



Concurrent Engineering Comes of Age

*By Brian K. Seitz,
Intellectual Arbitrage Group*

The engineering environment has changed in the past decade. Today, engineers participate in multiple projects, with multiple parties, in multiple roles and relationships. The lines between all facets of an engineering environment have blurred. Concepts such as virtual organizations and strategic alliances have turned what it means to be an enterprise upside-down. This is a good thing.

As we have moved into an age of electronic communication, virtual design teams, and concurrent engineering, the standard waterfall development methodology common in the age of mass production has been changed. Instead of a series of discrete activities cascading from step to step, concurrent engineering seeks to merge the activities of customer requirements, product design, and process design into a single, integrated activity.

Implementing Concurrency

In implementation, concurrent engineering requires cross-functional integration - teaming across typical organizational boundaries, be it inside a corporation or external. This teaming starts early in a project, when marketing, design and process engineering, and production personnel form a team to develop a product together.

Each group contributes its expertise as always. However, the issues that each must deal with are now addressed earlier and more broadly by the whole team. Couple this new process of development with disparate geographic locations, and new infrastructure requirements are exposed.

The Virtual Engineer

Take, for example, the hotel office. It's six o'clock, and you've arrived at your hotel in Detroit for a conference. You receive a fax telling you a change has been made: The client you've traveled 2,000 thousand miles and two time zones to meet has decided on a new design requirement, or has asked your company to investigate other product options "as long as you're here."

What do you do? Re-draw the product on a cocktail napkin? Pretend you never received the fax, thereby avoiding a less-than-professional presentation to the client? Not to long ago, this would have represented a missed opportunity (or at best a panic) back at the home office.

If this sounds all too familiar, scenarios like this happen all the time. Product life is shorter, design cycles are shorter, and there are more options to consider than ever. In short, the day of Henry Ford's maxim, "A customer can have any color he wants as long as its black" has reached its end. We are now living in the age of mass customization, the ability to mass-produce products for small market niches as low as in units of one.

New Tech Infrastructure

In order to address this new scenario, businesses are building a new infrastructure that merges advances in telecommunications (the Internet), microcomputers (lower cost of entry), and component-oriented software.

In particular, component-oriented software has taken the industry from an expensive craft with high development, integration, and operating costs into an age where these activities are more manageable. Technologies like Object Linking and Embedding (OLE) and the Distributed Component Object Model (DCOM) yield several benefits to the construction of an engineering environment. They enable a developer to reuse functions that previous programmers created, and build more sophisticated applications more quickly, using modular components.

Standard object and interface standards let applications "plug and play." This in turn allows engineering departments to choose applications as if they were modular components. Now, specialized packages can be installed that suit the needs of particular disciplines (e.g., electrical, mechanical, etc.), and at the same time, integrate with currently installed engineering environments.

The Future is Modular

The future, as trends indicate, will be a modular one, an environment of mass customization in organizations, systems, and products. The same is true for applications. No longer do application programmers write thousands of lines of redundant code, at least not in the Windows environment. Today's application developer works in a dual world where he or she exploits services developed by others, while creating services for others to exploit.

The result: a more competitive market for solution developers and a windfall for engineering organizations. In addition with the new communications advances, concurrent processes and business models, an engineer's tools and environment now offer undreamed-of flexibility. It is now up to the engineer to exploit this brave new world and craft it to his or her needs.

Originally Published in Software Strategies Magazine June, 1997