KC01 - COTS

2016년 6월 8일 수요일
오전 10:52

Commercial Off-the-Shelf Software—
Benefits and Burdens

The chief information officer's (CIO) job has never been tougher. Demand for information technology workers is at an all-time high and far exceeds the supply of skilled people. Retention is problematic, with attrition as high as 25-40 percent in many IT companies. Users are demanding ever-increasing quantities of business applications—the pursuit of on-time, on-budget, and bug-free projects remains the only acceptable goal.

How should you approach these application projects? There are many options: buy a commercial off-the-shelf (COTS) product from a vendor and integrate it into your environment, build your own application, or try to minimize your involvement and just use an Application Service Provider (ASP) (a company that offers individuals or enterprises access over the Internet to applications and related services that would otherwise have to be located in their own personal or enterprise computers). Selecting a COTS product seems to be the easy answer. You avoid the lengthy development cycle associated with building an application yourself and the inherent security risks of using an ASP. Why build an application if you can just go out and buy it? It will certainly save time and money. The product will be well tested for you. Integration should be a snap. It solves the users' problem and it's bound to be less expensive too!

So, where do you begin? Well, first you have to find products that address your needs. A Web search can yield quite a few leads. Talking to CIOs of similar companies can be helpful too. So can attending a conference dedicated to solving the business problem you are facing. In most cases, within a few days you have a myriad of options from vendors that range from "garage shops" to multi-billion dollar companies. The price range and the sufficiently different product features make selection nontrivial. Contacting vendors, gathering product literature, and arranging for product demonstrations takes time. At this point it can seem like it would have been easier to just start writing code.

Next, you have to pick the "right" product from the list of possible solutions. (See page 10, "Selecting COTS Products or Services.") As part of that process, identify the technical capabilities that are "must haves" versus those that are "nice to have." This will help focus discussions with the vendor community and allow you to cut through the marketing hype. It is helpful at this time to try and get user participation in the project.

Unfortunately, in many cases the CIO has excellent software application developers on staff, but they may not have the right system engineering skills to lead a product selection. The first step is to eliminate products that don't really exist yet (vaporware)—in most cases it is desirable to acquire a product that has been in production for a while. This allows you to assess how dynamic a product's Application Programming Interface (API) is by reviewing past changes and determining if the vendor has a backward compatibility policy. Next, eliminate products that don't run on your supported platforms and those that clearly don't meet the essential users' requirements, or are out of your price range. This should reduce the number of options to a manageable number.

Now it is time to do some serious work. First, conduct a gap analysis of differences between user procedures and product adaptability. It is reasonable to consider re-engineering processes and eliminating less urgent requirements in order to use a COTS product.

In addition to comparing product functionality, also talk to customers using the product under consideration. Assess how well the product met their needs, as well as how well the vendor performed. Was the vendor responsive to their problems? Was the documentation adequate? Was the product easy to use? If the number of products is down to just a couple, you might even want to bring the products in-house to run them through a scenario-based trial. This allows you to see how well the product works in your environment and how customization may affect your implementation.

Let's assume that your decision-making process leads to a proper decision. You have selected a product that comes closest to meeting your needs from a vendor you believe will be responsive to you during the integration and deployment phases. Now you need to focus on getting the product installed and integrated with the rest of your information systems. Most sophisticated applications require at least some data exchange with your other application systems. For example, you might need to pull current user data from your human resources system or valid project data from your financial system. This data must be kept synchronized for the entire system to behave properly. The simplest case to deal with is one where the new application requires a “downstream feed” from one or more of your other information systems. In this case, the users interact with the new application and no data needs to be returned. In more complex situations, the applications must operate together, concurrently using the same resources and jointly meeting the performance constraints. Unfortunately, there is no silver bullet. No two integration efforts are exactly alike. The one thing you can count on is that integration costs both time and money. In many cases internal development may be unavoidable, as custom software is required to “glue” the various COTS components together. This phase of the project requires intense system-engineering skills. Lack of planning and proper risk mitigation (see page 3 "Understanding Risks") has resulted in many COTS integration efforts being late, over budget, or judged by users to be a failure.

So now you’ve got the product integrated from a technical standpoint. Don’t make the mistake of thinking that you are off the hook when it comes to testing. Vendors are responsible for testing their products, and sometimes they don’t do that very well. You have the responsibility to make sure that the system works as desired. Debugging is made more difficult because each COTS system is in essence a black box. You can only make inferences about the product by observing component behavior. Even with vendor support to address compatibility issues, when things go awry, vendors tend to “blame” each other.

Assume that you made it through the system test phase. It is now time to plan the product rollout and training for the user base. You have had some users involved in both requirements definition and testing, but even so, this phase will present its own challenges. Planning an enterprise rollout is never simple. Unless we are talking about an extremely simple application, users require training. It will be difficult to get them together for training, and in most cases they will have difficulty with the change. No one likes change. No matter how carefully you tested the application, plan on encountering product bugs. Sometimes the product has more functionality than you need, which makes the application seem more complex. Sometimes, under a full user load, product performance declines. At best, your users will likely request some product customization. Only time will tell if the product really suits your users’ needs and business processes.

You’ve got the product installed and tested; your users are trained and they are using the product. Unfortunately, it’s still not clear that your project is a success. You have to deal with the application lifecycle. Sometimes the vendor you pick may go out of business or lose market share or be unresponsive to you as an individual customer. While you were worried about integration, the vendor was busy

working on its next release. At some point, maybe very soon after your rollout, you have to deal with a COTS upgrade. If the inter-faces are not stable (i.e., written to a robust API), you may have to rewrite them with every upgrade. Upgrades of other products in your information systems environment require you to test all the interfaces. You will have to deal with license renewal and figure out when to replace the product. So, did the CIO make the right choice in choosing COTS? It’s hard to say. It certainly was not as easy as he thought it was going to be. The cost of integration, licenses, upgrades, customizations, testing, training, and rollout are all variable costs that must be compared to the “custom” alternative for a true picture. In truth, COTS integration efforts are less understood than software development efforts and can fail in

more than one phase:

Selection—Did I pick the right product?

Integration—Is the “fit” right? Do I have the skills to get the job done?

User acceptance—Can they get their job done? Is it easier than the alternatives?

Factors that determine how successful integration will be include the following:

Skill in ”integrating” vs. development
Ability to test ”systems” vs. products
Determining alternatives
Establishing reasonable expectations with regard to cost and schedule

Weighing the benefits versus the burdens when deciding whether or not to make the leap to COTS requires decision makers to make hard choices. Choose wisely.

Understanding Risks AlleviatesCOTS-based Systems Woes

The wise manager will recognize risks that are specific to the integration of multiple products into a COTS-based system and will take actions to mitigate and control those risks. There are three major sources of such risk:
Operational requirements—Risks associated with using a COTS-based system to meet the functionality and performance requirements of the users.
Technical approach—Risks associated with the technical characteristics of the COTS products and their impact on the environment into which they must be integrated as components of a system.
Business strategy—Risks associated with the vendor of the COTS products, the need for continued availability of support for the products over time (see page 6 ”COTS Software Licensing”), and the funding pro-file over the life of the system.
Among the important techniques for con-trolling risks are the following, which are interrelated.

Market research

The rapid turnover in the COTS product marketplace can be both a risk and a missed opportunity for the program manager who is unaware of these changes. Market research is a technique for recognizing and anticipating market-related risks during acquisition in order to manage operational system capabilities and technical and business strategies, and to sustain the system as the market changes. It involves collecting and analyzing information on an ongoing basis to determine the availability in the commercial market of products or services that can support operational requirements, system design, and the system itself. A program manager must continually conduct market research in order to track trends in industry standards, the emergence of technologies and products, and the obsolescence of technology and withdrawal of products from the commercial marketplace.

Early and frequent user involvement

One source of risk that managers must address in planning to use COTS products occurs when products that are available will not exactly meet the operational requirements stated by the system users. Early and frequent involvement of the users can result in postponing or modifying some operational requirements in order to have early access to capabilities provided by the COTS products. This kind of negotiation can be achieved by stating mission objectives rather than detailed requirements, distinguishing mandatory from desired capabilities, and providing demonstrations and prototypes for users to assess.

Early and frequent integration

Technical risks center around how well a selected product will perform in the environment provided by the system. For software, the environment is the hardware platforms and configuration, the development tools such as the compiler, and the other software with which it must operate. There are risks that the product will not be reliable, will not meet response time requirements, will not scale up to the system load, and will consume too many resources, and that its interfaces will not be as stated so that it cannot be integrated with other products. Risk mitigation activities include early establishment and use of an integration laboratory for frequent integration and operation of COTS products. This allows the accumulation of actual experience in how well the products integrate and how easily they can be adapted to each other and to the operational requirements. Early and frequent tests can be performed to see how they scale up under realistic loads and how reliable they are. The integration laboratory can be used to demonstrate COTS products to users to gain their acceptance and assure them of the usefulness of the system.

Plannedreplacement and obsolescence

Using COTS products can take away the manager's control over when products are upgraded and which features will be modified. t also introduces uncertainty into the funding profile over the system's life cycle. The manager can mitigate these risks by planning funding and activities to refresh the hardware and software COTS products on a regular basis. This may mean postponing the introduction of new products and upgrades so that the process can be orderly rather than dependent upon release dates. New products should be tried in the integration laboratory to determine their technical and operational impact and possibly pilot-tested in the field on a limited basis. Upgrade strategies must take into account the impact of new and upgraded products on training, installation, and support. A more detailed list of risks and risk mitigation activities can be found at http://www.mitre.org/pubs/edge_perspectives/ march_01/risks.html A guide to maintenance of COTS-based systems can be found at http://www.mitre.org/resources/ centers/sepo/sustainment_support.html