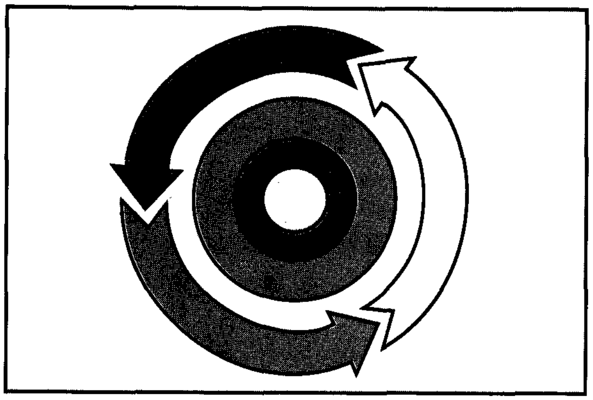
Technical forum



Management of reuse at IBM

For years, programmers have believed that productivity gains could be achieved if programs could be reused. The logic is that it is better to obtain code from libraries that contain reusable parts than to write new code. One can easily see the productivity gains from not writing code, as long as measurement of productivity is not made in the standard form of lines of code per unit of effort. Building systems from existing proven parts can also lead to increases in quality.

Significant accomplishments have been made within IBM since the early 1980s in reuse technology. Sites such as Böblingen (Germany), Houston (Texas), and Poughkeepsie and Owego (New York) have participated in this work. This entry within the forum briefly outlines the management structure created to support and guide the use of this technology.

The application of reuse technology can occur at different levels within an organization, from the project level through the business area level, as well as across the site. Each of these levels allows the focus on different domains. Reuse can be of benefit to all stages of the development life cycle, from requirements to post-ship.

The management structure. The current effort to formalize the management and use of reuse in IBM can be traced to the work of M. Lenz in Böblingen, in which the creation of *building blocks* lead to the realization that formal reuse was a worthwhile investment.[1]

Learning from the experiences of the building-blocks group, a reuse council (the IBM Corporate Reuse Council) was established to propagate software reuse through the software community.

The council began with the approach to establish broad communication mechanisms. These took many forms, including newsletters, a "starter kit," electronic bulletin boards, symposia, and papers. The council published its first newsletter in 1989, leading the way with articles on how reuse was being managed and supported in several organizations. The approach was viewed as consisting of four steps: (1) increase the level of awareness, (2) provide a supporting infrastructure, (3) provide education, and (4) establish management controls and technical participation.

With this approach defined, the council established goals to be accomplished over a five-year period. The goals would lead to:

- Integration of mature reuse technology and processes into the software development process (the effort included selected tools, processes, education, standards, measurements, and reuse parts)
- Achievement of significant increases in productivity (delivered product for the effort spent) and quality through the application of reuse

technology, resulting in a reduced product development cycle
- Establishment of IBM as a leader in reuse technology application

This approach called for the formation of a focal group charged with implementing the goals. This new group, the Reuse Technology Support Center was formed in January of 1991. Its primary responsibility is to coordinate the reuse effort within IBM, provide consulting to technical organizations and management, assist in issue resolution, and provide funds for tools and reusable parts creation, storage, and retrieval efforts, as well as other relevant technology innovations.

In addition to the Support Center, the concept of reusable parts technology centers was created, where the technical people take guidance from the Support Center and accept and act on requirements from the council and other users.

Visible management support for reuse was seen as critical to its success. This support must provide for additional resources if required, to contribute to the education of the programmer population, foster development of the necessary tools beyond those supplied by the Support Center, and fund development of reusable software for future use. Writing reusable software costs more than nonreusable software; however, software reuse is an investment for the future and management support is necessary to make that investment possible.

Business-area-level reuse. The implementation of reuse in a business area entails exploiting opportunities for software reuse across multiple contracts or products as opposed to addressing the needs of a specific contract or product. Reuse efforts at this level build on techniques applied at the project level; however, the scope extends across time (both past and future products) and into additional areas of the software life cycle.

The development and use of reusable software within a business area offers potential for cost and schedule savings. This has been demonstrated both within IBM and by others, including McDonnell Douglas Corporation's Common Ada Missile Program (CAMP) sponsored by the United States Air Force.[2] Projects like these demonstrate the range of component types beneficial to a business area—from small components (measured in hundreds of shipped lines of code) that implement specific algorithms, structures, or functions unique to a business area, to large-scale components (measured in thousands) that implement major portions of an application.

Project-level reuse. Implementing reuse on a project level is an effective means of increasing both productivity and quality. It can be accomplished without radical change to the standard software development processes but it does require some additional activities and a new emphasis on reuse in some traditional life-cycle activities.

The key reuse activities at project start-up time are the establishment of the project reuse team leader and the definition of the project reuse strategy.

The reuse team leader is the person who provides the continuity and focus for the reuse effort within a project, leading the reuse meetings and participating in all of the project design reviews and inspections. In some projects, the team leaders have identified members in each programming team to act as team reuse leaders. These team members are then responsible for the identification of reuse candidates, and may establish goals for the periodic identification of reusable parts.

The team leader must be aware of external sources for reusable components, including reuse candidates available in the various software reuse libraries from previous projects. These might be sources for reusable or salvageable software. Additionally, the team leader leads the effort to identify parts of the software product under development that may be made reusable and shared within the project and, optionally, incorporated in a site software reuse library. Finally, the team leader is the focal point for management, responsible for the evaluation of the reuse effort and for providing regular status reports.

Design phase reuse activities. There are two types of reuse activities that may occur during the program design phase. The first is reuse candidate identification, and the other is an activity to augment the existing software inspections.

The reuse candidate identification activity looks for available parts to satisfy project needs. In addition, this activity also looks for opportunities to turn current project deliverables into reusable parts. These goals are supplemented with additional steps in the existing software inspection process.

Post-development activities. These activities focus on lessons learned during the development effort using reusable software. Reusable components generated are moved to the correct libraries, and the lessons learned from the reuse effort are reviewed and analyzed for future applicability.

Site-level reuse. Implementing a reuse program for a site requires additional communication and coordination. This is provided by assigning a site "champion" with broad coordination and implementation responsibilities. Along with the champion, a common library of reusable parts is established and acceptance standards enforced. The primary focus of the IBM reuse program is to establish reuse across an entire site.

Libraries can be created at different levels in the organization, serving different purposes. By creating a hierarchy of libraries, organizations can focus on narrow domains, improving the opportunity for reuse. Parts that have a wider domain of applicability are moved to libraries higher up on the hierarchy. These libraries are shared across organizations by using tools that provide functions for search and retrieval, along with the necessary management and control functions.

Different sites within IBM have taken different approaches to populating their reuse libraries. Some examine their current development efforts and identify and build reuse candidates, while other sites solicit donations. Obtaining donations is a less rigorous approach than having ongoing efforts and existing projects submit reusable parts. This latter approach is a low-investment approach to populate libraries with large numbers of reusable parts in a short period of time.

The site may also establish a "candidates" library that can hold incomplete candidates. This allows development organizations to plan their work based on the future availability of reusable parts, and also prevents two organizations from potentially developing the same or similar software simultaneously due to a lack of visibility into each other's future plans.

Summary. To be successful, software reuse requires changes in technology, process, and the software development culture. Few major breakthroughs are necessary to exploit software reuse today. Certain attributes of software make it easier to reuse, and object-oriented methodologies certainly facilitate reuse. However, these are not necessary for reuse to take place. Our experience shows that reuse can be accomplished successfully in existing products using existing techniques and knowledge.

The progress being made in this effort within IBM is very promising. When the Reuse Technology Support Center was formed, it was originally targeting five sites for support during that first year. Currently there are close to 30 sites worldwide involved with the Support Center. As may be expected in such a large effort, the results are mixed. The best programs are showing savings in the millions of dollars and reuse has accounted for 20 to 30 percent of the software. Some organizations have applied reuse to allow them to include in their products a functionality for which they had no prior capacity. Some organizations are using high-quality reusable parts, such as the building blocks from IBM Böblingen, to actually improve their products. There have been cases where the finely-tuned correct data abstractions provided by the building blocks have exhibited better performance characteristics than custom-built data structures. These projects have benefited from the reduced maintenance costs as well as the improved performance gained.

Cited references

1. M. Lenz, H. A. Schmid, and P. F. Wolf, "Software Reuse Through Building Blocks," *IEEE Software* **4**, No. 4, 34–42 (July 1987).
2. C. Anderson, "The CAMP Project: A Pragmatic Approach to Software Reuse," *Military Computing Conference*, McDonnell Douglas, M/C 3064025, P.O. Box 516, St. Louis, MO 63166-0516 (May 1988).

General references

W. S. Humphrey, *Managing the Software Process*, Addison-Wesley Publishing Co., Reading, MA (1989).

R. Prieto-Diaz and P. Freeman, "Classifying Software for Reusability," *IEEE Software* **4**, No. 1, 6–16 (January 1987).

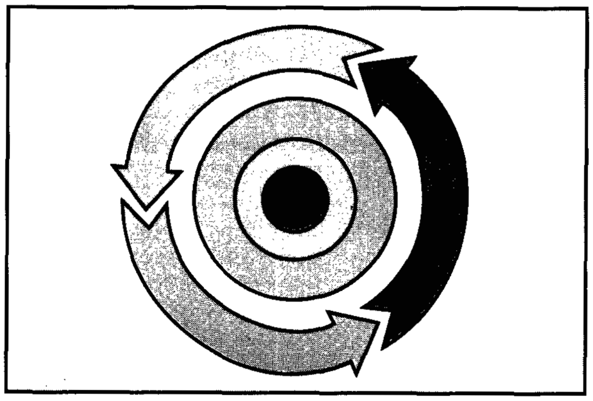
J. R. Tirso, "Championing the Cause: Making Reuse Stick," *Proceedings of the Fifth Annual Workshop on Software Reuse*, M. Griss and L. Latour, Editors, Department of Computer Science, University of Maine, Orono, ME 04469 (November 1992).

J. R. Tirso, "Establishing a Software Reuse Support Structure," *Proceedings of the IEEE International Conference on Communications*, Denver, CO (June 1991), pp. 47.2.1–47.2.5.

W. Tracz, "Software Reuse: Motivators and Inhibitors," *Proceedings of Computer Society International Conference,* (COMPCON), Spring '87; IEEE Cat. No. 87CH2409-1 (February 1987), pp. 358–363.

J. R. Tirso
IBM Personal Software Products Division
Boca Raton
Florida

H. Gregorius
IBM Large Scale Computing Division
Poughkeepsie
New York



Information reuse parallels software reuse

Organizations that place a high value on their information can best leverage and maintain that information if they apply the same infrastructure and techniques used for reusable software. The software life cycle produces several types of reusable information such as customer information, product development information, and process information. Just as software reuse benefits from structured programming practices, information reuse benefits from the use of common tools, centrally coordinated standards and terminology, and development practices consistent with good writing and design.

Reusable information. Information reuse is the reuse of nonexecutable entities. We distinguish information reuse from information management by the end use of the retrieved information. In information reuse, the user incorporates the retrieved information into a work product, whereas in information management, the user retrieves information to read or reference it. Computerized library card catalogs and information retrieval tools are examples of information management systems.[1]

The nonexecutable entities in information reuse can be grouped into four categories because of differing characteristics and requirements: customer information, process information, product development information, and miscellaneous information. Customer information describes software to the customer. Process information focuses on the process, such as ISO 9000 documentation, process diagrams, schedule documentation, and quality projections. Product development information includes business cases, requirements, designs, test cases, and plans. Miscellaneous information includes reusable forms and product-independent graphics.

Further complicating the issue of information reuse are the numerous media in which businesses can deliver the above categories of information. Four such forms are hardcopy, softcopy, integrated on-line, and hypermedia. Hardcopy consists of text and graphics printed on paper; softcopy is this same material when displayed on a video display terminal; on-line information consists of all text and graphics stored with the code for display and use in an integrated, interactive manner; and hypermedia is text, graphics, animation, audio, video, image or executable code stored in various places and logically linked together.

To determine whether a piece of information should be supplied as a reusable part, three questions may be asked:

- Is there a known near-term need for other uses of this information or document?