5 reasons why offshoring software product development is radically different from traditional IT offshoring

Sunday, November 16, 2003

John Shinal, Chronicle Staff Writer

5 reasons why offshoring software product development is radically different from traditional IT offshoring

While offshoring IT projects has been going on for more than a decade and is expected to become a $8bn export market for India in 2003, offshoring software product development is far less mature. More than 15 of the top 20 ISVs have offshore development centers in India, but only a handful are leveraging the Indian operations to create a strategic advantage.

Offshoring software product development is inevitable, much for the same reasons why offshoring IT project has been so successful. 70-80% of the development effort in most mature enterprise software companies is around maintenance activities, such as bug-fixing, service patches, upgrades and minor enhancements. These activities are easier to transition to an offshore development team. However, the risks for a software product company are very high.

Most offshoring consultants or vendors often do not distinguish between offshoring IT projects or software product development. They usually propose a one-size fits all model, not recognizing that while an IT project is a discrete stand alone project, product development is an entire manufacturing process. At TechStrategyPartners (TSP), we focus on helping software product companies on developing their offshore plans and helping execute them. In this article we share our perspectives on why offshoring product development is radically different from traditional IT projects.

Why the scope is typically far more complex?

Software products generally are comprised of multiple modules or packages. Unlike IT projects, these need to be maintained over multiple releases at the same time. For example, ERP vendors like SAP still support multiple versions of R/3 [launched in mid 1990s]. This results in a 'SKU explosion' and multiple source trees. In addition, these releases often support different combination of hardware platforms, databases, operating systems, and even different release levels of related modules and execution environments. The offshore vendor needs to understand not just the individual modules but also the various release and platform combinations.

The volume of maintenance activity with packages is usually orders of magnitude higher than traditional IT projects. Products typically have thousands or even tens of thousands of customers with hundreds of thousands or even millions of end-users. That typically means end- users will be reporting orders of magnitude more bugs than with an IT project. While an IT project may produce several dozen bugs per week, a complex product line can produce hundreds of bugs per week or even more. When trying to manage an entire installed base as opposed to a single or even a small number of instances of an IT project, the development, testing, and delivery complexities are orders of magnitude greater.

Finally, mature software companies build their core intellectual property through decades of development, often combining different generations of programming methodologies, languages and architectures in a single product. It is rare for a software company to throw away decaying code or rewrite code. Matters are only made worse by the lack of proper documentation. This makes transfer of knowledge to the offshore team challenging.

Why the development process is typically far more complex?

The process of software product maintenance is inherently more complex than IT projects. Software products essentially go through a "manufacturing" process from bug report to fix distribution to thousands of customers with tens of thousands of production "instances" of a system.

This manufacturing process involves making sure a fix works across multiple release levels. That means regression or integration testing of a fix across releases. Testing on all certified platform and topology combinations follows for each release before the software vendor can distribute the fix to customers. Offshore vendors maintaining IT projects, by contrast, will simply stage a fix for testing on a special server, or a partition of a development server, before moving it onto the production server. Here, instead of thousands of customers on many release levels, there is one customer with one or a few systems. Many software companies have an entire additional layer of complexity on top of this. They might have a technology or runtime layer underneath their applications that creates another set of permutations.

Offshore vendors must also understand how to manufacture service packs from multiple fixes of a single release. Although this simplifies the customer's process of applying fixes, it requires the offshore vendor that produces them to go through another integration and platform certification test cycle. Finally, every new release necessitates training the offshore team to learn new code and simultaneously manage higher volume of problem tracking (PT) requests typically associated with any new release.

Why the supporting infrastructure is more complex?

Because software vendors have to go through a much more rigorous manufacturing process as outlined above, offshore vendors typically greatly underestimate the infrastructure requirements. Typical IT projects have development, test, and production systems. But a software product vendor may maintain dozens of instances, each with several servers. These might support call tracking, development, unit testing, source code control, build and integration testing, platform configuration testing, and customer distribution.

So when an offshore vendor with nothing more than IT project experience proposes putting a development server at their facility, they fall short of providing an infrastructure plan. An infrastructure plan must take into account the trade-offs between several alternatives. For example, servers could remain on-site and with the offshore vendor connecting over leased lines. Or, if workstation-based tools require too much bandwidth, they might consider a remote desktop tool such as Citrix. If this suffers from too much latency, some of the servers in the maintenance process chain might move offshore. But great care has to be exercised if the software vendor is going to create a second master of some or all of the source tree. Although this may save bandwidth, it necessitates the greatest changes to the current development process.

Why the vendor sourcing process is different?

Vendor selection has to be based not just on their technical and domain competencies but also an assessment of business conflict and intellectual property protection. Most of the Indian vendors today have relationships with multiple software product companies in any single space. These relationships vary from implementation partner, re-seller, peripheral product development partner, and strategic development partner. More over, several Indian vendors, such as TCS and Infosys have product offerings of their own. Signing up a vendor who has a strong relationship with your competitor is a clear recipe for disaster. Finally, the Indian vendors are on their way to become full service systems integrators. Given the ‘healthy tension’ between software and services vendors, it is important for software vendors to take a longer-term view before providing some of these vendors access to their core IP.

Why the cultural challenges are different

Development is the heart and soul of a software company. An IT shop, by contrast, is a support function and usually is less central to the company's vitality and ability to deliver products or services. Any software company's management team proposing to offshore some part of development is going to encounter strong resistance from the core of their organization. The idea that a new team of people can come in and take over responsibility for big parts of a product or process when the home team has taken years to master it is likely to be anathema.

Forcing the organization to accept an offshoring strategy as a fait accompli is unlikely to work. Its members are the source of the product and process knowledge that are required to make any transition work. Helping team members come to grips with the situation has to start with an honest discussion of the competitive reality, the business imperative for offshoring, and the company's offshoring strategy. It also requires a targeted plan for gaining the support of the pivotal developers. Enlisting their active support means offering the best performers new challenges after the knowledge transfer is complete. If maintenance is going offshore, the best and brightest devoted to that task should have the chance to work on new development or other tasks that were previously resource constrained.

TSP's approach

TSP works with ISVs to develop and execute their offshoring strategy. We work seamlessly with our clients across the entire lifecycle of an offshoring initiative to make sure that our clients get the results. This includes (1) Crafting the overall offshoring strategy; (2) Developing the business case and implementation game plan; (3) Selecting the vendor and structuring the contract; (4) Getting organization and vendor ready for execution; and (5) Helping execute and capture benefits.

Our aim is to help clients do offshoring right, do it quicker and maximize their payback from this initiative. We add value to our clients in the following 4 ways

1. Advise based on our experience and fact-based analysis on how to make the difficult decisions
2. Frameworks, templates and models that help shrink the time for launching the offshoring initiative
3. Benchmarks across various client situations on financial and process metrics that help validate the assumptions
4. Assistance at various stages of the process with the analysis and fact-finding, e.g., site visits, RFI evaluation, etc.

For further information, please contact contact@techstrategypartners.com