# Creating Incentives for Reuse in Your Organization

*By Dr. Jeffrey S. Poulin*
*Lockheed Martin Systems Integration- Owego*
*7Dec2001*

While an organization can easily identify the business reasons to launch a new software reuse program, the cultural changes required to support new software development processes are often much more challenging. Mandating that developers participate in the program seldom works without also providing worthwhile incentives. Creating incentives allows developers to share in the benefits that they have brought to your organization and provides a more immediate return on your reuse investment.

This article suggests ways to implement and manage incentives for software reuse within your organization. It presents the experiences gained in various incentive programs throughout industry and focuses on their common characteristics [Poulin95]. In most of these programs, recognition ranges from simple public recognition to significant financial awards.

A basic incentive program may simply reward *any participation* in reuse. This can provide a low-cost way to motivate developers and generate interest in reuse. Suitable incentives include public recognition and presentation of a token award (i.e., certificate, coffee mug, gift certificate). However, the most successful incentive programs tie the type of recognition or award to the value that the participation has brought to the organization.

To do this, the incentives will closely correspond to the metrics that the organization uses to measure reuse and build its reuse business case.[1]

## Tying Incentives to Metrics

Tying incentives to your reuse metrics does several things. Primarily, it helps bridge the gap between project-level metrics and a developer's daily work. Most developers are indifferent to metrics because the metrics are gathered too infrequently or at such a high level that the developers don't see how their work makes a significant impact. If the developer can see the correlation between "what I did today" and "the metrics you present tomorrow," you will be providing developers with some immediate gratification and, more importantly, objective, supporting data upon which to base recognition for their ongoing efforts.

The idea is a simple one: reward those individuals who are implementing reuse on the front lines of system development and have embraced reuse principles. This approach will also encourage developers to become more familiar not only with the metrics used by your organization, but also the underlying concepts of metrics as a toolset.

[1] Applying the same metrics for both purposes also reduces overall administrative effort.

Incentives may have a fixed value or consist of a percentage of the business value to your organization. However, I encourage you to think beyond the token coffee mug or golf shirt and consider offering your developers participation in one of many popular Reuse Workshops or IEEE Conferences and BOF working sessions. By encouraging participation by team members in professional reuse organizations you can be sure that you will also be encouraging professionalism, infusing enthusiasm, and sustain the process changes that you've implemented. In either case, the incentive program needs to recognize:

- Suppliers, developers who write and contribute quality software or other assets for use by others, and,
- Reusers, developers who use software or other assets that they obtain from external sources (such as a Reusable Software Library (RSL)) rather than develop it again.

We'll address how to encourage each of these in turn:

## Recognizing Suppliers

Authors of reusable components bear an increased responsibility during development due to the added requirements, design, and testing that they must perform. Furthermore, they must provide additional information required by their RSL, such as user documentation, performance information, test cases, and integration instructions. Incentives can help encourage developers to do this extra work.

For suppliers, the incentive program should encourage not just participation but contributions that have significant value to your organization. Otherwise, the organize risks populating its RSL with components of questionable usefulness. To avoid this, the incentives must focus on suppliers that provide highly reused components.

An organization can use any reasonable way to place a value on these "popular" components. For code-based organizations, the *Source Instructions Used by Others (SIRBO)* metric [Poulin97] provides an ideal way to recognize part suppliers. To calculate SIRBO, simply multiply the size (in lines of code) of each component that the developer has supplied by the number of times that it has been reused. The organization can then map the result to a monetary value (e.g., such as multiplying SIRBO by dollars per line of code).

This easy-to-obtain metric helps ensure that suppliers do not take advantage of the incentives by putting useless components into the RSL. Instead, it rewards suppliers who identify widely needed functions, as objectively demonstrated by how many times it is reused. The incentives, therefore, encourage activity that brings the most value to your organization.

## Recognizing Reusers

The decision to reuse a component ultimately results from a business analysis (hopefully based on reuse metrics) that shows a financial savings to your organization. This savings can also be used as the basis for encouraging reusers. As with incentives for suppliers, the incentive for reusers should increase with the level of reuse.

An organization can use any reasonable way to place a value on reusing components. For example, the *Reused Source Instructions (RSI)* metric provides an excellent way to measure the amount of reuse [Poulin97]. To calculate the RSI for a reuser, simply count the number of lines of code in each component that he or she reuses. Calculate the incentive after mapping their RSI to a monetary value.

Encouraging developers to look for and reuse components has the desirable side effect of creating demand for reusable components that, in turn, encourages suppliers to develop useful components. In short, it instills in the developers the same goals that your organization seeks to achieve: a small and robust set of highly useful components.

## Recognizing Teams

As an alternative to individual incentives, an organization can build participation through team recognition. Team recognition can add an element of fun and healthy competition to a reuse environment. A team incentive might encourage accomplishments such as:

- High participation in the reuse program. As with the individual participation award, this makes a good introductory incentive to recognize teams that have a high percentage of their team participating in reuse activities.
- Providing significant amounts of reusable software. Just as with incentives for individual suppliers, this encourages groups that have responsibility for building reusable software for use by other teams. The number of actual uses of their software (as reflected by SIRBO) indicates a strong customer focus and value to the organization.
- High levels of reuse. Achieving either: (1) a pre-specified goal for the level of reuse on their project (for example, achieve a Reuse%$^{©}$=20%), or, (2) an unforeseen, exceptionally high level of reuse.

## Conclusion

This article discusses how to encourage individuals and teams to participate in software reuse. A cornerstone of the incentives lies in having objective metrics upon which to assess and reward participation based on the value that it has brought to your organization. This allows your organization to consistently evaluate the value of reuse and share those benefits with he people that make it happen!

### References

[Poulin95] Poulin, Jeffrey S., "Populating Software Repositories: Incentives and Domain-Specific Software," *Journal of Systems and Software*, Vol. 30, No. 3, September 1995, pp. 187-199.

[Poulin97] Poulin, Jeffrey S. Measuring Software Reuse: Principles, Practices, and Economic Models. Addison-Wesley (ISBN 0-201-63413-9), Reading, MA, 1997.