Programming regains importance during tough times

Practice Matters

By Satish Rao, Assoc. AIA

Clients have always known that architects can save them money by designing space that can be constructed economically. Now, a new generation of building owners and facility managers is catching on to the fact that employing architects to do programming may help them save even more. Clients who are looking for every possible way to cut costs know that "real estate costs can have a significant impact on profitability," according to Lori Walker, a principal at NBBJ Architects in Seattle. For the many architects who haven't done much programming, the following review of what programming does and how it produces value for clients should be useful.

Programming can help owners analyze the way they do business; how much space they need; how it will be organized; and what the cost implications of different alternatives will be. Architects can help them tighten operations, streamline work flow, and function more efficiently. This new interest in programming reflects the current U.S. economy. During the late 1990s, the owners of both institutions and high-tech companies were wooing prospective employees with generous and costly amenities, as well as trying to gain investors and enhance branding by using image-building architecture. Today, things are different. Lewis Goetz, FAIA, of Group Goetz Architects, a Washington, D.C., firm specializing in workplace environments, says, "Clients are now cutting back on such amenities and increasing the proportion of work areas." Private offices are giving way to open-plan, collaborative spaces, and work areas that enhance multitasking are gaining popularity. It is also clear that clients are reassessing how their companies function as organizations. They are eliminating redundancies and curtailing nonessentials as growth projections are getting more conservative. And, as stockholders force corporations to be more fiscally responsible than they were the 1990s, extravagant image-building architecture has fallen out of favor. But doing programming in this more conservative business climate means that there are new issues for architects to consider.

The first is to clearly describe the degree to which cost will be regarded as a design criterion. "Blue skying"—the practice of envisioning idealized and often unaffordable solutions—is less prevalent now. When the program is based on economic reality, it is easier to keep the client focused on the essentials. User requirements should be evaluated to verify that they coincide with the new goals of the organization. The program therefore must identify a first phase that will be self-sufficient in supporting the client's operation and not rely on future phases to function effectively.

Less urgent requirements can then be relegated to a later phase. In the past, when funding was often not in doubt, phasing was often merely the response to a deferred cash flow or anticipated growth. Now, phasing serves to identify what the client can initially afford to build, and later phases might never get built if the anticipated growth doesn't materialize.

Next, you need to learn everything possible about the client's operation and building type. If you are not familiar with the operational and economic dynamics of your client's business, immerse yourself in them before and during information-gathering interviews for the program. Some architects advocate becoming "squatters," that is, working at a company's facility for a few days or weeks in order to observe its work flow.

THE FISCALLY CONSERVATIVE CLIMATE MEANS CLIENTS NOW NEED ARCHITECTS TO CONSIDER DIFFERENT ISSUES.

Some businesses require more space during certain times of the year, so tune in to your client's annual business cycles. If they want to reduce square footage, you might suggest that they consider removing redundancies by dovetailing tasks—although you should make it clear that your purpose is not to help managers cut jobs.

The next important step in programming is data gathering, through interviews with the client, users, and other stakeholders. The current emphasis on running lean and mean has made the role of the client's chief financial officer more important than ever. Jasna Bijelic, Associate AIA, a principal at Davis, Carter, Scott, a Virginia A/E firm, says, "Architects are now using automated models." For instance, Kevin Pennington, project director at this firm, has created a customized Excel spreadsheet that can be formulated to quickly translate a change in square footage data into resulting changes in parking requirements, unit mixes, and so on. This dynamic program permits immediate assessment of the down-the-line financial impact of each proposed change. As a fundamental part of the programming process, architects should assess the cost implications of each new need or idea that emerges.

After you've started getting an idea about your client's work flow and what their financial priorities are, you can start developing a list of the spaces and support systems needed. This has changed, too. Multitasking is gaining in popularity, so space is no longer customized for specific uses. Standardized modular workstation types are the rule today. These enable work spaces to be adapted to the changing needs of staff, and increase flexibility, says Lewis Goetz. Dr. Andrew M. Laing, managing director of DEGW North America, an international design consultancy focused on the changing nature of work, supporting technologies, and places, goes further. "Technological developments,"

Satish Rao, Assoc. AIA, is director of project management for Torti Gallas and Partners, Fairfax, Va.

Basic Steps for Successful Programming

Programming transforms the client's goals into a statement of needs and design parameters. In case you haven't done it lately, here are the basic steps:

- **Determine the client's vision and goals for the project: What they want to achieve, and why.**
- **Interview client groups, including users, to gather and categorize data: occupancy parameters, space guidelines, site data. Be a facilitator; don't impose your preferences. Filter out data not relevant to design.**
- **If necessary, do background research on the client's operation and building type.**
- **Understand and analyze the processes: people, goods, information, services, circulation. Identify the required activities, spaces, equipment, furnishings.**
- **Distinguish between needs and wants; prioritize needs based on congruence with goals, beneficial returns, and costs. Determine phasing.**
- **Compile a set of requirements, both numerical and qualitative, for each indoor and outdoor space.**
- **Write a clear, concise statement of the problem and design goals. Address image and character, function, relationships, ambience, and costs.**

says Laing, "are enabling people to be increasingly mobile outside and within the office. This mobility challenges many conventional assumptions, such as that sharing workplaces is the exception rather than the rule." Traditional concepts of hard-copy document storage are changing, too. "Increasing attention is being given," continues Laing, "to knowledge management as an important component of office design, to ensure that paper as well as electronic information is managed collectively."

Teleconferencing affects the need for large conference rooms and big pinup surfaces, replacing them with more small conference rooms that double as work centers, equipped with sophisticated multimedia equipment and increased acoustical treatment.

Finally, once the problem has been defined and the client's needs have been identified and categorized, draft the program statement. This is the link between the program and the design. It must clearly and concisely convey the client's goals and needs and suggest alternative directions for design, without committing to a single solution. A program is not a series of problem statements, but must supply the designer with the information needed to create innovative solutions that will satisfy the client's objectives.

Programming, design are one

Many clients prefer to use specialists for programming and want to choose their design architects later. But continuity between programming and design has many benefits.

First, an architect who is knowledgeable in the design of a specific building type knows what information will be needed when the time comes for making design decisions. For example, complex buildings such as laboratories have highly interdependent building systems. If any aspect of this interdependency is neglected during programming, the design architects usually end *(continued on page 3)*

continued from page 64

up making revisions to the program at their own cost, to address issues that were not adequately covered by the programmer. If information is gathered in a holistic way, the architect can use it to make the right trade-offs between components.

Second, the client-architect relationship that is developed during programming gives each party valuable insights into the other's work processes, and this mutual understanding pays off during design. A third benefit is that time is saved when programming steps can be conducted concurrently with design. For example, a "schematic" building program can be written during predesign work such as master planning. Then the program is further developed before design development, incorporating the new ideas generated during schematics. This juxtaposition is much harder to achieve with an outside programmer. And, architects generally have access to more current cost data than people who only do programming, and they are also more familiar with the latest information on new building technologies. They can use this knowledge to make realistic assessments of program options based on both fiscal viability and buildability.

FOR COMPLEX PROJECTS, PROGRAMMING CAN NET EVEN MORE THAN AN OPTIMISTIC 25 PERCENT PROFIT.

All clients appreciate the economies and other benefits that are generated through programming—they yield the biggest bang for the construction buck—so it is likely that the interest in paying extra for the extremely valuable service will outlast the weak economy. And it can be profitable. For complex projects, programming can net more than even an optimistic 20 percent or 25 percent profit margin. But to achieve this, the programming contract should spell out the limits of work in detail and provide for additional fees if the listed tasks are exceeded. Because the programming process can take unexpected turns, guard against underestimating the effort involved. Since the programming process, and the level of effort, could vary substantially from one project type to another, architects have traditionally established fees from the "bottom up," by listing the tasks and estimating professional time required from experience.

Marketing is also an issue. Sometimes the programming precedes the public announcement of the project, so you must start marketing to potential clients early, sometimes before there is even a hint that a project is going to take off. To be successful in getting programming work, you must be alert to emerging opportunities and catch proposed projects before your competitors do. ■

Resources for Programmers

The classic text on programming is *Problem Seeking: An Architectural Programming Primer*, 4th edition, by William M. Peña and Steven A. Parshall (John Wiley and Sons, 2001). Robert G. Hershberger, Ph.D., FAIA, wrote an excellent article on programming for *The Architect's Handbook of Professional Practice*, 13th edition (John Wiley and Sons, 2001). A sample matrix shows how information can be organized.