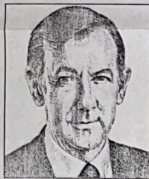


## APPLIED INTELLIGENCE

## IBM's Repository Is Key to a Successful CASE Strategy



This is the last of a series on IBM's recently announced Repository, a significant component of its applications development strategy for the 1990s.



The adoption of a standard repository for design information is vital to a successful strategy

for computer-aided software engineering (CASE) for both end users and vendors. In accepting a repository standard, users of compliant CASE tools will be able to share design specifications, while CASE vendors will be allowed to compete in an open environment. It's that simple.

The introduction of the Application Development (AD)/Cycle by IBM clarifies many of the issues in defining a strategy for CASE. Third-party vendors of CASE tools now can decide whether or not to adopt the de facto standard for design specifications defined by IBM. Users of CASE tools also can decide whether to limit their purchase of CASE tools to those compliant with the published standard. Still, there are other issues relevant to the implementation of a successful CASE strategy. These include support of top management, selection of highly integrated CASE tools, use of a methodology optimized for high-speed development, and issues related to the training, education and motivation of the development staff.

A chief objective of a CASE strategy is a substantial boost in the productivity of the applications-development process. As shown in the figure, a large improvement in productivity depends on four factors (depicted as four tall pyramids): tools, methodology, personnel and management. Correct judgments in all of these areas must be made to be successful in CASE. The management issues and the people issues are especially critical. There are many examples of CASE strategies that have failed due to a lack of attention to these issues in making the transition to a new development environment.

The announcement of the IBM Repository provides information systems (IS) executives with an opportunity to review their overall CASE strategy. It is important for them to gain a broad perspective on the entire subject of CASE, because three years from now, the IBM Repository and the AD/Cycle applications development environment will be widely used. Planning now for the introduction of these new architectures and development techniques will reduce costs and increase the effectiveness of implementing them.

Organizations planning for CASE should use diagramming techniques, such as entity-relationship diagrams, as well as data-modeling and process-modeling techniques. Above all, they should plan to use the most powerful development methodologies available, such as information engineering, to build the

strategic applications of the 1990s. Information engineering is a methodology that was specifically designed to support the development of the complex, mission-critical applications required for the computerized company of the future.

Managers need to learn how to implement new business thrusts faster than their competition through the use of high-end, integrated CASE tools and methodologies that are optimized for high-speed development. A large part of the success of high-speed development depends on building an infrastructure. It is possible to develop applications on a project-by-project basis with no infra-

structure a strategy for CASE. The manager must understand the fast application life cycle and reusability, and be able to select CASE tools that meet the evolving needs of the organization.

It is important to select tools that are highly integrated and that support the IBM Repository standard. Unfortunately, the IBM Repository is not available; even in June 1990, it will still be a fragment of what is available now in the repositories of highly integrated CASE tools from third-party vendors. The repositories in these tools are being rapidly brought into compliance with IBM's Repository.

Some of them are also supported by

is much easier to demonstrate results quickly with single project development than it is with information engineering, because information engineering requires approximately two to three years to build the infrastructure. However, once it is in place, it is possible to build applications fast consistently.

To effectively monitor productivity, it must be consistently measured. Top management should be informed that a metrics program is being put in place and that measurements of productivity will be reported to them on a regular basis. This may alarm some personnel because it gives the CEO a means of measuring the success of the IS department. However, consistent measurement and reporting of productivity gains are a powerful way of attracting and retaining the interest of top management in a CASE strategy.

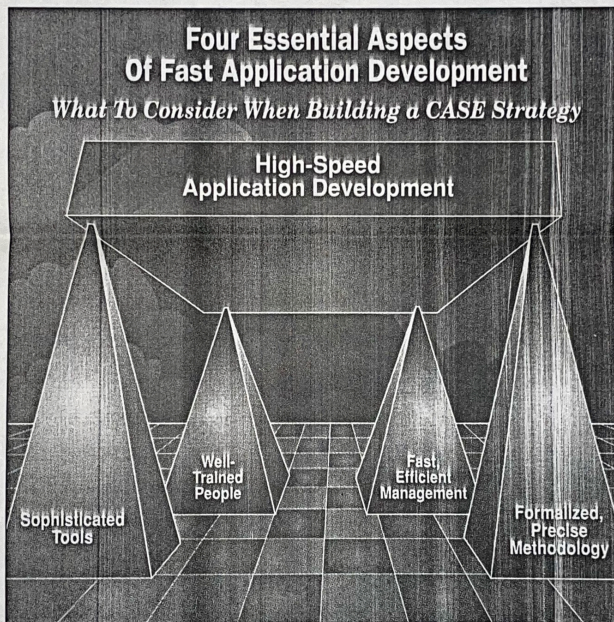
Some CASE vendors are beginning to provide metrics automatically from information stored in the repository, but much more needs to be done. As the analyst creates design diagrams, the system can track the number of inputs, outputs, inquiries, files, interfaces and algorithms. These measurements can be expressed in the form of function points, which are a measure of the complexity of the module. As the design progresses, the function-point estimate is refined. Before entering the construction phase, a detailed function-point computation is made, which can be combined with other project-management information to provide productivity measurements. All CASE vendors should provide these metrics automatically.

### Choosing the Right Vendors

Another important step that organizations can take now to prepare for the introduction of advanced applications-development environments such as AD/Cycle is to select CASE vendors who have committed to the common Repository standard. As of last week, 32 CASE vendors have pledged support for IBM's AD/Cycle Repository.

In summary, the successful implementation of a CASE strategy relies on many factors. Primary among these is attracting the interest and support of top management. Additional factors leading to success include selection of a highly integrated CASE tool and a methodology that is optimized for high-speed development. Issues involving training, motivating and educating the development staff may be more important to the successful introduction of CASE than technology issues. The management of cultural change within an organization will be a major challenge for the computerized company of the 1990s. ■

To learn more about the subject of these articles, please call The James Martin Report, an information service updated quarterly, at (800) 242-1240. For information on seminars, please contact (in the United States and Canada) Technology Transfer Institute, 741 10th St., Santa Monica, Calif. 90402 (213) 394-8305. In Europe, contact Savant, 2 New St., Carriforth, Lancs, LA5 9BX United Kingdom (0524) 734 505.



John Avakian

**With a repository standard, users of compliant CASE tools will be able to share design specifications, while CASE vendors will be able to compete in an open environment.**

structure at all, but development will be much faster if resources are devoted to building an infrastructure that promotes a high level of portability, connectivity and reusability.

To achieve fast development, the methodology must resemble a finely tuned clock, in which all the gears mesh perfectly. If any of the gears do not mesh, development will not proceed at a rapid pace.

An important aspect of a strategy for CASE is selection of the right manager. The manager can be described as the vice president of CASE—someone who clearly understands the critical success factors of the organization and can ar-

very detailed methodologies, such as the Information Engineering Methodology from James Martin Associates.

Productivity improvements of two to three times over conventional techniques are quite achievable with high-end, integrated CASE tools, including the effort devoted to extensive front-end design activities. Such improvements do not require sophisticated development methodologies.

There are many examples of corporations that have improved productivity by a factor of 10, and many of these do not use methodologies such as information engineering; rather, they were the result of single project development. It