Incentive compensation for information systems departments

by HOWARD A. RUBIN
Hunter College, City University of New York
New York, New York

and

D. L. VON KLEECK
Equitable Life Assurance Society*
New York, New York

ABSTRACT

The search for productivity improvement has reached new levels in the information systems world. Organizations are stockpiling tools such as application generators, program analyzers, test data generators, and workbenches in the frenzied quest for productivity. Many organizations that have taken this tool acquisition route are beginning to realize that tools alone are not enough. What is needed is an approach that motivates the use of productivity technology. In this vein, incentive compensation strategies can be applied to the information systems world in a way that focuses on productivity improvement and couples it with quality considerations. From a management viewpoint, this forces more precise definition of productivity metrics and quality quantification.

* Currently at Management and Computer Services, Great Valley Corporate Center, Valley Forge, Pennsylvania 19482.
The Senior Officer had declared war!

The arsenal was filled with some of the most fearsome weapons known to mankind, and they were now trained on the enemy position.

The molecular program degenerator was aimed at the Source Library.

The program analyzer algorithm was ready to decode the innermost secrets of the Maintenance Conspiracy.

Automated workbenches and regenerators were poised to rebuild the ruins.

A negotiations committee fluent in a new user-friendly language was prepared to discuss peace with Userland.

The Senior Officer called for the attack.

Nothing happened!

The troops had vanished!

An investigation revealed that some of the troops had been lured away by a tribe of headhunters, while others were simply out to lunch.

This story, though an exaggeration, typifies what many information systems organizations are experiencing today. An attack is being mounted on traditional development and maintenance techniques to increase productivity. The chosen weapons are tools that accelerate the powers of the individual to perform critical tasks. The organization arsenals are filling up fast, but the projected benefits are far from being realized.

Why?

THE PRODUCTIVITY IMPROVEMENT TRAP

A common thread of reasoning pervades most attempts to improve information systems department productivity. The typical attack plan is formulated as follows:

The first step is to know the enemy. In the realm of information systems, this translates into understanding the nature of systems and the resources necessary to carry them from conception to implementation to maintenance to obsolescence. From such an analysis, areas of greatest potential productivity improvements can be identified.

Life cycle modeling is the cornerstone of this process and results in a view of a system's life that looks like the following:

<table>
<thead>
<tr>
<th>Life Cycle Phase</th>
<th>Percentage of Effort</th>
</tr>
</thead>
<tbody>
<tr>
<td>Development</td>
<td></td>
</tr>
<tr>
<td>Plans &amp; requirements</td>
<td>2.0</td>
</tr>
<tr>
<td>Design</td>
<td>6.0</td>
</tr>
<tr>
<td>Detail design</td>
<td>9.0</td>
</tr>
<tr>
<td>Code &amp; unit test</td>
<td>8.0</td>
</tr>
<tr>
<td>Testing</td>
<td>15.0</td>
</tr>
<tr>
<td>Maintenance</td>
<td></td>
</tr>
<tr>
<td>Fixes</td>
<td>13.0</td>
</tr>
<tr>
<td>Adaptation</td>
<td>14.0</td>
</tr>
<tr>
<td>Enhancements</td>
<td>27.0</td>
</tr>
<tr>
<td>Redocumentation</td>
<td>4.0</td>
</tr>
<tr>
<td>Efficiency recoding</td>
<td>2.0</td>
</tr>
</tbody>
</table>

Sorting this chart by percentage of effort lays the groundwork for developing attack priorities. The areas with the greatest potential payoff become evident.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Area</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enhancement</td>
<td>27</td>
</tr>
<tr>
<td>2.</td>
<td>Testing</td>
<td>15</td>
</tr>
<tr>
<td>3.</td>
<td>Adaptation</td>
<td>14</td>
</tr>
<tr>
<td>4.</td>
<td>Fixes</td>
<td>13</td>
</tr>
<tr>
<td>5.</td>
<td>Detail design</td>
<td>9</td>
</tr>
<tr>
<td>6.</td>
<td>Code</td>
<td>8</td>
</tr>
<tr>
<td>7.</td>
<td>Conceptual design</td>
<td>6</td>
</tr>
<tr>
<td>8.</td>
<td>Redocumentation</td>
<td>4</td>
</tr>
<tr>
<td>9.</td>
<td>Plans and requirements</td>
<td>2</td>
</tr>
<tr>
<td>10.</td>
<td>Efficiency recoding</td>
<td>2</td>
</tr>
</tbody>
</table>

Assuming that all areas can reap the same potential productivity improvement by use of the appropriate tools, this view suggests working on selected maintenance aspects and testing first.

(As a brief disgression, it must be noted that there are numerous linkages between the various areas. A testing improvement—use of a test data generator, for example—should decrease the repair or Fixes effort required in maintenance. Likewise, uniform coding practices via a program generator might decrease the maintenance burden by minimizing the effort required to understand an existing system. In fact, productivity improvement in one area can be amplified across others. This implies that the long-range focus should be on the development side of the process.)

The next step in planning the attack involves the assessment of organizational strengths and weaknesses to adjust the previously determined priorities. Coupled with the knowledge of which areas might yield the highest benefits, a plan can be formulated.

At this point, a scouting team is usually sent in to survey the terrain, select and test weapons, and make a final choice.
The final step is to train the troops and mount the attack—this is the TRAP.

First, tools alone are not enough! The information systems staff must be motivated to learn to use them effectively and efficiently.

Second, the greatest area of potential productivity improvement in the information systems environment lies with the staff itself and not with the technology they use. Barry Boehm has characterized the cost drivers associated with systems projects. Whereas tools (or the lack of them) may drive costs up by a factor of 250%, staff capability may account for cost swings up to 400%.

The message is clear: Information systems organizations must create an environment that contains the proper tools AND the motivation to apply them. In addition, a mechanism to monitor the success of the whole operation must be in place.

Many organizations believe that they are in the process of creating this total environment. They have been lulled into a false sense of security because tools acquisition is relatively easy when compared with implementing organizational and work habit change. In many instances motivational aspects have not been ignored but are being addressed by productivity measures that are fed back to the staff. Again, this is not enough. Measurement by itself does not supply the motivation for productivity improvement. (By analogy, an overweight person may weigh himself/herself every day, but not take any action).

In this vein, there have been a number of significant corporate examples in which information system development productivity measures have been implemented and later abandoned. The prime reason is that they were being applied in a vacuum—they imposed an additional administrative burden, and they had no perceived value directed toward the staff being monitored by the measurements.

Herein lies the productivity trap. Tools and measurement do not constitute the total productive environment. The way out of the trap is to add and integrate a single missing piece: incentive.

Additionally, the view of information systems organizations as solely development/maintenance shops is not robust enough. With the advent of the information center concept, personal computing, office automation, and data-base administration, a broader view of productivity improvement is needed.

AN INCENTIVE COMPENSATION (IC) PLAN FOR INFORMATION SYSTEMS ORGANIZATIONS

A traditional industrial method for staff motivation (behavior modification) is incentive compensation with the underlying philosophy of rewarding on-the-job performance. Let’s see how this can be applied in the system’s world.

The philosophy of the program is that meeting customer-based performance standards while increasing productivity will be rewarded. In addition, the program itself must meet certain design criteria: auditability, clear cause and effect, minimal overhead, and meaningfulness.

Assume for the moment that a method can be devised to measure the productivity of the many activities of an information systems organization. If the goal of the IC program is improved performance, it seems that staff bonuses should be based on productivity increases. Unfortunately, such a reward scheme is not broad enough to satisfy both the needs of the information systems organization and the business requirement of its customers.

Productivity measurement allows management to answer only the single, but important, question “Are we improving?” with regard to some unit measure of department output. It does not directly address quality and potentially can be gamed so that productivity improves while overall quality goes down. It therefore seems to make more sense to make the performance of the department the focus of our improvement efforts. By performance we mean customer accountability in cost, time, and quality.

To be of true business value the IC program must address the customer’s concerns: Are my products being delivered on time, on cost, and at an acceptable quality level? Obviously productivity measures alone do not address any of these critical success factors. Therefore, the IC program must reward behavior that yields improvement on all fronts. It is on this basis that we must confront the key IC issues: funding, structure, and metrics.

FUNDING

Where does the money come from?

One alternative is to have the staff contribute a percentage of their salaries to an escrow pool and have it matched by the organization. The bonus, if achieved, allows them to retrieve their contribution plus additional funds from the pool. This approach probably would not readily gain acceptance in today’s inflationary world.

A second more palatable alternative is possible. Funding can be based on savings to the organization based on productivity improvements. To size the pool, the organization has to answer the question “How much more money would this year’s work cost us at last year’s level of productivity?” (adjusted for inflation of course). This funding scheme has the advantage that the pool materializes only if productivity has improved and real dollars have been saved. This pool is then allocated to both the staff and the organization itself—they both share in the benefits.

In summary: Funding from direct dollar savings generated by productivity improvement savings will be shared between the company and the department.

STRUCTURE

As pointed out before, productivity alone is not enough to satisfy all the criteria for rating the information systems organization. The basic structural components of the program are performance (which includes accountability and quality) and productivity. They are defined as follows (see Figures 1 and 2):
Performance is oriented toward getting deliverables to the customer on time, on cost, and with acceptable quality.

Productivity is oriented toward lowering the unit cost of producing products and services while maintaining quality at an acceptable level.

While the primary focus is on performance level, productivity improvement is required as well. The following equation, used for departmentwide calculation, summarizes the nature of the program:

\[
\text{Bonus percent} = \text{performance rating} \times \text{productivity improvement}
\]

The performance rating can range between 0 and 1. This implies that performance is only rewarded if the customer criteria are satisfied too.

As noted, this equation is applied on a departmentwide basis. Rewards are not based on what any individual does, but the behavior of the group. Later we will see exactly how this works.

An interesting aspect of this equation is the one to one nature of bonus percent and productivity improvement. Basically, what this means is that a one percent productivity improvement can be rewarded by up to a one percent salary bonus. Operationally it may be wise to impose a ceiling (such as 10%). This approach gives us a clear cause and effect relationship—one of the design criteria.

We will now take a top-down view of the program. Each of the structural components will first be examined in concept and then in detail.

**PRODUCTIVITY**

The basic productivity improvement computation is summarized below:

\[
\text{Productivity improvement} = \frac{\text{last year's unit cost} - \text{this year's unit cost}}{\text{today's dollars}}
\]

Notice the focus of the productivity measure is "unit cost," what we are comparing is the cost of production (be it a system or a service) this year and last. If after adjusting for inflation, our unit cost has gone down, we are improving and in fact have saved the organization money. Productivity in this model is viewed based on the previous year—we always want to be improving.

**PERFORMANCE**

Performance is based on three components: cost, time, and quality, scored by percentage of projects falling within + or - 10% of standards.

For all three we apply the 90/10 rule: 90% of products and services should be delivered within 10% of the established estimate (for time and cost) or acceptance standard (quality). The equation computes the percentage of products and services so delivered.

The cost, time, and quality results are matched against a set of success criteria, discussed later, to arrive at the score, which can range from 0 to 1. The meaning of this is simple. To reap the full productivity bonus, these criteria must be met too.

The Computation Algorithm:

2. Compute productivity increase.
3. Compute total organization savings.
4. Compute bonus percentage.
5. Multiply bonus percentage by salary base for total bonus.
6. Subtract total bonus from saving to get organization share.
7. Distribute bonus to each employee.

**THE METRICS**

The measures needed to support the IC program constitute a set of management metrics for information systems organizations. If properly chosen, they should be meaningful, easy to obtain, auditable, and evolutionary.

To understand the application of the metrics it is first necessary to look at an operational model of a contemporary information systems organization.
For purposes of this analysis we will characterize four basic functional groups of products and/or services applied:

A staff (STF) function, which is the management and administration of the organization as a whole. It does not deliver products or services to the customer community.

An advanced technology support (ATS) function, which supplies personal computing, office automation, information center, and database products and consulting support.

A business application products (BAPS) function, which houses the traditional development and maintenance shop.

A system development services (SDS) organization, which supplies interactive computing services, staffing, and training to the information systems areas.

Although these groups may not fit your organization exactly, in total they represent a wide range of potential products and services (with the exception of the raw-data-processing plant itself).

Metrics in our two structural areas are applied to the organization as shown in Figure 3.

The interpretation of this figure is as follows:

1. Performance measures will be uniform for the areas of ATS, BAPS, and SDS. The STF group has been purposely omitted because it does not supply external products or services.

2. Productivity measurement will be unique for each functional area. As we go deeper into the model, we will find that this is applied on a lower level function basis. The uniqueness of the measures for each function is necessary to ensure that the criteria of clear cause and effect and meaningfulness are met.

3. The STF budget will be allocated across the department for computational purposes.

**ATS PRODUCTIVITY**

The following table describes the general types of measures used within ATS to measure the productivity of personal computing (PC), office automation (OA), data base (DB), and information center specialists (ICS).

<table>
<thead>
<tr>
<th>Measure</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC penetration</td>
<td>$</td>
</tr>
<tr>
<td>OA penetration</td>
<td>$</td>
</tr>
<tr>
<td>DB service</td>
<td>$</td>
</tr>
<tr>
<td>ICS service</td>
<td>$</td>
</tr>
</tbody>
</table>

The productivity measures are of two major types: penetration/$ and service/$. They clearly need explanation as both concepts are new.

First consider penetration. Suppose for the moment that the corporate mission of services such as PC and OA is to penetrate the organization with the appropriate technologies. Peak penetration will be reached when all staff performing all applicable business functions will be using the technology for all applicable work. The job of a PC or OA group then is to fill up this penetration space in the most effective and efficient way. The space and the penetration can be visualized as shown in Figure 4.

A simplistic measure for the penetration productivity of a PC might be

\[
\text{# PC delivered} \times \text{# packages delivered} \times \text{# users} \quad \text{Real cost}
\]

This measure supplies the unit cost or volume delivered per dollar.

It does seem that this can be "gamed" by perhaps just installing a lot of underused PCs. However, such manipulation can be counteracted in two ways. First, the equation can be changed so that the numerator is the sum of the products of packages and users for each PC delivered. Second, on the quality side, the customer's view of the business utility of the delivered system can be rated. A "useless" or "not used" system would score a 0. Productivity would be pushed downward.

The scheme could be made more sophisticated if, for example, the PC group mission were to penetrate first with specific equipment types and particular packages. A multiplying weight could be applied in the equation. For example, if large VISICALC usage were desired by management while word processing on personal computers were not, a weight of 2 might be used to multiply each VISICALC installation; a word processing acquisition would rate a 0 or some small number.
A similar structure for productivity measurement is also applied to OA. Again there is a potential use space; and each work station-user-package combination represents penetration into the space. Weighting could be applied to move the organization in the direction of encouraging electronic mail versus spread sheet work on the OA equipment.

In both DB and ICS, SERVICE/\$ is the chosen metric for productivity. When we use this metric, we are looking at the output of each of the groups in business terms.

For example, the ICS group functions to supply reports to the business community and also builds the spinoff databases to derive them. Simply counting the reports produced and databases built, and perhaps weighting them by desired types or complexity, allows measurement of total output. Dividing this by dollars spent again yields the unit cost. The same technique can be extended to the DB group.

The quality aspect of performance measurement for ATS may seem elusive. The prime concern is that quality remain at an acceptable level or exceed it. As a management practice, delivery of a product or service to a customer should include a postdelivery review. In the outside world, even buying a toaster supplies the customer with a feedback form—why not the same in the information systems world? The key questions involve rating the quality level and business utility of the product or service provided. Remember that performance is a function of both productivity and is only rewarded if quality is there.

The productivity-quality linkage is critical. However, most organizations seem to believe that it will always remain an intangible. Quality measurement, in the context of the IC program, gives new meaning and strength to the quality assurance function, and atrophying arm of most information systems organizations.

**BAPS PRODUCTIVITY**

Productivity measurement in the development/maintenance environment has been the subject of almost frantic activity in the past three years. A number of measures have been discussed in recent conferences and publications: function points/labor period, lines of code/labor period, \% of chargeable expenses, and \$ benefit/\$cost.

BAPS by our definition is the developer and maintainer of information systems. If BAPS is viewed as a production plant, its inputs are customer requirements or repair requests, and its outputs are new systems, modified systems, or repaired systems. Parsing out the repair aspect of BAPS, we are left with measuring the output of a plant producing or recycling information systems.

Contemporary trends indicate the end of the monolingual shop. Hence lines of code measures decrease in meaningfulness. Some argue that the advent of program generators and fourth-generation languages, conversions can be applied to generate equivalent lines of code measures. Unfortunately, this technique has not been accepted as credible by most system developers. What is needed is a productivity metric that in some way reflects the content of an information system in a manner independent of implementation language and technique. Both Albrecht's function points and Halstead's software science measures seem to meet this requirement. Function points are a score of the inputs, outputs, files, interfaces, and inquiry options offered by an information system; they are weighted by their complexities. This score is considered a raw score, which is further adjusted up or down by some operational design factors.

The software science measure of program volume attempts to quantify the contents of an information system on the basis of the total number of bits it would take to represent all the operators and operands required to implement the system. Although this concept seems abstract, it has been demonstrated to have greater power as a predictor of both development and repair effort than does function points. In fact, software science allows the computation of an ideal program volume, which can be used as a base for measuring the quality of a program.

With these two alternatives in mind, a method can be formulated for productivity measurement. Function points and software science are related in that they both use the basic parameters of information systems in their scoring schemes—inputs, outputs, files, etc. Although both treat these parameters differently and consider other aspects of the information system, their common base offers a good starting point on which to build. At the simplest level, an information system content unit (ISCU) can be defined as a score of the basic features of a system—inputs, outputs, files, inquiry types, interfaces, and number of business functions being realized. In this way the ISCU count for any new system can be computed as follows:

\[
\text{ISCU} = \# \text{Inputs} \times 3 + \# \text{Outputs} \times 5 + \# \text{Inquiry} \times 4 + \# \text{Function} \times 10 + \# \text{Interface} \times 8 + \# \text{Databases} \times 9
\]

The weights have been chosen on the basis of the relative complexity of each entity class being considered instead of the complexity of a specific one. For maintenance ISCU computation, an additional weight is needed as a multiplier to prorate the percentage of the entity being modified or reused in some form.

This scheme is evolutionary in that it is expandable in either the function point or software science direction as they become more refined. It also overcomes many of the subjective features of the function point approach, as well as aspects of this approach that negate its utility as a basis for estimation and projection.

The productivity of BAPS can be looked at as follows:

\[
\text{new development ISCU}^* \frac{\text{ISCU}}{\$}
\]

*Information system content units

ISCU's per dollar are used to score both new development and maintenance (with the exception of corrective maintenance).
Again, renewed emphasis is placed on the quality assurance group, whose job it is to monitor the measures and audit scores.

SDS PRODUCTIVITY

The measures used here follow the mold of the ATS group. Interactive computing services (ICS) are defined as being the support services for the development staff. Penetration is again the key. Customer satisfaction and system availability offer a clear base for assessing quality. Two additional functions have been included to demonstrate the completeness of the suggested approach, recruiting (REC) and in-house training (TRN). Both can be viewed on a unit cost basis with regard to productivity.

The equation is as follows:

ICS: MEASUREMENT
 PENETRATION/$
REC: HIRES/$
TRN: STUDENT HRS/$

PERFORMANCE

Performance scoring is based on a set of standards derived from 90/10 rule:

<table>
<thead>
<tr>
<th>Score</th>
<th>Base</th>
<th>STD.</th>
<th>Exp.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rating</td>
<td>75%</td>
<td>85%</td>
<td>95%</td>
</tr>
</tbody>
</table>

The interpretation of this equation is that if only 75% of products and services are delivered within the 10% bandwidth a score of 0 is applied, at the 85% level .5, and at the exceptional level of getting 95% of the products and services within the bandwidth 1.0. Any value in between these is prorated. The baseline for establishing the bandwidth for a given product or service will be specified as part of the project or service initiation request process. All products and/or service supplied to a customer will be estimated at this time. In addition, acceptability criteria relating to quality will also be specified. It will be the function of the quality assurance group to set both long-term overall quality criteria relating to such things as future rate, time to repair, expected cost of operation versus actual cost, etc., and specific requirements for a given product or service.

LOOPHOLES?

At first review some questions arise:
- What happens if a planned project is canceled?
- What happens if an unplanned project is initiated?
- What happens if there is a hiring freeze?
- How can work not completed within a calendar year be treated?

The first three situations are handled in a similar manner. The IC plan is designed to be flexible. Corrections to targets are possible at any time, although they must be audited. The situation of multiyear projects can be managed by either pro-rating completed work by the life cycle phase percentages or scoring projects only at their completion.

Gaming in the face of the IC plan is natural. But again, revitalizing the quality assurance (QA) group and supplying upper management support should furnish the needed controls.

INSTALLING THE IC PLAN

The following tasks need to be carried out prior to installation:

1. Form quality assurance group.
2. Develop performance tracking procedures.
3. Define/refine measures: productivity measure definition, integration, and data flow, and quality baselines.
4. Develop standards for integration and data flow.
5. Sell to management and to department.

The scope of these tasks should not be underestimated. Changes in internal reporting of all facets of the organization may be required. However, if the changes implied by the plan are implemented, a stronger organization should result, driven by a set of management metrics that are both meaningful and complete.

A possible scenario for the first year is as follows: End of first quarter, form QA group. End of second quarter, QA implements measures on data collection. End of fourth quarter, QA publishes baseline findings. The goal for the first year is to install the plan. With this in mind, incentive compensation should be awarded at the end of this period simply for getting the IC plan itself defined and in place. Full operation should begin in the second year.

CONCLUSION

A recent Harvard Business Review article reached the conclusion that most managers rate the current flurry of productivity improvement programs as ineffective. The emphasis has been on the quick fix. Nowhere has this been more evident than in information systems departments, who by now should have learned the lesson that tools and technology are not a sole solution to business problems. Capital investment in facilities, equipment, and tools are the simplest aspect of productivity improvement. A number of more crucial elements exist:

1. Management commitment, support, and involvement
2. Employee relations, support, and participation
3. Effective training

The proposed IC plan offers a comprehensive structure for addressing all of these within the information systems environment and provides a coherent framework for managing and motivating this critical business resource.