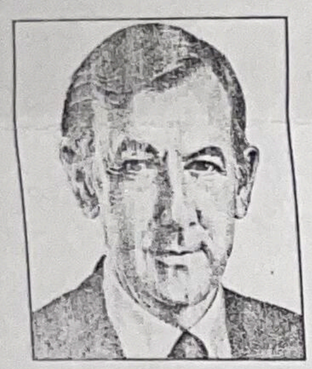
APPLIED INTELLIGENCE

User/Analyst Teams Make the Best Systems Developers



JAMES MARTIN

This is the sixth in a series on rapid applications development (RAD), a development methodology designed to yield much faster results than traditional methods.

RAD is more successful than the traditional life-cycle process because it delivers high-quality solutions faster. Just as important, the RAD life cycle is designed to meet users' needs as effectively as possible at the time the system is implemented.

The RAD life cycle is fundamentally different from the traditional development life cycle. It combines many of the techniques found to be most successful in building systems fast.

These include rapid prototyping, reusable designs, end-user workshops for planning and design, integrated computer-aided software engineering (I-CASE) tools, code generators and industry-standard repositories.

Other techniques found to be successful are automatic rule-based validation and coordination, data modeling, process modeling, joint applications-design techniques, highly motivated small teams and innovative management processes.

These techniques are individually valuable if used correctly. In synergistic combinations, they become more powerful, resulting in a revolutionary change in the development of systems.

Modern development methodologies such as RAD combine all of the above techniques into an effective life-cycle process.

Tool sets have evolved to support these combinations with a maximum amount of automation. The RAD methodology guides the practitioners in the best use of the combined techniques.

The traditional life-cycle process has serious deficiencies. Users cannot obtain strategically important applications when they need them. There's often a delay of years. Systems that are delivered often fail to meet user requirements. Systems cost much more to develop and maintain than anticipated.

Users and information-systems (IS) professionals do not work well together in the traditional process. Users have difficulty communicating precise specifications to IS. Specifications documented by IS, on which the users have to sign off, are hard to check and are usually full of inconsistencies, omissions and errors. In addition, requirements often change after the specifications are frozen.

Because of the long delay in obtaining results, systems seen as important to the business are not implemented. IS development is seen as a bottleneck that prevents management from evolving the business in ways that would increase its competitiveness.

The traditional development life cycle uses inefficient, outdated techniques. The tasks are often boring, work-intensive and time-consuming.

The creators of this methodology assume that the developers do not understand how to develop an application and must be told, step by step, what to do.

In reality, the best development is often accomplished by skilled teams who know exactly what to do. They resent being forced to follow a slow and rigid methodology when they can build a high-quality system faster.

As shown in the figure, the traditional life cycle for the development of an application with a complexity of about 1,000 function points requires a high degree of IS involvement over an extended period of time.

End users are involved only in the initial specification phase of the project and in the final integration and cutover phases. During the lengthy process of design, coding, testing and integration, the business environment may change significantly, invalidating many of the original functional requirements.

The deficiencies of the traditional life-cycle process can be resolved by the RAD methodology. The RAD life cycle for an application with a complexity of about 1,000 function points is relatively short, as shown in the figure.

The development period is short enough that the business environment is unlikely to change substantially during the RAD life cycle. End users are actively involved in all phases of the development effort. Requirements planning and system design are performed by small teams of end users and IS analysts using joint applications-development techniques under the guidance of a workshop leader.

During the construction phase, a highly trained team of IS analysts evolves a prototype of the application using an I-CASE tool. As the prototype evolves, it is evaluated by a small team of end users.

The construction phase is performed within a tightly constrained window of time, called a time box. At the end of the time box, the prototype evolves to the point that it supports all of the functions required by the end users.

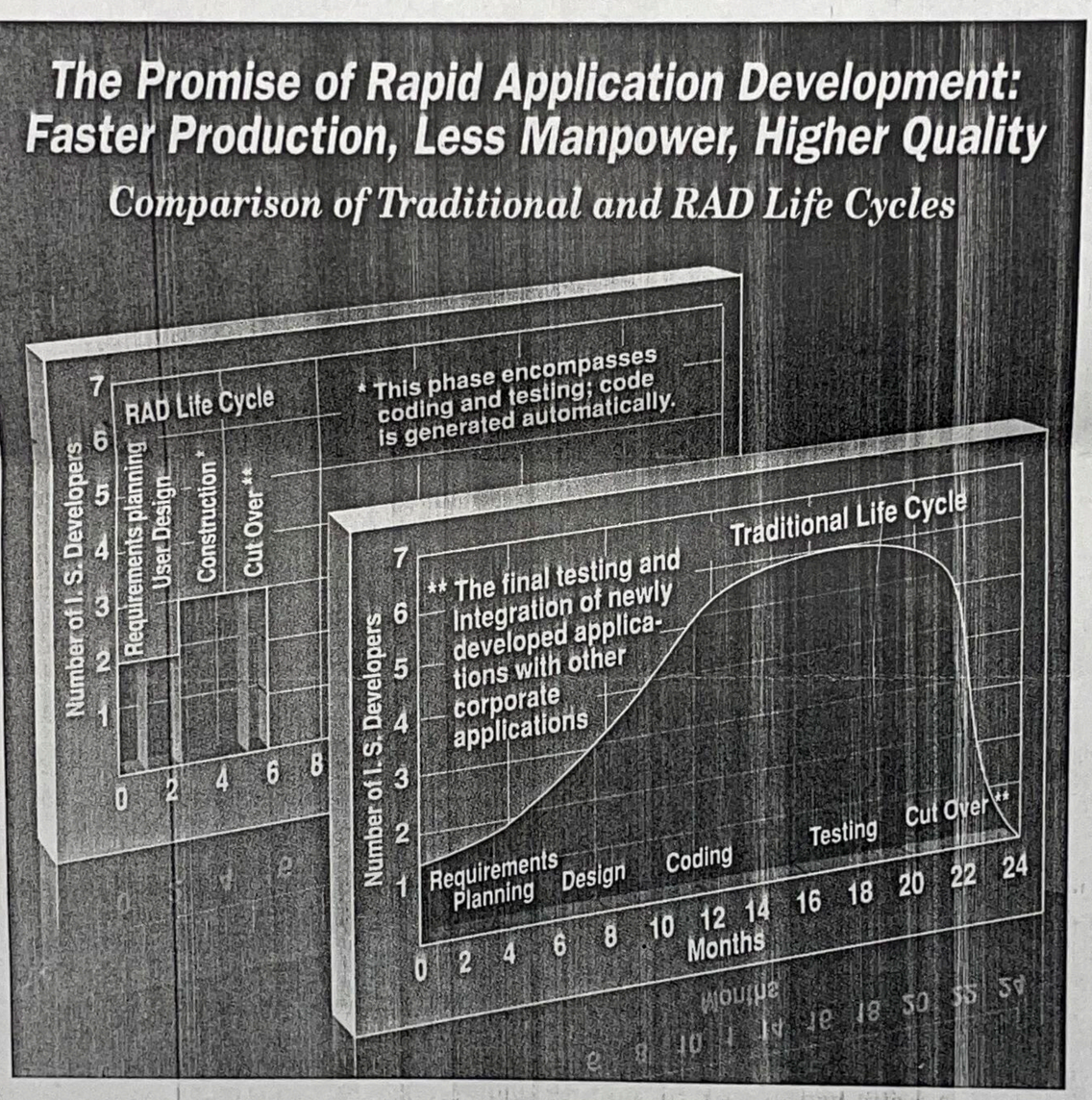
Cutover includes development of the final production product, final testing, acceptance testing, final training, installation of the production-system environment, data conversion, moving the system into production and reviewing system operation. End users are heavily involved in all aspects of the effort.

The RAD life cycle has major advantages over the traditional life cycle. These include the following:

- **Speed:** RAD produces results very quickly, with high-quality design and implementation.
- **Quality:** This means meeting the true user needs as effectively as possible at the time the system is cut over.
- **Avoidance of programming by hand:** Code is generated automatically (ideally on the desktop) for an entire application by the integrated generator of the CASE tool.
- **Better communication with end users:** The knowledge of users is harnessed and requirements are communicated through joint applications-development workshops. A CASE tool is used within the workshops to capture design specifications and build the prototype.
- **Enterprisewide planning:** RAD incorporates techniques to define an enterprisewide strategy, planning, data modeling and process modeling. Formal interaction with high-level users encourages identification of how new systems can improve the business.
- **Verification of correctness:** All syntax and internal errors in semantics are caught automatically by the CASE tool analyzer.
- **Consistency:** All design specifications are stored in the repository of a CASE tool, which enforces the integrity and consistency of the specifications.
- **Reusability:** Object-oriented planning and design are incorporated to achieve the maximum degree of reusable data structures, procedures, components, templates and designs that are used to generate code.

Unlike the classic life cycle, the RAD techniques get the intended system users involved in every stage of the life cycle. The RAD life cycle is designed to thoroughly incorporate interaction with end users. Involving the users throughout the development of the system ensures that their needs will be met when the system becomes operational. This is the true measure of quality and success of the system.

Next week's column discusses the phases of the RAD life-cycle process. ■



John Avakian

Methodologies were created because of the difficulties inherent in traditional development methods. But in much of IS development, the methodology is the problem.

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.