rules. If a custom tax calculation system is built, tax rates and jurisdiction rules must be monitored and maintained.

System maintenance. Software providers frequently release patches and updates for their products. It is important to track new releases and updates for tax software and for the financial system interface to the tax software in order to evaluate if any functionality changes impact the company's implementation. Set up policies to provide for monthly updating of rates, content, and other releases from all tax software partners, including who has update responsibility, which environments are updated and in what order, the extent of testing to be performed on the updates, and any necessary communication to users affected by the changes.

A best practice policy is to establish a key contact in the company's information systems department who will tend to system needs and who understands the tax system and its integration with the financial system. It is also critical to educate upper management about the need, early in the process, for tax personnel involvement in system changes.

Conclusion

Implementing a new transaction tax system is undoubtedly a large undertaking, but companies that take shortcuts or do not apply best practices during implementation will find that their new systems will yield less-than-ideal results. To the contrary, companies that invest the time and money necessary to properly implement a new system will find that their new system will help improve company efficiency, reduce the number of government audits, decrease assessments, and even help contribute to the company's bottom line. ■