MACHINE DESIGN

PROVEN AMERICA'S MOST USEFUL DESIGN ENGINEERING MAGAZINE

A PENTON PUBLICATION SEPTEMBER 28, 1995

METRICS MAKE YOU FAST

One area of concurrent engineering still in its infancy is that of measuring progress during development efforts. The only widely used metric today is performance to schedule, which gives no insight as to the source of problem areas.

There is a move afoot to more precisely determine, early on, how projects are doing. On this score, a few key metrics for measuring product development have been described by Brad Goldense, president of Goldense Group Inc., a consulting firm specializing in business and technology management practices. World-class companies, he points out, have begun gauging all the steps between the time a product concept is first identified and when the finished article finally reaches the market. Beginning the measurement process at the inspiration stage tends not only to raise warning flags early, but also to reduce the time needed for ideas to bubble up and hit production, says Goldense.

Techniques that can expose difficulties include measuring target project size, constructing a concurrency matrix, and boosting project staffing speed.

Target project size — Most organizations are good at handling projects defined by a specific duration and employing a finite amount of company resources. Projects that are too small may not be worth the effort. Projects that are too big risk delays and gross misjudgment of required assets. So it makes sense to quantify the scale of projects your firm can reasonably undertake. Though many companies have an inherent feel for such limitations, real insights can come from a more structured analysis.

Take a set of representative projects (Goldense suggests 10 or 15) and for each one plot "total project person-years" against time to market. Now categorize each project first as either successful or unsuccessful, and second in terms of whether it was an incremental or a next-generation development effort.

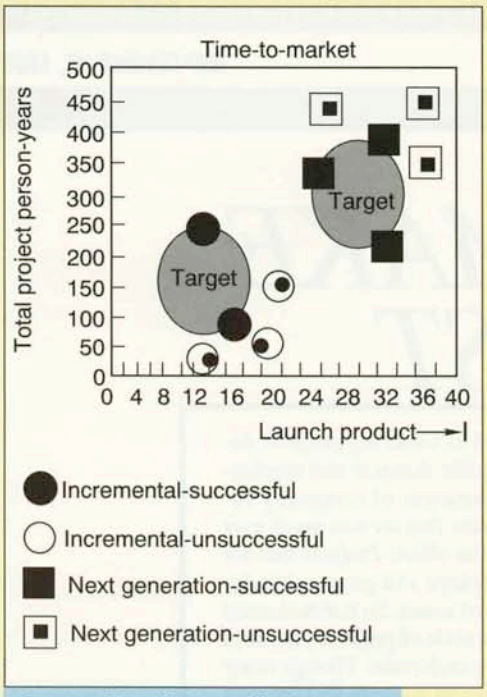
Most companies will have successful projects congregating in what could be called sweet spots on the graph. Problem children are more likely to be splattered all over the map. The best way to make use of this information is to pick out the common attributes of the successes, and of the failures, then make some assessments.

It is neither possible nor prudent to have all your projects in the sweet

continued on the back

spots, says Goldense, but be forewarned if a new effort looks as though it will fall outside their boundaries.

Concurrency matrix — Many organizations construct a 2D matrix to describe when and how different disciplines should begin work during development. But the approach can be taken a step further to highlight where skills are being applied too late. This is important because though numerous manufacturers now preach early involvement by cross-functional teams, the reality often remains too-little/too-late resource deployment.



Successful projects are likely to cluster together in sweet spots when plotted on a time-to-market versus person-years diagram.

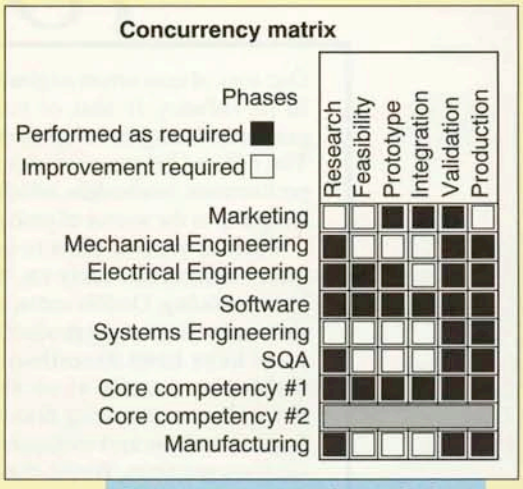
A concurrency matrix can show graphically where organizational commitment is lacking. It is of most benefit when first embarking on CE. Ideally, you would construct before and after concurrency matrices, one using pre-CE development procedures, the second later on to verify changes brought about by concurrent methods.

Matrix definition requires that program managers agree on common terms for row and column headers. First to be decided is the list of functions that are central to the development of new products. Each function becomes a row. Typical functions would be software development, structural analysis, and so forth.

Columns become the major phases in the product development process. There are typically between four and seven phases. What's tricky about this is that different phases may mean different things to different managers.

It is useful to define three to five specific activities that occur in each matrix box. Now fill in every box with a graphic representing whether or not (or to what degree) needed functions participated in each phase.

The benefits of the approach include an ability to see late commitments at a glance. Companies that notice the same pattern cropping up frequently probably have a glitch in their system. You can take the method further and calculate a "% concurrent" figure for each project. Assign a certain number of points per completed box. If all activities took place on time, the box is worth two points, say. Award one point for every box where only a portion of the tasks happened or happened on time, no points for a complete bust. Dividing the total score by the maximum possible score gives a percentage.



One concurrency matrix revealed that the system engineering and marketing functions were systematically under involved in the early phases of development.

An advantage of this exercise is it can highlight deficiencies that should become the first priority of any reengineering effort. Examining a concurrency matrix this way, says Goldense, led one manufacturer to redefine its marketing department. Marketers had merely been glorified salespeople in the old regime. The function changed to include product planners dedicated to assisting technical personnel starting in the early phases of product definition. Similarly, nobody in the systems engineering department was taking ownership of specific projects, discouraging early involvement. A more formal process solved the problem and helped management understand that systems engineering was really one of the firm's core competencies.

Staffing speed — A common malady of product development efforts is full staff ramp-up too late, typically in the "middle" of the project when the design is almost complete and it's time for a prototype.

One answer: Focus attention on the rate at which projects ramp up by graphing the number of involved people versus time. The idea is that if you start earlier, you will finish sooner. Goldense says fast project ramp-up will work if three things occur during the feasibility phase.

First, analysis and experimentation must cut development risk so the schedule can be predicted to an accuracy of ±20% or better. Second, customer requirements must be fairly complete and documented — in other words, they are not going to change much as the project proceeds. Third, the architecture of the product must be complete, though the specifications need not be. Use this information to confidently estimate project resources early on.

Most companies perform these activities, but don't carry them far enough to reduce development risk, says Goldense. This leaves too much work for the development phase. Conversely, they may estimate funding and resources early, but before the product has been sufficiently defined. The obvious result can be efforts that are drastically overdue and overbudget.

Earlier staff ramping can be beneficial even if planning isn't entirely complete. The act of assigning most professionals at the feasibility phase is likely to force expedited definition and planning. ■