



How to manage the
outsourcing relationship for
maximum business value

Table of contents

- Executive summary** 3
- The benefits and risks of strategic sourcing** 3
- Model 1: Outsourcer validates quality and performance** 4
 - Advantages 5
 - Limitations 5
- Model 2: Client validates quality and performance** 5
 - Advantages 6
 - Limitations 6
- Model 3: Third-party validation** 6
 - Advantages 6
 - Limitations 6
- Model 4: Integrated Model** 6
 - Critical role of the Center of Excellence 6
 - The Center of Excellence (CoE) Model 6
 - Advantages 6
 - Limitations 6
- Choosing an outsourcer** 7
- Summary** 8

Executive summary

Whether your company sees IT as a strategic weapon or simply as a “business enabler” or cost center, the outsourcing model has tremendous appeal. By outsourcing IT projects such as quality and performance testing, your company can increase the strategic value of its internal IT organization and cut IT operations costs at the same time. According to Gartner, the total ITO (IT Outsourcing) market worldwide is projected to have a compound annual growth rate of 7.4% from 2005 to 2010. ITO accounted for \$216 billion of end-user IT spending in 2006. (Source: Gartner, **Hype Cycle for IT Outsourcing**, 2007.)

Yet, along with the benefits of outsourcing come added complexity and risk. And when it comes to application quality and performance, any risk must be carefully understood, measured, monitored, and mitigated. The cost savings achieved from outsourcing testing and other processes can be more than offset by the business cost of frustrated customers and end users due to poor-quality applications.

Without compromising on application quality and performance, your company can reap the rewards of outsourcing, by tackling a host of issues which include:

- Addressing who owns responsibility for quality and performance—the client or the outsourcer
- Ensuring that outsourced IT initiatives are tested and deployed to meet the same quality and performance standards as in-house applications
- Governing a mixed portfolio of in-house and outsourced IT projects to keep them in alignment with business goals

- Avoiding inconsistencies and redundancies in processes
- Determining the service-level agreements (SLAs) and how they are to be measured

This paper helps you answer these questions and provides guidance on how to establish “strategic sourcing” relationships that deliver maximum business value. It presents four basic models of outsourcing and describes the benefits, trade-offs, and limitations of each. The paper is intended to serve as an overview of your options for establishing a relationship with an outsourcing partner. It is not an in-depth examination of the many organizational, contractual, and implementation issues that will need to be addressed subsequently.

The benefits and risks of strategic sourcing

In recent years, IT budgets have shrunk and many enterprises have turned to outsourcing IT projects and business processes to reduce costs. Many of these companies have discovered that outsourcing not only delivers financial benefits, but also offers strategic advantages. Some of the benefits are listed here:

- Outsourcing can reduce the burden on overworked, overstressed IT staff, resulting in better results on critical projects.
- Outsourcing can enable the internal staff to make the transition from “fire-drill” mode to proactive mode, allowing them to focus on higher-value applications and services.

- Preparing a project for outsourcing (in some cases) can actually serve as a catalyst for “getting your house in order,” requiring the IT staff to formally define and document processes, metrics, and best practices.
- Outsourcing, in other cases, can help improve internal IT operations by introducing industry best practices and processes.
- Outsourcing can allow certain distributed development projects to proceed at an accelerated rate due to time zone differences, since there is always someone working on the project.

As a result of these benefits, more and more companies have been experimenting with “strategic sourcing”, optimizing the mix of outsourced, offshore, and in-house applications. However, as many enterprises have realized, outsourcing can also add complexity and risk to application projects, and the advantages of outsourcing can backfire if the new challenges are not addressed immediately. Some of the common challenges faced are:

- **New skills:** Managing the outsourcing relationship requires additional skills and processes beyond what is required for internal projects. Defining performance metrics, escalation procedures, and change management processes can be difficult—particularly in the face of language barriers, cultural differences, time-zone differences, and communication issues.
- **Garbage in, garbage out:** Outsourcing a problem with the belief that the outsourcer can fix it often results only in a worse problem for the outsourcer, along with higher costs, longer delays, and lower quality of the application.
- **Tracking SLA performance:** Keeping projects on time and within the budget requires that the outsourcer

meets critical SLAs, but it can be difficult to determine who is responsible for defining and measuring them.

- **Immature offerings:** Many outsourcing service providers are new to the market and are relatively untested and unproven. As a result, you need to make sure their capabilities truly match your requirements.

The next sections of this paper examine four models or alternatives for working with outsourcers to achieve benefits, minimize risks, and deliver high levels of application quality and performance.

Model 1: Outsourcer validates quality and performance

Since one of the primary goals of outsourcing for many companies is to reduce the burden on the internal IT staff, it is not surprising that these companies often look to the outsourcer to assume responsibility for virtually all aspects of the project. This includes testing and validation of application quality and performance.

In this model, the client and outsourcer agree upon specific targets for quality and performance, along with specific key performance indicators (KPIs) to be measured. Typically, the testing and validation processes, practices, and tools used are at the discretion of the outsourcer and the client is concerned only with ensuring that the end result meets the expectations.

A growing number of outsourcing firms are more than happy to offer this model of “turnkey” service, not only because it is straightforward but also because it provides them full control over the end results. However, for clients there are trade-offs, as discussed in this white paper.

Advantages

The key advantages for the client are simplicity and cost efficiency. Requirements are clearly defined at the outset and the cost is contingent on the outsourcer meeting the requirements. Quality and performance KPIs such as unit testing and application-level quality metrics, performance benchmarks are established up front. In many cases specific processes are also agreed upon to drive consistency between the client and the outsourcer. These include processes for handling application changes, managing patches, adding new modules, as well as reporting processes.

In addition, this model provides a strong incentive for the outsourcer to deliver high-quality applications. Since the outsourcer has complete control over the results, the value provided to the client is highly tangible. By meeting the client's requirements on time and within budget, the outsourcer can create additional business opportunities and build its reputation for high-quality service.

Limitations

With this model, the client company does not control quality and there is the inherent risk in turning over such a critical element to an external organization. Since the client company does not control the testing processes or tools used, there is also a risk that the outsourcer will develop processes that the client will later become tied to. And this also gives rise to the question of what happens if this year's outsourcer is out of business the following year.

Moreover, the quality and performance results achieved by the outsourcer are achieved in the outsourcer's environment. In many cases, there are unanticipated issues when the applications validated by the outsourcer are actually moved into the client's production environment. There are often interoperability or incompatibility issues in the shared infrastructure that might require re-testing and re-validating applications in the actual production environment.

Equally important, outsourcing application projects may save money in the short term, but do nothing to build the competencies of the internal staff in terms of improving application quality and performance.

Finally, as a client, you should be cognizant of the fact that it is not always practical from a business perspective to enforce the penalties you have agreed upon for missed SLAs. In some cases, the consequences can actually be financially ruinous to the outsourcer, and enforcing the penalty can be detrimental to your business objectives. Therefore the risk of leaving the outsourcer in control of quality validation can be higher than it appears.

Model 2: Client validates quality and performance

In this model, the outsourcer does much of the application development work, but the client retains final responsibility for quality and performance. The outsourcer transfers the finished code to the client for testing and validation, and it is the responsibility of the client to ensure that it meets specific KPIs or meets the same standards as other in-house projects.

Advantages

The obvious advantage of this model is that it enables the client to keep tight control over the tools and best practices used in the testing process, ensuring consistency in validation standards, KPIs, and expectations for the finished application.

In general, this model works best for projects that have extremely rigorous requirements, involve complex integration testing, and have multiple stakeholders. Examples would include new functionality for mission-critical applications, new customer-facing Web services, and customizations for packaged enterprise resource planning (ERP) and customer relationship management (CRM) applications.

Limitations

In this model, the outsourcer does not necessarily have adequate motivation to provide high-quality code since the responsibility for quality and testing costs is that of the client. In addition, as the outsourcer does not adopt much of the project workload, the resultant cost savings are less. However, there is still the administrative overhead of coordinating the project with the outsourcer and transferring code to and from the outsourcer's team. Also, some of the same disadvantages of the first model still apply: The client must cede a degree of control to the outsourcer during the coding process and may become dependent upon tools and processes used by the outsourcer.

In addition, errors made by the outsourcer's development team may end up deeply coded and may be difficult, time-consuming, and expensive for the client's team to fix.

Model 3: Third-party validation

There appears to be a growing trend toward "multi-sourcing" or outsourcing varied aspects of the same project to different outsourcers. In this model, the client contracts with one outsourcer for application design/development, and a third-party outsourcer for testing and validation.

Advantages

Third-party validation can be a cost-effective option for companies that have poor or non-existent testing processes, or which have very few software projects. Through this model, they can take advantage of the specialized expertise of outsourcers and avoid the expense of developing coding/testing/validation capabilities internally.

Limitations

This model has all the disadvantages of the first two models—loss of control over processes; risk of dependency on the outsourcer’s tools and techniques; lack of consistency from one project to the next; the potential for unanticipated problems that can occur once the validated code is actually migrated into the client’s production environment; and so on. Moreover, while relying on outsourcers to deliver high-quality applications may save money in the short term, it does not help the client company build its own competencies. Sooner or later, growing companies will need to address the need for internal application development and testing expertise.

Model 4: Integrated Model

For companies that wish to take advantage of the cost-efficiencies of outsourcing without handing over control of quality and performance and risking dependency on an external vendor, one additional option remains—an integrated model of outsourcing.

In this model, the client and the outsourcer have joint responsibility for application quality and performance. The outsourcer does the initial testing and validations, ensuring that the code meets the agreed-upon standards. The client then re-validates the code, migrates it into the integrated application, and verifies that the end-to-end business process meets all requirements.

Critical role of the Center of Excellence

The integrated model requires close coordination between the outsourcer and the client. Ideally, the client’s internal team and the outsourcer agree upon standard tools and best practices in order to simplify and streamline the testing and validation processes. The use of the Center of Excellence (CoE) approach can greatly simplify this effort.

Simply put, a CoE is an internal organization focused exclusively on optimizing application characteristics

such as performance, quality, and availability. It provides a management and automation platform for processes, consulting and support services, as well as leadership and advocacy to help the organization optimize these attributes. Through the CoE model, project teams are able to take advantage of the expertise, toolsets, and best practices the CoE has developed and consolidated.

A CoE can also provide a central platform for asset management that will create visibility into quality and performance parameters of the delivered application. This helps keep everyone informed and also aids in keeping applications aligned with business objectives.

By applying the CoE model to the outsourcing relationship, the client company can collaborate and share critical information with the outsourcer rather than simply hand over tasks. The CoE model functions as follows:

- The client’s application delivery team and the outsourcer’s team can share a common application delivery dashboard that presents KPIs across all quality and performance projects—providing “one version of the truth” to all project stakeholders through real-time visibility into the health of applications and the infrastructure.
- The teams can share test assets (scripts, files, and best practices) for higher consistency, repeatable processes, and faster time-to-market with high-quality applications.
- Each team has control over and responsibility for the quality and performance of the finished application and the management has visibility into and control over the validation process.

Advantages

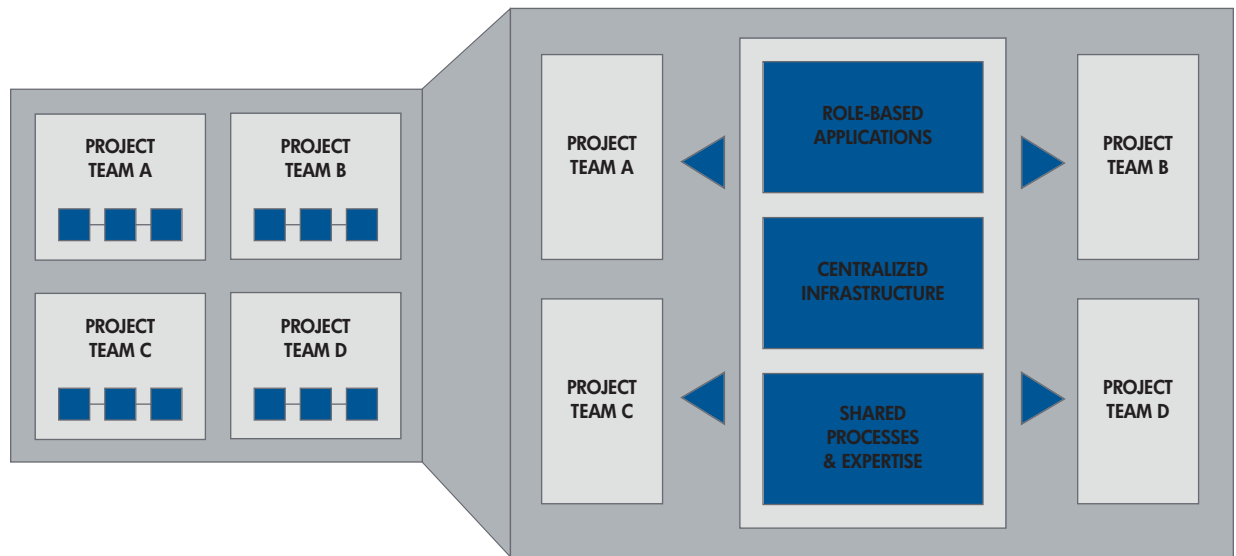
The integrated model establishes a true partnership between the client and the outsourcer and is likely to produce better quality and performance results than any other model. It enables the client and the outsourcer to verify that the application meets quality and performance standards with consistent use of KPIs and validation processes. The client can also quantify how well the application meets the original business objective.

Limitations

If strategic sourcing of application development and delivery is not a pervasive strategy for the client company, the integrated model may not prove to be as cost-effective as other models of outsourcing.

The Center of Excellence (CoE) Model

Figure 1: shows a traditional Center of Excellence Model



In the traditional model, every project team is an island, with its own staff, tools, and practices. The CoE model centralizes the expertise, processes, and sharable assets.

Choosing an outsourcer

Irrespective of which aspect of application quality or performance you are considering, it is important to evaluate each prospective outsourcer on the following criteria:

- **Experience:** Examine the outsourcer's overall business experience as well as specific testing methodologies, tools, and procedures. Ask for specifics about which QA personnel would be working with you and examine their work experience carefully.
- **References:** The outsourcer should be able to demonstrate a track record of success with your specific type of project. In some cases, the companies who serve as references may not agree to be contacted by the outsourcer's prospective clients due to privacy considerations. However, you can request to see the outsourcer's records with metrics relating to quality improvements.
- **Certifications:** Make sure the outsourcer is certified on the products they will be using to conduct testing and validations. For instance, HP Software certification requirements are very stringent, so you can have a higher degree of confidence in vendors who have achieved HP Software certification.
- **Privacy guarantees:** The security and privacy of your data are extremely important considerations. In the United States, legislation is now in process that will mandate stricter controls on how nonpublic information is handled by outsourcers. Your company should consider creating a test data group that provides depersonalized test data for your applications; outsourcers should in turn be willing and able to work with the test data group and offer optional tools to help create depersonalized test data.
- **Partners:** In many cases, a single outsourcing provider cannot provide every service required by the client, and "multi-sourcing" or involvement from additional service providers is required. Investigate the additional resources your outsourcing candidate has access to ensure you get the full spectrum of services/providers you need.

Summary

As distributed development and outsourcing become “business as usual,” it becomes increasingly important to understand the options for dealing with outsourcing service providers and the risks, rewards, and trade-offs of each option.

Traditional models of outsourcing have led to the “throw-it-over-the-fence” syndrome, where the client’s team and the outsourcer’s team are focused only on their assigned tasks and their specific responsibilities. The integrated model of outsourcing described in this paper is the first approach focused on creating a true partnership between the outsourcer and the client. It is also the first that lays emphasis on the business process and the business requirements rather than the tools and techniques used to validate quality and performance.

To learn more, visit www.hp.com/go/software

© 2008 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

4AA1-7347ENW, January 2008

