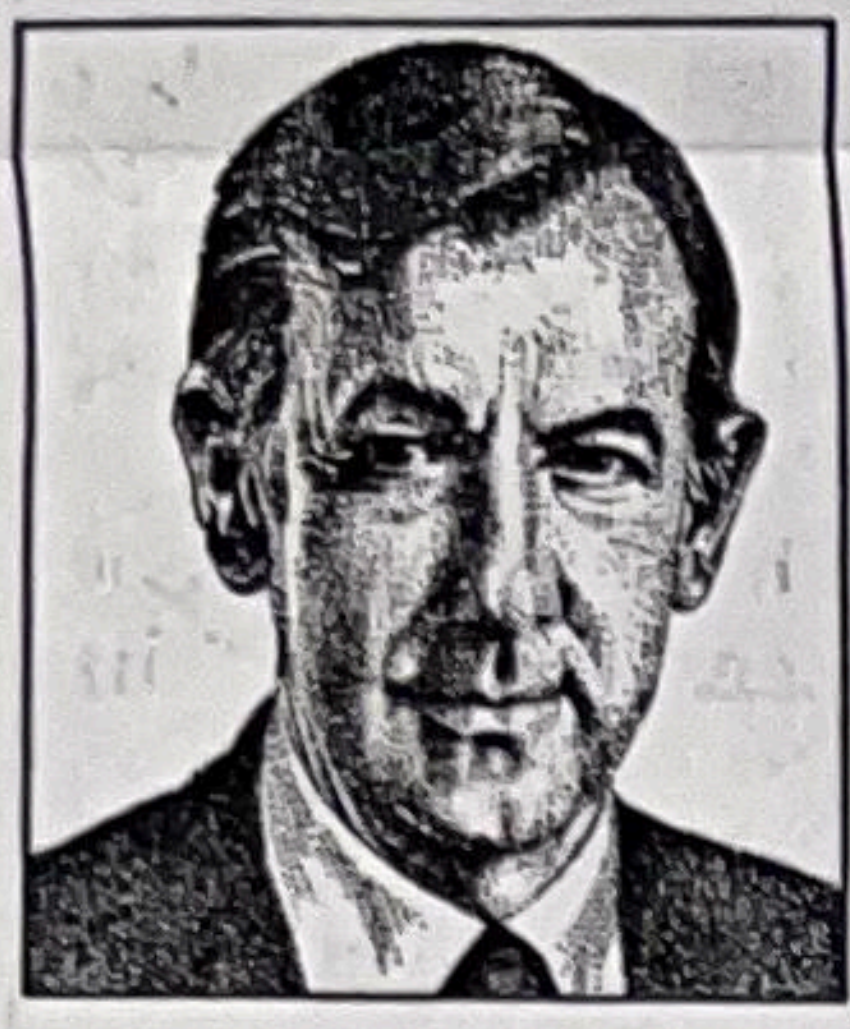
APPLIED INTELLIGENCE

Tight Synchronization Key to High-Speed Development



JAMES MARTIN

Well-managed tools, life-cycle processes, personnel and management all contribute to the success of new development methodologies such as rapid-applications development (RAD).

The RAD methodology is based on the techniques used by enterprises that have already achieved fast applications development. RAD results are often impressive, with some achievements of more than 200 function points per person-month. The most successful techniques used by these organizations were synthesized to form the basis for the RAD methodology, described in my previous columns.

Achieving high-speed development is a complex process. Although many companies were very successful in achieving fast development, no enterprise was doing everything right. Many lacked critical elements of the RAD life-cycle process. Some attempts at fast development weren't productive because inadequate computer-aided software engineering (CASE) tools were used. And some misfired due to lack of user commitment.

Still other projects didn't employ end users in the joint requirements planning (JRP) and joint applications design (JAD) workshops. Some projects lacked a high-level executive owner to "grease the wheels," while others didn't involve users in the evolution of the prototypes. Some projects slipped because of inadequate cutover planning; others didn't emphasize quality.

One of the most powerful approaches to fast development is to incorporate techniques for reusable design. The majority of the organizations studied, however, weren't doing this. And some that were lacked an integrated code generator; others didn't have a CASE repository. Reusability has a long-term payoff, giving steadily better results as more reusable items are built.

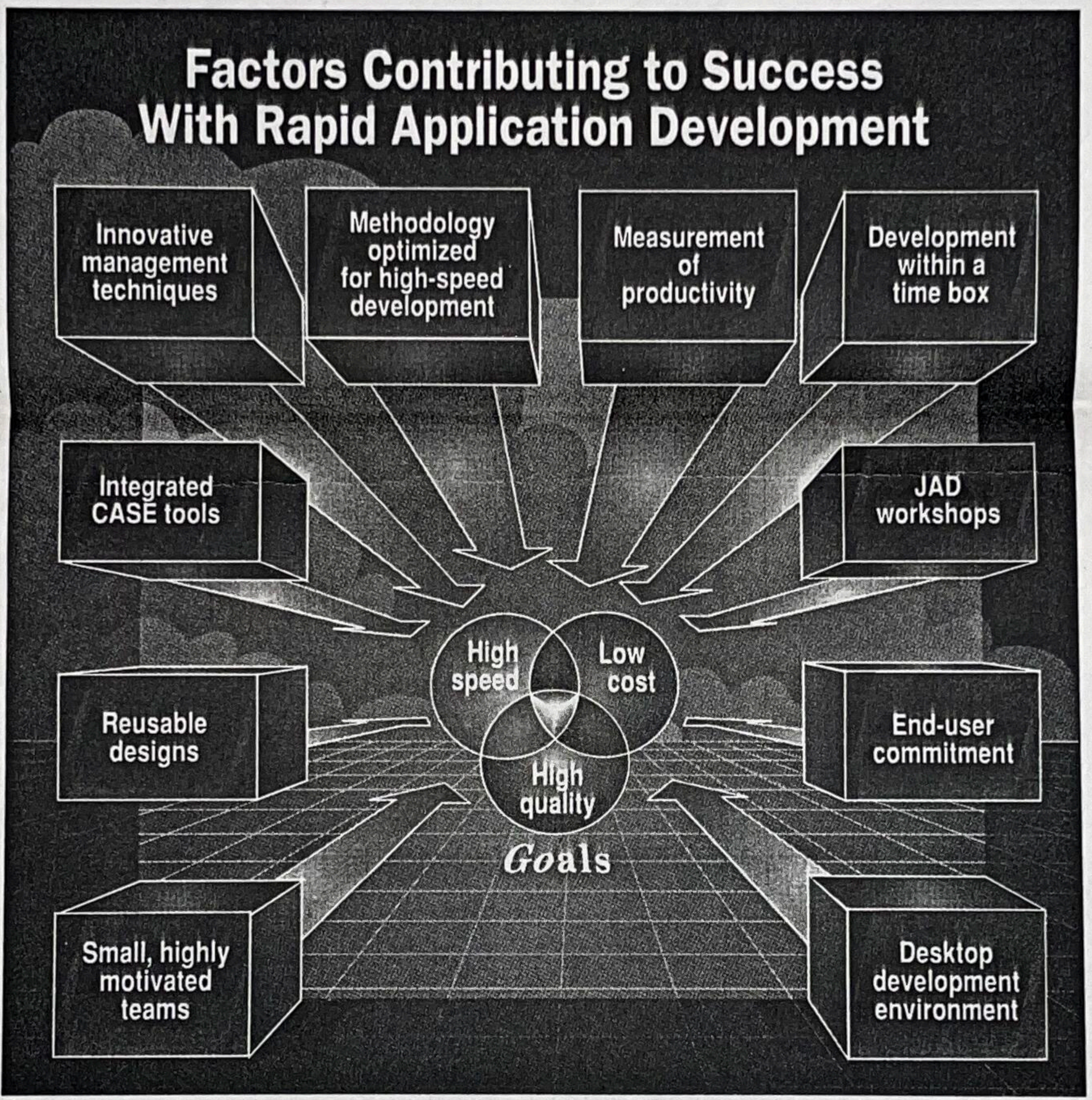
Information-systems (IS) managers can maximize productivity by identifying and combining all of the development techniques that have demonstrated high productivity. The RAD methodology attempts to do this. The figure summarizes the techniques that have proven to be most effective.

Integrated CASE (I-CASE) tool sets are used in the early phases of the RAD life-cycle process to capture user specifications, analyze them for logical consistency and generate prototype applications.

During the construction phase, a highly trained team of IS analysts uses I-CASE tools to expand the prototype system into a production application. To support these functions, the team requires a graphically oriented I-CASE tool set that integrates data-modeling, process-modeling, system-design, prototyping, screen-painting, dialogue generation, report generation, database-generation, efficient code-generation and testing tools.

Specifications for an application are stored in an I-CASE repository that's available on-line to all developers. A design analyzer ensures that specifications stored in the I-CASE repository are analyzed automatically for logical completeness, consistency and correctness.

CASE developers prosper most when they use I-CASE tool sets that generate all the code for an application from specifications stored in the I-CASE repository. RAD encourages the use of a desktop development environment, which gives the fastest possible cycle of design, generation and testing. Several commercially available I-CASE tool sets can generate code for an application on the desktop, eliminating the expense of a mainframe code generator. A code optimizer may also be used where appropriate.



Factors Contributing to Success With Rapid Application Development

Innovative management techniques
Methodology optimized for high-speed development
Measurement of productivity
Development within a time box
Integrated CASE tools
JAD workshops
High speed
Low cost
High quality
Goals
Reusable designs
End-user commitment
Small, highly motivated teams
Desktop development environment

John Avakian

In order for complex projects to succeed, development activities need to be synchronized. One of the best approaches to fast development is to reuse effective techniques.

It's important to have a high-level executive owner responsible for the system who is financially committed to it and is determined to move fast. The executive owner must be briefed on the events and timetable of the RAD life cycle. He or she must ensure that the users want the system and are committed to helping establish its functions and design. All users must know their role in the development life cycle.

The most knowledgeable users must participate in the requirements-planning and user-design workshops, which are conducted during the early phases of the RAD life cycle; appropriate end users should be selected to review the prototypes as the system evolves; and users and management must be highly motivated and enthusiastic about the development process. Social disruption and the effects of cultural change can be minimized by ensuring that planning and education for job changes are done early in the life cycle.

The IS project manager must understand the RAD life cycle and, if possible, have prior experience with it. The project manager must also guide others in the correct use of the methodology.

The JAD leader organizes and conducts the user JRP and JAD workshops. This person, by doing these tasks continuously, will build up a high level of expertise,which is critical to consistent success with RAD.

The construction, or SWAT (specialists with advanced tools), teams should be small, highly trained and skilled, with a powerful I-CASE tool set. They should be able to build systems rapidly from the design output of the JAD workshops. A person skilled in data modeling can build or validate both the data model used and the coordinating model for parallel development.

Construction teams should be motivated by a sense of pride to achieve the RAD objectives.

The RAD methodology, ideally represented in hyperdocument form on each developer's PC, should give guidelines—and warnings of pitfalls—for each stage. The life cycle and methodology should be customized to adapt it to specific development practices and to make the best use of the tool sets selected.

The life cycle should have the following phases: requirements planning, user design (these may be combined), construction and cutover.

The requirements-planning phase should incorporate a user JRP workshop. The user design phase should include one or more user JAD workshops, using I-CASE tool sets and prototyping techniques. The design specifications developed in this phase should be stored in the repository of an I-CASE tool set so that these specifications can be used directly in the construction phase, where the design is defined in detail and used to drive a code generator. The planning and preparation for the cutover phase should proceed in tandem with the other development phases.

Obtaining High Quality

High quality can be obtained if all design specifications and procedural code for the system are defined with the I-CASE tool and stored in the I-CASE repository, so the tool set can check their integrity and consistency. Hand-drawn diagrams and English text of traditional specifications should be avoided because a computer cannot check them and they are usually full of errors.

Specifications and design should be created in user workshops. This is faster than the traditional process of specification writing by analysts, and it meets users' needs much better. Wherever possible, documentation should be printed from the I-CASE tool set. Some tools generate development workbooks.

Standards and guidelines for usability are also necessary. A person with skill in this area should check the evolving system for these qualities.

The use of these techniques, within the framework of the RAD life cycle, will result in consistent improvements in software-development productivity.

Next week, I'll begin a series of articles on reverse engineering, a new set of techniques that attempts to convert data definitions and procedural code into higher-level design specifications. ■

The concepts embodied in RAD are described in a new volume in the James Martin Report Series. For more information on this volume, call (800) 242-1240. For information on seminars, 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.