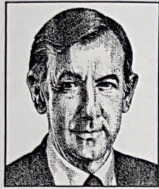


## APPLIED INTELLIGENCE

# Within the Timebox, Development Deadlines Really Work



**JAMES  
MARTIN**

*New development methodologies such as rapid application development (RAD) use Du Pont's Timebox concept, in which the development deadline is immovable, but the functionality may slip. People in many walks of life*

deadlines. Writers, television producers and seminar developers, for example, must deliver material by a certain date, which means the content of the material may slip.

Methodologies such as RAD apply a similar constraint to the building of IS applications: There is a deadline that is immovable, but the functionality of the system may slip. The system must work, carrying out its basic functions, but the refinements may have to be postponed in order to meet the deadline.

Seventy-five percent of a system's functionality can be created relatively quickly with an automated code generator, especially if reusable structures are employed.

The next 15 percent may take as long to create as the first 75 percent, and the system may have to be successively refined before the last 10 percent is completed.

Many of the features in the last 10 percent could be dropped or postponed for a subsequent release.

If the refinements are added after the first version of the product has been in use, they will probably be subject to change.

## Adding Refinements

It is often better to add refinements after the first version has been in use because users change their minds about the required functionality of a system after working with it for a while.

One of the dangers of prototyping methodologies or iterative development is that the functions of a system can grow in an uncontrolled fashion. Users or developers often add functionality so that the design does not converge quickly into a usable system. This is referred to as "creeping functionality."

The more powerful the prototyping tools, the more the developers are encouraged to experiment, to add functions or to overengineer the system. This can become expensive and can prevent a system from being delivered on time.

Perhaps the best way to combat creeping functionality is to place a rigid limit on the time permitted to produce a working system. Within a defined timebox, a working system must be built.

With RAD, the core of a system is built quickly and the refinements are added later. The system grows like an onion, with successive layers of refinement being implemented.

It is this refinement that makes set-

ting a deadline practical. The Timebox life cycle allows refinements to be made until the deadline nears, at which time a working system must be delivered.

Du Pont's Timebox was first used in the company's fibers division, which was moving to a highly automated manufacturing environment. It was necessary to create complex application software quickly, and Du Pont recognized that it is better to get a basic version of the system working, learn from the experience of operating with it and then design an enhanced version than it is to wait for a comprehensive system at a later date.

Du Pont's experience implies that:

struction (or SWAT) team is given a time frame within which a system must be constructed.

Prior to the assigned time frame, the functions and design framework of the system are defined. After the time frame, the system is evaluated and a decision is made as to whether to put it into production.

The time frame cannot be extended; however, the functionality of the system may be trimmed to complete the system within the time allotted.

Within the time frame, continuous iterative development is done with end users and IS developers working closely together. The team is under pressure to

timation as to how long the project will take. This estimate should be made at or before the end of the final joint applications design (JAD) workshop, within the RAD user design phase.

By the end of the final JAD workshop, the following should be estimated: the number of function points, inputs, outputs, logical internal files, external files and queries, and their complexity (high, medium or low).

An experienced SWAT team knows what it can achieve. It examines the JAD and uses its own guidelines for determining how long the project will take and whether additional assistance is needed.

It may allocate a certain period of time for startup, a number of person-days for each transaction type, a period for integration testing and a period for handing over the project.

SWAT teams should work on systems or subsystems that can be completed in a short time. If the overall project is large and complex, it should be divided into subsystems, which can be built at the same time by separate teams.

The SWAT team effectively signs a contract with management saying, "We will build a system of that scope by a certain date."

## Changing the Scope

If the scope changes during the construction phase, more development time may be needed, or required functionality must be dropped. Any change in scope should be agreed upon with management during the construction phase.

Limiting functionality to meet the deadline does not mean quality must be compromised. It is essential that SWAT teams have pride in their work; no self-respecting team would put out work that was less than the highest quality.

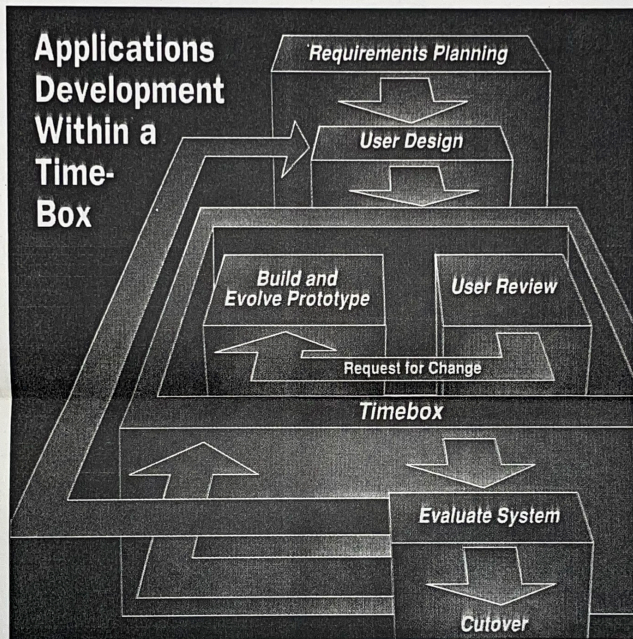
The delivered system must be as bug-free as possible and must provide a set of functions that meets the user's needs when the system is put into production. Integrated CASE tools make it much easier to achieve technical quality.

Functionality can vary substantially without reducing the usability of the system.

Many software and electronic products today have excessive functionality that often bewilders users. The return on development investment goes down when unwanted or marginal functions are added.

The Timebox development concept places an inflexible limit on the functionality to be supported and the time taken to produce a working system.

Next week I will discuss how innovative management techniques can lead to higher productivity. ■



John Avakian

**Within the time frame, continuous iterative development is done with users and IS developers working closely together. The team must produce a working system by deadline.**

- The first version must be built quickly.
  - The application must be built so that it can be changed and added to quickly.
- Du Pont stresses that the Timebox methodology works well for the company and is highly practical. It has resulted in automation being introduced more rapidly and effectively.

Du Pont quotes large costs savings from the methodology, and variations of Timebox techniques have since been used in many other corporations.

RAD methodology applies the Timebox concept to its construction phase, as shown in the figure. A con-

produce a working system by deadline.

A Timebox approach, or team development in general, does not work well if there is pressure to meet an impossible deadline.

The setting of the deadline, or the selection of functions to be accomplished within an allotted time, should be the responsibility of the team or systems analyst working on the project. Meeting the deadline is easier if the team is familiar with the tools and techniques, and confident in what they can accomplish.

Before deciding on a realistic deadline, the SWAT team should make its own es-

*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., Carriforth, Lancs., LA5 9BX United Kingdom (0524) 734 505.*