Technology projects, like most other initiatives, carry a certain degree of risk. For large, enterprise-wide projects such as an enterprise resource planning system, that risk can be quite large, with millions of dollars committed to an effort that will last for two years or more. Smaller projects might not have quite the same exposure, but the risks are similar:

- The project could be underestimated and therefore might not be completed on time, on budget, or within the established scope.
- Organizational priorities could change in the middle of the project, rendering some objectives moot and elevating previously discarded concerns.
- Staff resources might be reallocated to other initiatives.
- Third-party vendors and contractors might not perform as advertised, and deliverables (previously identified portions of the project that are to be completed at a specified time) could be late or of substandard quality.
- Funding could be reduced or eliminated.
- Computer hardware or other equipment might not be sized properly to support the goals of the project.

These are just a few of the concerns that finance officers face when assessing risk and identifying ways to mitigate that risk.

**MILESTONE SCHEDULES**

As a best practice, the GFOA advocates deliverable-based milestone schedules as the best way to protect the organization against implementation risk. In this model, payments are made according to a predefined milestone schedule, with payment contingent on receipt and acceptance of deliverables that were defined before the contract was signed. The contract itself, or statement of work that accompanies the contract, can specify the deliverables. It should include the following:

- A narrative description of the deliverable and a sample format.
- The timing of when the government should expect the deliverable.
- The level of detail for the deliverable.
- Clearly defined roles and responsibilities for developing the deliverable.
- Defined dependencies, or other tasks or deliverables that have an impact on content and timing (that is, one deliverable depends on completion and acceptance of another).

Because of the volume of information that can be attached to each deliverable, some governments and vendors choose to create deliverable expectation documents outside the contract, to keep the contract itself more readable. See Exhibit 1 for an example of the resulting schedule.

Along with a deliverable schedule, there should also be an agreement on acceptance criteria and a review process. Vendors will not likely agree to such a payment schedule without knowing the timeframe.

**Exhibit 1: A Deliverable Expectation Schedule**

<table>
<thead>
<tr>
<th>DED Number</th>
<th>Deliverable</th>
<th>Phase(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DED01</td>
<td>Contract</td>
<td>I</td>
</tr>
<tr>
<td>DED02</td>
<td>Management Plans</td>
<td>I</td>
</tr>
<tr>
<td>DED03</td>
<td>Project Plan/Schedule</td>
<td>All</td>
</tr>
<tr>
<td>DED04</td>
<td>COA Analysis &amp; Spreadsheet</td>
<td>I</td>
</tr>
<tr>
<td>DED05</td>
<td>Best Practices Recommendations Spreadsheet</td>
<td>1,2,3,4</td>
</tr>
<tr>
<td>DED06</td>
<td>Final Best Practices Document</td>
<td>1,2,3,4</td>
</tr>
<tr>
<td>DED07</td>
<td>Core User Procedural Documents</td>
<td>1,2,3,4</td>
</tr>
<tr>
<td>DED08</td>
<td>Data Conversion Crosswalks</td>
<td>1,2,3,4</td>
</tr>
<tr>
<td>DED09</td>
<td>Data Conversion Files</td>
<td>1,2,3,4</td>
</tr>
<tr>
<td>DED10</td>
<td>Data Conversion Comparison Reports</td>
<td>1,2,3,4</td>
</tr>
<tr>
<td>DED11</td>
<td>Modification/Customization Specifications</td>
<td>TBD</td>
</tr>
<tr>
<td>DED12</td>
<td>Modifications/Customizations</td>
<td>TBD</td>
</tr>
<tr>
<td>DED13</td>
<td>Interface/Import/Export Specifications</td>
<td>TBD</td>
</tr>
<tr>
<td>DED14</td>
<td>Custom Interfaces/Imports/Exports</td>
<td>TBD</td>
</tr>
</tbody>
</table>
for the review and acceptance process, and in any event, governments should conduct those reviews quickly to keep the project moving forward. Typically, governments need five to ten days to review a deliverable, and each deliverable usually has an acceptance form the government can sign to indicate acceptance and willingness to make a corresponding payment.

Milestone payments provide sound contract protection because no funds are expended until the government is satisfied with the quality of the deliverable. At no point in this type of contract does the government make payments based on consultant hours — payments are made only when a deliverable of value has been received and is of acceptable quality. If the vendor fails to deliver, or provides a deliverable of substandard quality, the government is not obligated to pay. (Of course, the vendor will have an opportunity to rectify any problems.)

Further protection can be provided by using retainage, or holdbacks. A government that employs retainage holds back a specified amount, typically around 10 percent of the total dollar value of the deliverable, until it is satisfied that the project has been completed appropriately. Releasing the retainage can be controversial, as the government will want to wait until it sees all of the desired outcomes of the project, while the vendor will want to see those funds once the project is complete. For example, in an ERP implementation, the government will want to keep retainage until the first year-end processing cycle has completed, while the vendor may want the release to occur after the first month-end. Negotiating the retainage release date is therefore an important part of the contract.

WARRANTIES

When purchasing hardware, software, or technology services, governments generally have a list of requirements for the products and services being purchased. These are typically listed in a request for proposals or request for quotation document, with vendors responding with a yes for each requirement that their hardware or software can accomplish.

Of course, simply indicating that a requirement can be met does not guarantee that the software or hardware actually meets that requirement. The government can protect itself by ensuring that responses to requirements are included in the contract as a warranty. To do this, the vendor proposal is often included as an attachment to the contract.

The finance officer should be aware of two concerns with warranties. First, if the software vendor and implementation vendor are different, it might be difficult to get the implementation vendor to agree to warrant the software against requirements. This situation can be mitigated by ensuring that the RFP or RFQ states very clearly that responses will constitute a warranty, and that the successful vendor will be expected to attach those responses to a contract.

The second concern is internal. In the cases of ERP contracts or other software-based projects, the number of requirements can soar into the thousands, depending on project scope and the size of the government. In those cases, checking the deliverables against requirements can be a daunting task, and the government should plan accordingly. Some governments, recognizing the effort involved in matching deliverables to requirements, retain an “owner’s representative” or similar third party to assist with that work and help manage the contract.

SERVICE-LEVEL AGREEMENTS

Governments commonly outsource some portion of the IT function. Help desk services, network management services, and even program management offices can be successfully outsourced to a third party, allowing the government to obtain necessary expertise without adding to its payroll.

What makes outsourcing and other services-based contracts successful is the use of a service-level agreement. This is a form of performance-based contract where a vendor is paid a pre-determined fee or hourly rate contingent on meeting one or more service-level metrics. For example, a network management services contract might specify a certain amount of uptime or some other network availability.
metric. Help desk services might require that help desk calls are answered within a certain timeframe, and that problems will be resolved within another timeframe, based on severity of the issue. Program management offices can be measured by resource allocation and usage, project completion, or other metrics.

All these examples include penalties for failure to meet the agreed-upon metrics. Penalties vary from contract to contract but typically involve a forfeiture of fees or a reduction or elimination of performance bonuses. In some cases, the failure needs to occur over more than one time period before penalties kick in. For example, if the government outsources network management and then chooses to upgrade its network, there may be a dip in vendor performance for some days or weeks after the upgrade. Vendor performance may return to normal levels once the upgrade has stabilized. Vendors will understandably want to have allowances for such events accounted for in the service-level agreement.

SOFTWARE IN ESCROW

From the mid-1980s to the early 2000s, an escrow agreement was considered a necessary protection for a government entering into a software contract. The rationale was simple — with many software firms being acquired, changing business direction, or simply going out of business altogether, having the source code in escrow would ensure that the government still had access to the software. In theory, the software system could still be supported and used even if the vendor was no longer in business.

Software firms are still being bought and sold, so organizations still need to protect themselves against a vendor being acquired or getting out of a certain line of business. Unfortunately, changes in the nature of software technology mean that putting software in escrow is no longer an effective way to mitigate that risk. Systems in the 1980s and even into the 1990s generally featured large libraries of computer code that anyone with the requisite training in that particular language (and with sufficient documentation) could, in time, come to understand and maintain. This is what made escrow agreements effective. But computer programming is much different now than it was 20 or even 10 years ago. There are separate layers of code objects for transaction processing, user interfaces, web interfaces, network interfaces, and other constructs, so it is no longer a relatively straightforward task to look at a data model and begin to maintain code. Further, advances in program and database design have made the overall computing environment quite dynamic.

As a result, it is simply not as feasible as it once was to expect internal IT staff to take over if the software manufacturer goes out of business or is acquired, so putting source code in escrow does not offer the level of protection it once did. Escrow agreements remain beneficial for relatively small applications, but not, generally, for enterprise-level software. Governments should therefore investigate the stability and market presence of software and hardware providers before signing a contract. Reliable information on acquisitions and divestitures can be difficult to obtain, but governments need to conduct due diligence into market presence, investment in research and development, and other factors. As with any contract, clear expectations and mechanisms for dealing with unexpected events are essential.
comes to technology contracts and projects. Road or building construction contracts require oversight, reviews, and approvals from internal staff, but technology projects often go much further. Staffing for technology projects is often more than 50 percent internal employees in all phases of work, from initial planning through solution deployment and support. As such, it becomes difficult, if not impossible, to separate responsibility and determine if there was a performance breach on the part of the contractor. Collecting on the bond becomes extremely difficult when the government’s staff is heavily involved in producing the deliverables.

For example, on an ERP project, a vendor accused of poor performance will often respond that government staff wasn’t available, the government didn’t provide needed information, the government didn’t make decisions in a timely fashion, and so on. Because the contract and statement of work identify government participation in each phase, the vendor can often create enough questions about responsibility to make it very difficult to collect on the bond.

CONCLUSIONS

IT contracts are inherently risky, and the finance officer plays a crucial role in ensuring that such contracts include protections for the organization and mitigate that risk. A number of tools are available to do that, and some are more effective than others. In fact, tools that were commonly used just a few years ago have become less effective as the technology industry has changed.

The GFOA advocates the use of deliverable-based milestone payments, warranties based on requirements, and service-level agreements as the best ways to provide the necessary protections. Of course, there is no way to completely eliminate risk, but with these tools, governments can feel more comfortable that they can achieve their goals.

David Melbye is consulting solutions manager for the GFOA in Chicago, Illinois. He can be reached at dmelbye@gfoa.org.

Simply indicating that a requirement can be met does not guarantee that the software or hardware actually meets that requirement. The government can protect itself by ensuring that responses to requirements are included in the contract as a warranty.