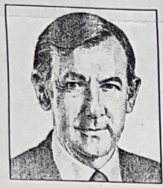
APPLIED INTELLIGENCE

RAD Designed To Facilitate Dynamic Change in Firms



JAMES MARTIN

This is the first of a series of articles on rapid applications development (RAD), a methodology for development designed to be much faster than the traditional life cycle.

One of the urgent concerns in enterprises today is the need for information-systems (IS) organizations to create and modify applications faster than is possible with the traditional development life cycle.

If applications take two or three years to build and the application backlog spans several years, businesses cannot create and react to competitive thrusts quickly enough. The vital ability for dynamic change is lost.

The Need for Speed

The pace of business is increasing. To stay competitive, corporations are shortening their manufacturing life cycles. They are under pressure to design new goods quickly and shorten the time required to fill orders. Factories are striving to reduce their inventories with "just in time" inventory control. In many situations, the time available to make decisions is decreasing.

To a large extent, this increase in the business metabolism is caused by the use of networks, immediate access to data and powerful decision-support tools on the desktop.

Corporations are looking increasingly for ways to beat their competitors to market with new goods or services.

American Airlines, for example, increased its market share by building Sabre, a complex system to put on-line terminals in travel agents' offices. TWA recently rebuilt its frequent-flyer program using a high-end computer-aided software engineering (CASE) tool and can now respond much faster to moves by competitors.

United Stationers grew much faster than its competitors by guaranteeing 24-hour delivery of orders placed with terminals. Canon devised computerized techniques for designing innovative lenses and cameras faster than competing companies.

In the 1990s, business will become a battle of high technology, often of competing computer systems. In such a world, the corporation that can create the necessary computer applications fastest can win. If a company takes three years to build and debug a complex application while its competitors does it in six months, the company will lose the business.

In a society with a worldwide mesh of intercorporate computer networks, electronic decision making, electronic funds transfer, computer-to-computer order placement and robot factories capable of building customized products overnight, it seems outdated to spend years handcrafting applications.

In the attempt to create applications faster, critical success factors for every IS department include:

- the speed of applications development;
- the speed of applications maintenance; and
- the speed of cutover of new applications.

There are many diverse aspects to achieving fast applications development. This series of articles discusses these new approaches and shows how they can be used together in new applications-development life cycles. If these techniques are widely adopted, IS can move into a new era of fast development that facilitates dynamic changes in the way corporations operate.

Rapid applications development refers to a development life cycle designed to give much faster development and higher-quality results than the traditional life cycle. It is designed to exploit powerful development software that has recently evolved.

The RAD life cycle is based on significant improvements in four areas:

- Tools—code generators, integrated CASE tools, prototyping tools and fourth-generation languages;
- Methodology—life cycles optimized for high-speed development;
- Personnel—selection of highly trained and motivated personnel; and
- Management—use of innovative management techniques to break down bureaucratic and political obstacles to high-speed development.

RAD encompasses a set of techniques that can be used to build complex, strategic and mission-critical applications in months rather than years. The RAD life cycle is highly flexible and can be adapted to the nature of the system and the tools being used.

Today there is an astonishing difference between the best applications development and that in the majority of IS departments.

As shown in the figure, the productivity of small teams using RAD techniques is more than 10 times greater than the productivity of those using well-managed third-generation language life cycles with productivity aids.

The productivity figures are measured in the form of a complexity metric called function points. Complexity is measured by counting the number of inputs, outputs, storages of system data, inquiries, files and external interfaces in a program module.

The method adds these factors and applies a weight to each element to produce a measure of the complexity of each module, independent of computer language or environment. Using this method, a function point corresponds roughly to 114 lines of COBOL code.

The productivity figures in the diagram summarize measurements of the number of function points that were produced per person-month in a range of actual applications using widely varying development techniques. The cost figures reflect the cost in dollars for each completed function point.

The high productivity figures summarized in the figure are based on real organizations that build systems quickly and can do so repeatedly. As in scientific experiments, the success must be repeatable in order to be interesting.

The examples cited use tools and techniques that are transferable to all IS organizations, given appropriate management.

Lower Cost

Techniques for fast applications development almost always result in lower-cost development.

A key to fast development is using small teams rather than large ones. Small teams can be made highly productive if they use methodologies for rapid analysis and power tools that generate code. A reduction in development time is thus combined with a reduction in the number of people needed.

Integrated CASE tools that can generate code for complete applications from design specifications are appropriate power tools for applications development in a RAD environment.

The CASE tools on the market vary greatly in capability: With some, little improvement in productivity occurs. It is critical to select integrated tools that generate program code for complete applications. The best tools are quite expensive, but not compared with the cost of large teams of IS professionals using the old techniques.

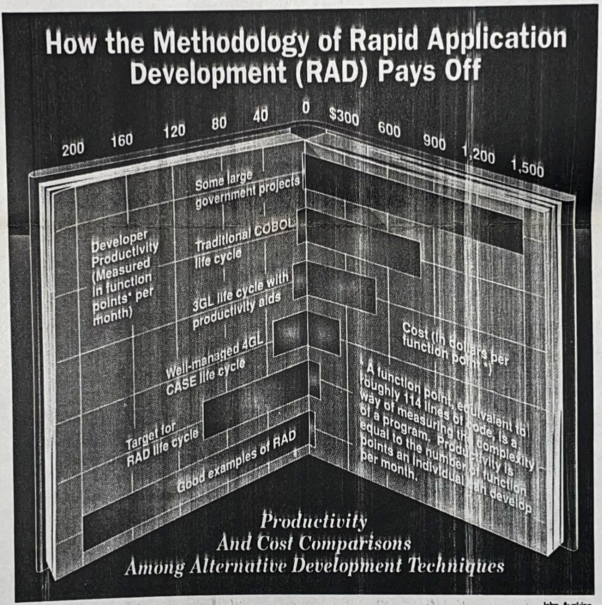
Fast development does not mean "quick and dirty." It is necessary to build applications of high quality—indeed, much higher quality is needed than in many of the applications built with traditional methodologies.

Some IS organizations using CASE tools have not improved productivity, lowered costs or shortened development times. Only a small proportion of CASE users have learned how to achieve dramatic improvements in these areas. Some IS executives achieve these improvements on virtually every project, and it is these users who are gaining a major strategic advantage for their organizations.

Next week, I'll discuss the actions that must be taken to retool the IS organization in order to gain a competitive advantage. ■

The concepts embodied in RAD were introduced by James Martin at a live satellite conference on Dec. 12. For more information, call (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., Carnforth, Lancs., LA5 9BX United Kingdom (0524) 734 505.

How the Methodology of Rapid Application Development (RAD) Pays Off



Productivity And Cost Comparisons Among Alternative Development Techniques

John Avakian

To a large extent, this increase in the business metabolism is caused by the use of networks, immediate access to data and powerful decision-support tools on the desktop.