what determines citizen involvement?

Ronald M. Jimmerson

Extension agents and others with responsibility for informal community programming share several concerns related to citizen involvement. Who should be involved? When, where, why, and how should citizens be involved in programming efforts? One answer is: "It depends on the programming situation."

Factors such as the complexity of the problem(s) the program is trying to solve, the number of feasible alternatives, the kinds of data available, the amount of accessible data, and the urgency of the situation all have an impact on citizen involvement.

Another factor often overlooked is our orientation as a programmer—that is, what are our beliefs, goals, and preferred methods that influence citizen involvement in our programming efforts? Our orientation has a direct impact on the type and extent of citizen involvement in our programming efforts.

I suggest that citizen involvement can be influenced by two major factors: (1) our personal orientation as a programmer and (2) the programming situation. Let's look at how these two factors promote effective or ineffective citizen involvement.

Programmer Orientation

As programmers, it's important to examine three aspects of our orientation that directly affect involvement of citizens in our programming efforts.

1. What are my basic *beliefs about individuals* and the individual's relationship to institutions and society? Two basic questions are:
   a. Do I view myself as more able than citizens to make decisions about programming or do citizens have the ability to develop their own programs?

Ronald M. Jimmerson: Assistant Professor, in Agriculture and in Adult and Continuing Education, Washington State University—Pullman. Accepted for publication: May, 1981.

b. Do I accept the idea that people are influenced by society and must adapt to it or do I believe individuals influence and change societies?

2. What *goals* or purposes do I have for programs?
   a. Do I focus on *product goals* that concentrate on the task and stress quality solutions?
   b. Do I concentrate on *process goals* that are more concerned with helping people and groups grow or develop as a result of carrying out programs?
3. What *methods* (style, approach, strategies, and roles) do I use to work toward accomplishing program goals? Two broad categories tend to follow logically from our beliefs and goals:
   a. Autocratic or top-down methods that place responsibility for program decisions on me (the programmer).
   b. Democratic methods that place much of the responsibility for programming decisions on citizen participants.

These three variables fit into a logical sequence that reflects certain basic beliefs about people and the methods used to reach program goals. Depending on our views and actions regarding these factors, we tend to align ourselves with one of two extreme "orientations." These are labeled "directive" and "developmental" and tend to have different influences on citizen participation.

Most programmers need a flexible orientation adaptable to the many unique programming situations they face. An orientation that allows the programmer to take responsibility when necessary and share responsibility with citizens when appropriate can promote both program goals and citizen growth. . . .

Many authors have discussed programmer roles, approaches, and functions. Robinson and Clifford discuss directive and developmental organizational styles in dealing with community groups.[1] Franklin outlines five change agent styles; four are considered noneducative and one an educative approach.[2] These authors believe that the "noneducative" or "directive" approaches limit citizen involvement. Fiedler, who describes two leadership styles ("relationship motivated" and "task oriented"), believes both styles are effective in achieving results, but the programming situation will determine which will be more effective.[3]

This discussion borrows ideas from these studies, but focuses on the effect of the two orientations on citizen involvement. Both orientations have distinct advantages. While it becomes clear that the developmental orientation allows for a broader range of citizen involvement, the directive orientation can save countless hours wasted on attempts to involve citizens at inappropriate times.

Keep in mind throughout this analysis that the two orientations are in sharp contrast and probably are seldom found in pure form. Most of us will have an orientation lying somewhere between the two extremes. However, in practice, we can usually place our orientation toward one extreme or the other. Thus, it's important to be aware of our orientation and its influence on how, why, when, and where we promote citizen involvement.

Directive Orientation

Beliefs

Directive programmers can be described as people who believe in themselves more than they believe in others. They feel most comfortable relying on themselves; maintaining control of their programs. They believe that most of their clients need guidance and direction in setting goals and carrying out activities to achieve these goals.

These beliefs lead to a top-down approach with the Extension organization and the programmer taking most of the responsibility for programming decisions. This orientation doesn't rule out citizen involvement, but will likely limit involvement to what Arnstein calls nonparticipation or tokenism.[4] That means citizens are involved to help legitimize or "sell" the programs without taking major responsibility for programming decisions.

Goals

The directive programmer views programming processes as means to an end, a way of achieving program goals. She/he doesn't see the process as particularly important in and of itself. The focus is on achieving program results, such as passing zoning regulations, establishing a downtown development plan, etc. People might be involved in actual decision making in situations where creative group effort adds to the quality of the decision. Involvement, however, isn't used specifically as an educative tool.

Methods

Methods used by directive programmers tend to be more autocratic than democratic. This follows logically since their beliefs and goals can best be carried out in this way. Because a quality program result is the highest priority and because of limited trust in group process, decisions can't be left to democratic processes.

Directive programmers pride themselves on their hard work and commitment to the program. Since they're self-reliant and trust their own judgments over those made by others, they involve people only to the extent necessary for getting their programs adopted. They pride themselves on the number of successful programs they've conducted using concrete program outcomes as criteria for program success.

Developmental Orientation

Beliefs

Developmental programmers see themselves as organizers of people. They view their expertise as being able to help people organize and make decisions about their own programs. This comes from a basic belief that people (under proper circumstances) are self-motivated, independent, responsible, creative, interested, and concerned about factors that affect their lives. They believe local people can and should influence the development of institutions and society.

These basic views of people lead developmental programmers to strive for citizen participation at all levels of program decision making. They seek development of programs from the bottom up with local citizens having significant control of the programming effort. Involvement in this case is seen as *us,* sharing in a program effort rather than *me,* the programmer, seeking support for *my* program.

Goals

The goals of the developmental programmer take into account process as well as products. Since the program is planned and carried out by the people, the programmer's main concern is that the process works effectively.

Because specific program goals can't be known when a program begins, it's the programmer's responsibility to help people decide program outcomes and work through the processes of achieving those outcomes. Goals, thereby, are concerned with helping people and groups gain the knowledge, skills, and confidence to carry out programs.

Methods

Methods used by developmental programmers become much more democratic than those of the directive programmer. Citizens are encouraged to make inputs and take on leadership roles. Cooperation is stressed so that the strength of each individual can be developed and used.

Typical roles taken by the developmental programmer are activator, facilitator, and encourager. Various other roles may be played, but always with the group process purpose in mind. For example, the programmer might play an abductor role (withdrawn from the group) to force the group members to take leadership roles or to break a dependent relationship.

Programming Situation

The differences in beliefs, goals, and methods of these two programmer orientations suggests that programmers with either orientation can fail to use citizen participation effectively unless a conscious effort is made to adapt their orientations to the programming situation.

We can view programming situations as either "closed" or "open." Closed situations have one or more of the following characteristics: concrete problems, few alternative solutions, hard data available, routine decisions, and immediate action needed. Open situations are those with: complex problems, many alternatives, soft data available (few concrete facts), and/or creative or negotiated decisions needed.

The advantages and limitations of directive and developmental orientations in open and closed programming situations are shown in Table 1.

Directive programmers can be relied on in closed programming situations because the problems lend themselves to solution by a competent programmer with little or no input from citizens. In these situations, citizen involvement generally isn't productive for achieving program results or for helping citizens grow. A developmental approach, on the other hand, wouldn't be effective in a closed programming situation.

Table 1. Programmer orientation vs. programming situation.

Programming situation	Programmer orientation	
	Directive	Developmental
Closed situation Concrete problems Few alternatives Hard data available Routine decisions Immediate action needed	*Productive* Fast, efficient problem situations Noninvolvement or token involvement is appropriate	*Nonproductive* Slow, inefficient problem solutions Nonproductive citizen involvement Programmer shirking responsibility
Open situation Complex problems Many alternatives Soft data available Creative or negotiated decisions needed	*Nonproductive* Inefficient use of citizen resources for problem solving Token or noninvolvement fails to take advantage of educational environment to promote citizen growth	*Productive* Efficient problem solving using citizen resources Involvement at all levels is appropriate for problem solution and citizen growth

Attempts to achieve democratic solutions would result in slow, inefficient solutions, as well as frustrated citizenry.

Developmental programmers can be most effective in open programming situations where complex problems, numerous alternative solutions, and the need for long-term solutions make citizen involvement necessary and efficient for problem solving. It also provides the setting for helping citizens grow as individuals and as groups. A directive approach in an open situation would likely result in inefficient use of citizen resources and failure to take advantage of an educational situation.

Summary

Most programmers need a flexible orientation adaptable to the many unique programming situations they face. An orientation that allows the programmer to take responsibility when necessary and share responsibility with citizens when appropriate can promote both program goals and citizen growth. By carefully analyzing our orientation (beliefs, goals, and methods) and each programming situation, we can develop a basis for answering many of the citizen involvement questions.

Footnotes

[1] Jerry W. Robinson and Roy A. Clifford, *Organizational Styles in Community Groups* (Urbana, Illinois: North-Central Regional Extension Publication No. 36-2, July, 1974).

[2] Richard Franklin, *Toward the Style of the Community Change Educator* (Washington, D.C.: NTL Institute for Applied Behavioral Science, 1969).

[3] Fred E. Fiedler and Martin M. Chemers, *Leadership and Effective Management* (Glenview, Illinois: Scott, Foresman, 1974).

[4] Sherry R. Arnstein, "A Ladder of Citizen Participation," *AIP Journal,* XXXV (July, 1969), 216-24.