Payoff

Successful outsourcing experiences are not based on the levying of penalties for failure but on the accrual of expected benefits by both parties involved in the outsourcing agreement. For this to occur, IS managers must implement a proactive, forward-looking oversight mechanism designed to ensure that the outsourcing provider operates in a performance zone that provides the expected business value. Outsourcing oversight metrics—key performance-monitoring parameters built into the outsourcing agreement and assessed on an on-going basis—form the basis of such a mechanism.

Introduction

As outsourcing expands into more areas, its dimensionality is changing as well. At first, outsourcing activity focused on the data center, followed by networks, the telecommunications infrastructure, help desks, and workstation support. The latest area of heightened activity appears to be in the area of applications development and support.

It is safe to say that outsourcing is exploding. Yet because outsourcing is also imploding in some cases, it is critical to leverage lessons learned from the dynamics of outsourcing agreements now in progress.

The most startling lesson is that good intentions and a 600-page contract are not enough to ensure the success of outsourcing agreements. The corollary to this lesson is that a successful agreement requires that all parties to it be successful. In other words, squeezing the last dollar out of a provider by negotiating a tough contract is not in the best interests of the overall success of the deal.

Another key lesson stems from the propensity for change. Although outsourcing agreements are typically specified in multiyear terms, flexibility must be built into the arrangement. In addition, most agreements contain backward-looking performance measures that specify penalties and rewards after the fact; if predetermined targets are not hit by the end of some time period, money changes hands. Such systems are, by design, sources of frustration and friction. The true lesson is that no organization wants to end up in the penalty box. An agreement is successful if objectives are met, not if penalties are paid.

Quite simply, outsourcing agreements must specify results and expected outcomes in basic business terms and contain the tools for continuous assessment of whether the agreement is operating in a zone that provides expected levels of performance and value.

From this perspective, the age of outsourcing oversight through metrics has definitely arrived.

The Basics of Metrics

IS managers and their organizations need to become aware of the potential to use measurement not only as a mechanism for looking backward to determine if outsourcing performance targets have been met, but also as a proactive navigational tool to support the achievement of performance improvements. In addition, the success of an outsourcing agreement goes beyond technical improvements in performance; in some manner, gains in productivity, quality, and service must be linked to business results.
Companies embarking on outsourcing should apply measurement through the following types of activities:

- **Contract-initiation actions.** These actions include baselining today's performance and benchmarking current performance against industry performance.

- **Contract goal-seeking activities.** Such activities include designing and implementing a measurement system that provides the necessary telemetry (i.e., metrics collected as part of natural work processes) to enable the parties involved in the agreement to actively monitor, manage, and navigate to performance targets from both technical and business perspectives.

- **Performance target-attainment actions.** These actions involve:
  - Continuous measurement support for the duration of the agreement in the form of real-time updates of performance data as it is generated; in other words, creating a continuous, forward-moving baseline.
  - Real-time monitoring of position relative to the attainment and movement toward contractual goals.
  - Periodic (i.e., monthly) formal management reviews on progress.
  - Quarterly benchmark analysis against external industry performance.
  - Production of an annual baseline/benchmark report as required by the contract.

This action-oriented approach focuses on five principle activity areas or workstreams targeted at:

- Rapidly establishing a baseline.
- Providing an external calibration benchmark comparison.
- Designing an overall measurement system that supplies the parties to the agreement with the measures to both assess contractual conformance and proactively manage performance in real-time toward contract targets.
- Implementing the measurement system.
- Providing continuous support, monitoring, reporting, and coaching for the duration of the agreement.

Workstreams one and two comprise the contract-initiation actions and are executed concurrently with workstream three. Workstream three focuses on the contract goal-seeking activities. By maximizing the concurrency between workstreams one, two, and three, short-term metric needs are met while the development of a more complete, long-term measurement system is under way.
Workstream four puts the measurement system into action, and workstream five makes use of it to steer the organization toward its goals while providing contractually dictated reporting and analysis.

Exhibit 1 depicts the design methodology for outsourcing oversight. The approach is neither universal nor a boilerplate, but designed instead to focus a forward-looking process of determining and enacting the metric elements critical to the success of an outsourcing agreement. By incorporating technical measures and their business linkages into the overall management structure, all parties to the agreement better position themselves to harvest the benefits of their outsourcing strategy. The methodology's synergistic use of multiview metrics (i.e., IT, internal business, external business, and external benchmarks) and flight-deck approach provide a model for effective outsourcing management.

**Design Methodology for Outsourcing Oversight Metrics**
<table>
<thead>
<tr>
<th>Workstream</th>
<th>Activities</th>
<th>Outcome(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rapid Baselining</td>
<td>Construct list of measures needed as perceived by both parties. Inventory</td>
<td>Metric baseline of current performance established in minimum time frame.</td>
</tr>
<tr>
<td></td>
<td>currently available data/assess data quality. Analyze contractual agreement in terms of business value and goals/assess measurement coverage. Create baseline.</td>
<td></td>
</tr>
<tr>
<td>Benchmark Analysis</td>
<td>Identify industry groups/company types to be included in benchmark. Construct benchmark analysis by overlaying baseline data on external performance data. Analyze and report on current performance vs. external performance.</td>
<td>Assessment of degree of stretch of contractual goals; potential for identification of additional performance improvement opportunities.</td>
</tr>
<tr>
<td>Measurement System Design</td>
<td>Link internal IT measures to business outcome measures. Identify progress/navigational measures. Create overall measurement system incorporating contractual performance measures, progress/navigational measures, linkage to internal business measures, linkage to external business measures. Identify and define all measures, data sources, collection and analysis tempos, and end uses. Create systems dynamic simulation (optional).</td>
<td>Organizational IT flight deck designed for monitoring and managing contractual performance parameters from both technical and business perspectives; expectations of business outcomes of the agreement fully charted; simulation environment available for what-if analyses (optional).</td>
</tr>
<tr>
<td>Implementation</td>
<td>Define roles and responsibilities for enacting measurement system. &quot;Wiring&quot; the organization.</td>
<td>Mechanism for contractual reporting and proactive performance management in place.</td>
</tr>
</tbody>
</table>
The Outsourcing Oversight Control Panel

The basic oversight control panel contains gauge clusters addressing nine critical areas of outsourcing oversight. These areas are equally applicable to all dimensions of IT outsourcing, from the data center to desktop management to applications development and support. The nine clusters are as follows:

- **Finance/budget.** The focus here is on cost management and on-cost delivery of services and work products.

- **Customer satisfaction.** This area focuses on the critical attributes that generate satisfaction with IT services and work products among internal business customers.

- **Work product delivered.** Here, the focus is on quantifying the amount of service or work product provided in a given time period.

- **Quality.** This area focuses on the objective and measurable aspects of quality of services and products.

- **Time/schedule.** The time/schedule cluster focuses on critical service, product, and project time frames and the ability to deliver on-time.

- **Business value.** This area measures the outsourcing agreement's outcome attainment from the financial/shareholder view, external customer/marketplace view, organizational learning and improvement view, and internal process improvement view.

- **Operational service levels.** The focus here is on critical service tempos, availability, and delivery of work products.

- **Human resources.** The human resources cluster focuses on changes to the skill inventory and internal job satisfaction.

- **Productivity.** Here, the focus is on the efficiency of the production and delivery of work products.

Two classes of measures are associated with each gauge cluster in the control panel. First, as a direct by-product of the creation of an outsourcing agreement, each gauge cluster must have target or destination measures that indicate the goals of the agreement. The measures may focus on a single value or on multiple values each linked to a point in time.

Second, each gauge cluster must have rate measures that focus on the direction and rate of movement toward the targets. These direction and rate measures are, in fact, the navigational measures essential for monitoring.

These ideas can be applied in the context of applications development outsourcing. More specifically, they can be applied to measuring the throughput performance of an IT organization.
Measuring Throughput Performance in Applications Outsourcing

Measuring the amount of work done by an applications development and support organization and the amount of product produced or delivered by such an organization has proved to be both difficult and elusive. Such measures are critical to outsourcing, however, because the most commonly asked executive questions in the world of IT are related to them. Examples of these questions include:

- Are we using our IT organization effectively?
- Is the IT organization efficient?
- Are we spending the right amount on IT?
- Are we getting value for our IT dollars?
- Are we doing more work than last year?
- Are we doing more with less (or less with more)?
- Are we doing the right work?

Suppose the CEO of an organization asks the IT director or CIO whether the outsourced IT organization did more work than last year or how much more work it expected to do next year versus this year. How does an IT executive obtain a quantitative answer to these questions?

Many executives typically attempt to solve the problem by applying function points. Although function points may be useful for quantifying the work product size and change in size for a particular set of classes of systems, they do not cover all aspects of work. IT organizations provide user support, for example, as well as help desks. They may also be using technologies (e.g., object orientation) for which function points do not apply. Function points do not even pick up small maintenance tasks that involve computation changes. In addition, technology updates (e.g., moving from one data base or operating system to another) also pose a challenge.

Clearly, function points have a low capture ratio; in other words, the amount of coverage of IT work types accurately counted by function points is low. Lines of code (LOC) as a metric faces similar problems. Finally, in the world of objects, specialty metrics such as metrics for object-oriented systems environments (MOOSE metrics) do not capture all the territory either.

There does not seem to be a way to use a single metric to express the amount of work done by an IT organization. There also does not seem to be a way to express the amount of work product produced by an IT organization. If neither of these two aspects of work can be computed, then overall efficiency cannot be computed and analysis of effectiveness is problematic.

By accepting the nature of the problem, however—that an IT organization performs many different work types, each of which may have its own natural sizing measure—the problem can be solved.
Work and Work Product Measurement

Focus for the moment on the notion of throughput. Throughput is defined as the amount of material put through a process in a given period of time. It can be viewed as something that is discernible or visible to the customer or user of a process in terms of process inputs and outputs. From the customer's vantage, these inputs and outputs are really requests and results (i.e., work product delivered in some form to the customer).

Exhibit 2 expresses this view of IT throughput similarly to the way that an application boundary is drawn around an application in traditional function point analysis. What the customer sees represents only a portion of what goes on. Inside-the-box internal work invisible to the customer (who may be paying for it) is also performed.

IT Throughput from the Perspective of the Applications Delivery Function

From a customer perspective, this internal work should not be measured. From an IT organization's perspective, however, the ratio of customer-visible work (i.e., throughput) to internal work clearly relates to the IT organization's efficiency and its overall process yield.

Returning to the definition of throughput shows that the amount of material put through the process in Exhibit 2 is essentially the total volume of work performed. The volume of work performed is a function of the types of work requested and the size of the requested work expressed as some sort of units.

Possible work types or categories of work that may be requested by a customer include new systems development, platform migration, adding of new functionality to existing systems (i.e., adaptive maintenance), improving existing functionality (i.e., perfective maintenance), fixes/repairs (i.e., corrective maintenance), report generation, preventive maintenance, support functions, and production support. It is important to note that work types are not static. New ones may emerge over time.

Internal work types include release-control activities, technology upgrades, and disaster recovery. This list too may grow over time.

Using this simple work type-based model yields two equations.

Total work performed equals: volume of work requested (executed) + volume of internal work. Throughput (as viewed by the customer) equals: volume of work requested (executed).

Computational Aspects of Throughput

The previous equations are unsatisfactory because they are at too high a level to be useful. More detail is needed to compute volume of work requested.

It is worthwhile to review this proposal. First, draw a box around IT and called it the IT/user boundary. Then identify all work types (i.e., requests and results) that move across the boundary and also those that do not.

Using insight gained from the function point measure, it appears that a logical next step and parallel would be to be able to count the number of occurrences of each work type and multiply that number by a weight to get an overall throughput score or volume. This is essentially what is done in function point counting; each function type is identified, multiplied by a weight, and the overall volume computed.
The example, however, lacks weights, and counting the number of occurrences of each work type does not do justice to the varying size of each type requested. This dilemma is remedied by the function point method, particularly the work of Charles Symons on MK II Function Points. This method provides the insights necessary to complete the computation framework for throughput measurement.

The problem should be tackled backwards, by first concentrating on the weights. In the traditional function point measure, the weights used were determined by trial and error. In Symons' method, the weights are calibrated.

For each work type, however, the weight used should be the average delivery rate per size unit of the work type. This, of course, raises the problem of having a size unit for each work type.

The concept of a natural size unit is used to deal with this issue. In simple terms, this is the size unit that best fits the work type: it is discernible and measurable and can be audited. For some work types, the size unit is function points. In other cases, it may be lines of code. In still another case, such as the help desk, it may be the number of hours spent on serving the request.

In essence, computing throughput involves, at a high level:

- Identifying all the natural work types.
- Establishing a size unit for each work type.
- Establishing a delivery rate for each work type to be used as a weight (e.g., hours per size unit).
- Computing the weighted sum of all the work volumes (i.e., size times rate).

Exhibit 3 depicts this process.

Computing Throughput

Readers who have followed this analysis carefully are aware that the weighted sum is essentially total work hours performed. This should not be alarming.

The Work Unit Concept

Throughput can now be defined as the sum of all the work associated with requested work types. From this point on, throughput is expressed as ITWUs (IT work units). Work units for each natural work type are either: (size of work requested) * (baseline delivery rate) or (number of requests for the work type) * (flat rate delivery rate for the work type).

Rates in both these cases are expressed as work units per unit of request size.

Consider the work unit to be an elemental measure of IT work similar to a calorie or the gold standard before currencies could float.

Although a work unit can be defined as virtually anything, for the purposes of this analysis, it is defined as the number of hours needed to convert one function point to operational code. Using the results of the 1995 Worldwide Benchmark Project, which calibrated the productivity of the average US IT developer as 88 function points per person year and 1,824 hours per person year, a work unit is equal to 20.7 hours.
Although almost any number can be used as a basis for work units, there are advantages in relating work units to function points.

Using this concept, any piece of work requiring one hour to perform is the equivalent of .05 work units or ITWUs. In this way, the ITWU measure is applied as sort of an exchange rate. Using this concept, an organization that performs a volume of work equal to 10 ITWUs (or 207 hours) could have either performed that particular work or had an average US IT professional implement a system with 10 function points.

Exhibit 4 provides a sample application of the ITWU computation. In this chart, the metric column identifies the natural sizing metrics used, the size requested column shows the total size of requests for the year, the natural rate size unit shows the baseline rate used for the weight, the work unit/size unit column shows the rate expressed as work units, and the work unit column shows the work units computed for the work type. The last figure in the exhibit is the total ITWUs performed by the organization for the period, which is one year.

Computing Throughput for a Sample IT Organization During the Baseline Year
<table>
<thead>
<tr>
<th>Type of Work Requested</th>
<th>Natural Size Metric</th>
<th>Size Requested</th>
<th>Natural Rate/Size Unit</th>
<th>Work Unit/Size Unit</th>
<th>Work Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Development</td>
<td>FP</td>
<td>5,000</td>
<td>6.56</td>
<td>.3169</td>
<td>1,585</td>
</tr>
<tr>
<td>Platform Migration</td>
<td>FP</td>
<td>2,000</td>
<td>8</td>
<td>.3865</td>
<td>773</td>
</tr>
<tr>
<td>Enhancements (large)</td>
<td>FP</td>
<td>1,000</td>
<td>6.56</td>
<td>.3169</td>
<td>317</td>
</tr>
<tr>
<td>Minor Enhancements (small)</td>
<td>FP</td>
<td>500</td>
<td>6.56</td>
<td>.3169</td>
<td>158</td>
</tr>
<tr>
<td>Minor Enhancements (computational)</td>
<td>LOC</td>
<td>250</td>
<td>.0656</td>
<td>.0032</td>
<td>1</td>
</tr>
<tr>
<td>Adaptive Maintenance</td>
<td>FP</td>
<td>1,000</td>
<td>14.45</td>
<td>.6981</td>
<td>698</td>
</tr>
<tr>
<td>Computation</td>
<td>LOC</td>
<td>250</td>
<td>.067</td>
<td>.0032</td>
<td>1</td>
</tr>
<tr>
<td>Repairs (large)</td>
<td>FP</td>
<td>500</td>
<td>7</td>
<td>.3382</td>
<td>169</td>
</tr>
<tr>
<td>Repairs (small)</td>
<td>LOC</td>
<td>250</td>
<td>.07</td>
<td>.0034</td>
<td>1</td>
</tr>
<tr>
<td>Ad Hoc Reports</td>
<td>Flat rate</td>
<td>200</td>
<td>2.21</td>
<td>.1068</td>
<td>21</td>
</tr>
<tr>
<td>Preventive Maintenance</td>
<td>FP</td>
<td>1,000</td>
<td>4</td>
<td>.1932</td>
<td>193</td>
</tr>
<tr>
<td>User Support</td>
<td>Flat rate</td>
<td>175,000</td>
<td>1.23</td>
<td>.0594</td>
<td>10,399</td>
</tr>
<tr>
<td><strong>Total ITWUs</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>14,322</strong></td>
</tr>
</tbody>
</table>

**Benefits of Work Units as an Outsourcing Measure**

Using the work unit form of throughput analysis forces an organization to define its work from the customer's vantage point. This is essential in an outsourcing agreement. The work types themselves are used as a basis for discussion, and the IT work units are used as a basis for planning and resource allocation.

Work-type analysis allows for multiple metrics to be used concurrently. There is no need to search for a single metric to size all work. If a single metric cannot be found for an individual work type, the work type probably needs to be split. In addition, as new work types emerge, they are added to the computation process.

Productivity/efficiency analysis is simple. When this technique is applied on a year-to-year basis, three things are apparent:

- A catalog of relevant work types has to be developed for each year, along with a sizing of the amount of each. This part of the analysis allows shifts in work to be easily tracked.
For each work type, a baseline rate is established each year to use as a weight. In this way, determining if productivity has changed for a given work type from year to year becomes an easy task.

Most important, the question of overall efficiency is easily answered. For any given year, the baseline rates from the previous year are substituted in the computation to answer the question of how much work (ITWUs) it would have taken to perform this year's work at last year's level of productivity. Alternatively, last year's requests can be recomputed at this year's baseline rates.

The most intriguing potential application of the throughput concept has to do with benchmarking IT organizations and making outsourcing decisions. This process is analogous to hardware benchmarking. Suppose a mix of work is picked. In other words, a set of work types and their associated sizes is selected. If the delivery rates of various organizations are obtained and inserted into the ITWU computation, the total amount of work required for each to execute the mix can be determined. This means that the amount of work each of them needs to perform a standard set of requests can be compared. This is true benchmarking. In addition, organizations using this framework can benchmark themselves on the distribution of work across work types; these differences can even be compared within and across industries.

Throughput analysis also provides a basis for outsourcing agreements involving applications development and support. The methodology has the advantage of being able to quantify exactly what is being committed to in terms of work and acts as a scorecard for contractual performance changes.

Conclusion

The success of outsourcing agreements is fundamentally based on the ability of the involved organizations to place themselves in a win-win situation. Doing so requires that all goals and objectives—technical, business, and otherwise—be visible and agreed to.

By codifying outsourcing objectives using measurement and control panels that include the business value measures, technical performance measures, and underlying change rates and targets in all areas, IS managers position their organizations for success. If they do not use these panels, they are launching their organizations into outsourcing like unguided missiles.

Author Biographies

Howard A. Rubin

Howard A. Rubin is professor and chair of the department of computer science at Hunter College of the City University of New York and an internationally recognized expert in the fields of performance measurement and software process modeling. He can be reached at 71031.377@compuserve.com.
Total Work Performed

Throughput + Internal Work

IT Work Units

New Systems Construction

$\sum \text{Size} \cdot \text{Delivery Rate}$

Systems Migration

$\sum \text{Size} \cdot \text{Delivery Rate}$

Adaptive Maintenance

$\sum \text{Size} \cdot \text{Delivery Rate}$

Perfective Maintenance

$\sum \text{Size} \cdot \text{Delivery Rate}$

Corrective Maintenance

$\sum \text{Size} \cdot \text{Delivery Rate}$

Help Requests

$\sum \text{Requests} \cdot \text{Delivery Rate}$

Ad Hoc Reporting

$\sum \text{Flat rate}$

Preventive Maintenance

$\sum \text{Size} \cdot \text{Delivery Rate}$

Emergency Repairs

$\sum \text{Flat rate}$

User Support Services

$\sum \text{Size} \cdot \text{Support Rate}$