CHAPTER THREE

USING LOGIC MODELS

John A. McLaughlin, Gretchen B. Jordan

Those who are responsible for designing, conducting, reporting, and using program evaluations are the primary audience for this chapter. We believe that program managers and staff will also find the logic model tool useful for conceptualizing, planning, and communicating with others about their program. The logic model serves as a useful advance organizer when evaluators and others are designing evaluation studies and performance measurement systems. It helps them to focus on the important elements of the program and to identify what evaluation questions should be asked and why and what measures of performance are key. The logic model in various forms has been around since the late 1960s, but it has come into increasing use in the past two decades because of the emphasis on managing for results and measuring performance. The logic model also helps evaluators frame evaluation reports so that findings from the evaluation and measurement can tell a performance "story" and results can be linked to program elements and assumptions about them. Evaluators can use this tool when asked to evaluate a program during its design phase, after it has ended, or at any other point in its life cycle. Managers may use this tool in program planning and design and when communicating the place of the program in a larger organization or context. The process of developing a logic model helps build shared understanding and expectations among program staff and other participants. We believe that while it is important for the evaluation to address the question of program results, it is equally important to focus the evaluation efforts on program implementation and early outcomes so the managers and staff know what's working and not working and where necessary make informed mid-course corrections to enhance the probability of longer-term success.

We use the term *program* loosely throughout this chapter. We have used logic models to describe internal management functions, websites, and the performance-based management process itself. A program can be described as an intentional transformation of specific resources (inputs) into certain activities (processes) to produce desired outcomes (results) within a specific context. We present a tool that evaluators and program managers can use to describe the unique program elements and show how they go together. This completed model can then be used for the purposes of communicating and testing the assumptions that program staff members have made about how the program is supposed to work.

A program can also be thought of as a hypothesis or theory of change: if a program is implemented as planned, then certain results are expected to follow, given the context within which it is implemented. Logic modeling is a tool that can be used to unpack this hypothesis in order to understand the underlying assumptions and create strategies to test the hypothesis.

The material in this chapter supports subsequent chapters in several ways. One of the assumptions that evaluators make is that a useful evaluation approach is based on an understanding of the objectives of the program and of the ways in which the program intends to achieve these objectives. Conducting an evaluation of a program without this understanding can be both costly and potentially harmful. Logic modeling can be a useful tool for performing an evaluability assessment. It can serve as an advance organizer for designing and conducting an implementation evaluation. The model presents a description of how the program staff members or other stakeholders believe the program works. If the evaluation finds that the program is successful in achieving its aims but works differently in practice, the logic model may be revised. If the evaluation determines that the program is not successful, it may be possible for the evaluator to recommend that the staff exert more pressure on the actual delivery of the program to bring it in line with their logic. Collecting and interpreting evaluation information is also aided by the logic model, as it establishes a framework for understanding the elements of the program, the assumed causal relationships, and the potential role of context. Finally, using the logic model in preparing and presenting the evaluation findings and recommendations can increase the probability that the evaluation results will be used.

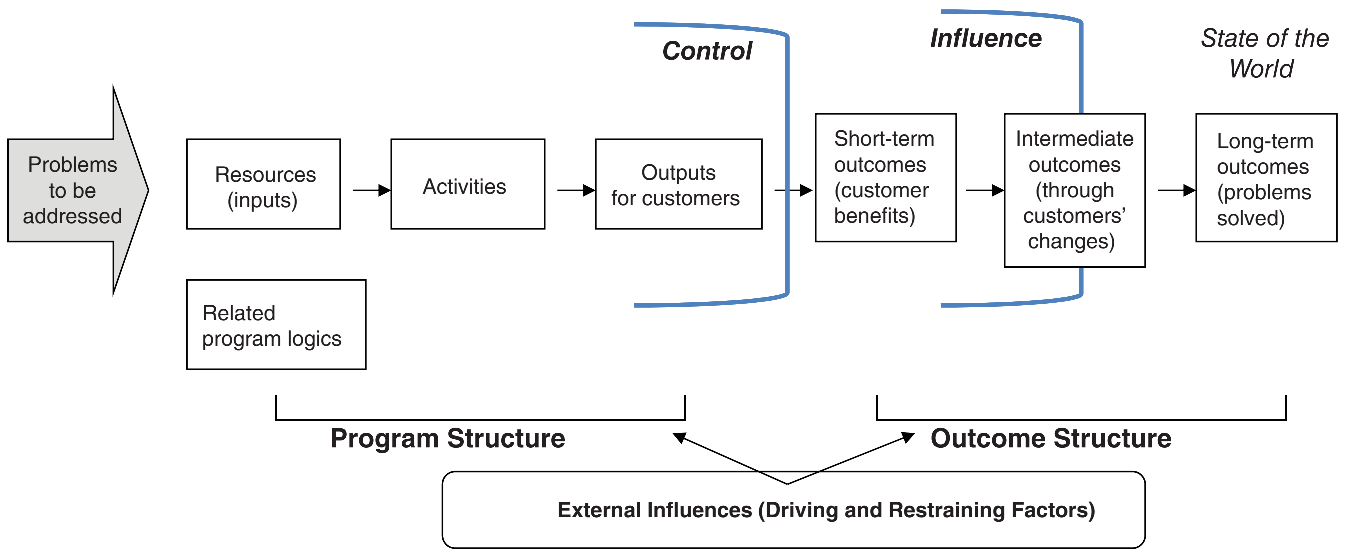
What Is a Logic Model?

A *logic model* is a plausible and sensible model of how a program will work under certain environmental conditions to solve identified problems (Bickman, 1987). It can be the basis for a convincing story of the program's expected performance, telling stakeholders and others the problem the program focuses on and how the program is qualified to address it. The elements of the logic model are resources, activities, outputs, short-term outcomes, intermediate outcomes, and long-term outcomes (Wholey, 1987). Some have added the customers reached to the logic model, as well as the relevant external contextual (antecedent and mediating) influences (McLaughlin and Jordan, 1999). (A historical review of logic modeling as a tool for planning and conducting evaluations can be found in McLaughlin and Jordan, 2004.) The interest in logic modeling has spawned a number of books and guides with instructions and examples. These include Centers for Disease Control (2010), Frechtling (2007), Funnell and Rogers (2011), Knowlton and Phillips (2008 and 2013), Taylor-Powell (2008), and W.K. Kellogg Foundation (2005).

Logic models can take many different forms, including diagram, narrative, and tabular forms. Evaluators can prepare a logic model at any time in the life cycle of a program, and they often revise this model as more program information is collected. A basic logic model is shown in Figure 3.1. It has three basic parts: program structure, outcomes structure, and context. These are consistent with the desirable dimensions of performance measurement and evaluation. That is, the goal of evaluation is to observe and explain change. The necessary information for explanation comes from performance measurement in the program and outcomes structure and context. Here are descriptions of the elements of the logic model:

- *Resources*: human and financial resources as well as other inputs required to support the program, such as partnerships. Information on the type and level of the problem addressed by the program is an essential resource for the program.
- *Activities*: the essential action steps necessary to produce program outputs.
- *Outputs*: the products, goods, and services provided to the program's direct customers or program participants. For example, the reports generated for other researchers or the number of clients completing a workshop could be outputs of an activity. Customers or "reach" is sometimes put explicitly in the middle of the chain of logic. Many evaluators do not separate out activities and outputs in their models. However, activities typically represent what the

FIGURE 3.1. BASIC LOGIC MODEL.



program does, whereas outputs are what the program produces, and so we like to break the two out because this supports implementation evaluation.

- Resources, activities and outputs are the *program structure*, and these elements of the logic are determined during the design phase and modified as experience is gained during implementation. Program structure is mostly under the control of program managers and staff, whereas short and intermediate outcomes depend upon action by the recipients of outputs, and programs can influence but not control those.
- *Outcomes*: changes or benefits to people, organizations, or other program targets that are expected to result from their being exposed to activities and outputs. Programs typically have multiple, sequential outcomes, sometimes collectively called the program's *outcome structure*. First, there are short-term outcomes, the changes or benefits most closely associated with, or "caused" by, the program's outputs. Second are the intermediate outcomes, which are expected to result from the short-term outcomes. Long-term outcomes or program impacts are expected to follow from the benefits accrued though the intermediate outcomes. For example, a teacher training program might have the following outcome structure. As a result of participating in training, teachers learn new skills and knowledge about classroom management techniques (the short-term outcome). Then they appropriately apply these new skills in their classrooms (the intermediate outcome), which leads to enhanced educational opportunities for the students, resulting in improved learning (the long-term impact the teacher training program was designed to achieve).

Key contextual factors external to the program and not under its control may influence its success either positively or negatively and are critical features of the logic model. Two types of context influence the design and delivery of the program: antecedent factors and mediating factors (Harrell, Burt, Hatry, Rossman, and Roth, 1996). *Antecedent* factors are those the program starts out with, such as client characteristics, geographical variables, and economic factors. *Mediating* factors are the influences that emerge as the program unfolds, such as changes in staff, new policies, a downturn or uptick in the economy, and new competing or complementary programs. Program planners and evaluators must take these into consideration when creating and evaluating the program. It is particularly important to consider how certain client characteristics might influence the outcome of a program. For example, if the program were designed to increase the reading skills of adult immigrants, the developer would consult the related research to identify useful instructional methods for adults from the countries involved.

The Utility of Logic Models

The utility of logic models has increased as managers are being challenged by oversight agencies at all levels of government and in the nonprofit sector. At the federal level, Congress and the White House Office of Management and Budget are asking managers to tell their programs' stories in a way that both communicates the program's outcome goals and shows that these outcomes have been achieved. For many public programs, there is also this implicit question: Are the results proposed by the program the correct ones? That is, do the results address problems that are appropriate for the program and that stakeholders deem to be important to the organizational mission and national needs?

The emphasis on accountability and managing for results that is now found in state and local governments and in nonprofit organizations such as the United Way of America and the American Red Cross represents a change in the way managers have to describe their programs and document program successes. In the past, program managers were not as familiar with describing and measuring outcomes as they were with documenting inputs and outputs. Program managers and evaluators have not been in the habit of using clear, logically consistent methods to make explicit their understandings about programs and how those programs work to achieve their outcomes given their specific operating contexts.

There is an increasing interest among program managers in continuous improvement and managing for quality. Yet, choosing what to measure and then collecting and analyzing the data necessary for improvement are new to many managers. Although tools such as flowcharts, risk analysis, and systems analysis can be used to plan and describe programs, logic models more comprehensively address the increasing requirements for both outcomes measurement and measurement of how the program is being implemented to allow for improvement. (Box 3.1 summarizes the benefits of using a logic model.) The logic modeling process can also be used by managers of existing programs to enable program staff members to step back and reexamine their existing program, asking, for example: Are the challenges the program is responsible for still relevant? Have they changed in any way? Are the strategies we have been using consistent with prevailing professional practice? Should we consider other approaches? Have new partners or technologies emerged that we can take advantage of? For planning, the logic model is worked from desired outcomes to chosen activities, whereas for evaluation and measurement, the important viewing is from the activities to the emerging outcomes.

Box 3.1. Benefits of Using the Logic Model Tool

- It points to evaluation issues and a balanced set of key performance measurement points, thus improving data collection and usefulness and helping managers and staff to meet performance reporting requirements.
- It helps with program design or improvement by identifying program activities that are critical to goal attainment, are redundant, or have inconsistent or implausible linkages to program goals.
- It communicates the place of a program in the organization or problem hierarchy, particularly if there are shared logic charts at various management levels.
- It builds a common understanding of the program and expectations for resources, customers reached, and results, and thus is good for sharing ideas, identifying assumptions, team building, and communication.

One of the uses of the logic model that should not be overlooked is communication. The process of developing a logic model brings people together to build a shared understanding of the program and program performance standards. The model also helps to communicate the program to those outside it in a concise and compelling way and helps program staff to gain a common understanding of how the program works and their responsibilities to make it work.

Logic models are increasingly used for program design and management. This usually requires development of a theory of change and means that more resources and time are needed to complete and update the logic model. The traditional linear, cause-and-effect logic models emphasize activities or sequence of outcomes and are often used for evaluability assessment, evaluation planning, or outcomes assessment. Logic models such as the ones put forth by Funnell (2000) and Funnell and Rogers (2011) are more dynamic, and they include behavioral change, risk, context, and mediating variables. These models take more time to develop but have added utility as integrating frameworks for evaluation and performance measurement.

Theory-Driven Evaluation

Assumptions about resources and activities and how these are expected to lead to intended outcomes are often referred to as *program theory*. A logic model is a useful tool for describing program theory and is often referred to as describing the program's *theory of change*. The program's theory of change evolves through persistent dialogue between the evaluator and program representatives. The logic model is then used to provide a graphic description of the theory of change. The hypothesis, often implicit, is that if the right resources are transformed into the right activities for the right people, then these are expected to lead to the results the program was designed to achieve. Some evaluators believe that making explicit the underlying assumptions about how a program is supposed to work increases the potential for evaluation utility. Although developing the program theory prior to the evaluation is considered most beneficial for predicting relationships, developing program theory at any point in the evaluation helps explain observed causal relationships.

The aim of evaluation is to explain observed performance. According to Mayne and Stern (2013), "explanation is impossible without theory." Logic modeling is a process through which evaluators can tease out theory through deep discussions with mangers and staff regarding their assumptions and beliefs about how their program will work or is working to achieve intended outcomes and results. The discussion between evaluators and staff centers on relationships between various elements in the program's performance spectrum. After these causal associations are sufficiently described the role of the evaluators is the design and conduct of a study that tests the degree to which the hypothesized relationships hold up. The "tests" might occur at multiple points in the life cycle of a program: during the design stage, at implementation, emergent outcomes, and end outcomes or results.

Leeuw (2003) provides an excellent review of three approaches to restructuring program theories after the program has been implemented:

- The *policy-scientific approach* is more empirical than the other approaches and consists of generating a series of propositions, or assumptions, about how the program is supposed to work. The evaluator then tests these propositions through a review of relevant scientific research, interviews with key staff, and document reviews.
- The *strategic assessment approach* is driven through conversations or dialogues with program staff and participants. The focus is to draw out the underlying assumptions about how the program works and then subject these to open debate among stakeholders and staff.
- The *elicitation approach* aims at recovering the mental models or cognitive maps that program staff hold about their program. The various maps are then compared, contrasted, and assessed for their validity through open dialogue and reviews of existing related research.

The central theme in all three approaches is discovering the underlying *theory* and assumptions held about how the program is believed to be working to achieve its outcomes and then testing these assumptions once they have been made public. All three approaches make the program transparent, allowing the evaluator and others to see how it is supposed to work or thought to be working from multiple perspectives. Logic modeling is a tool that can effectively be used to display the assumption pathways. Most evaluators do not actually enter statements about the underlying theory in the model. The model is a *graphic* representation of the elements and how they go together. The arrows connecting the elements (see Figures 3.1 and 3.3) represent the theory or assumptions. (Box 3.2 offers some tips to consider before starting to construct a logic model.)

Building the Logic Model

A logic model is constructed in five stages:

Stage 1: collecting the relevant information

Stage 2: clearly defining the problem the program will solve and its context

Stage 3: defining the elements of the program in a table: early sense making

Stage 4: drawing the model to reveal the program's theory of change

Stage 5: verifying the program logic with stakeholders

Box 3.2. Tips to Consider Before Starting

- Think of developing a logic model as a process. In general, it is important that program managers and staff be involved in developing their logic model. They should be able to "do it themselves" after having had training in the logic modeling technique.
- Do not try to do the job alone. It is important to involve a workgroup with a full range of key stakeholders who are associated in some way with the implementation of the model and its results.
- Be careful with jargon. Because logic modeling is often a new way of thinking about the program, using familiar language helps others understand it. The format and terminology used in creating the logic model should be adapted to the program.

- View logic modeling as part of long-term cultural change. Do not shortcut the process. Make the model an iterative process, updating it as program and program context change.
- *Avoid* letting the logic modeling process become a time sink. Leave some elements unknown. Plan costs and a schedule that can include downstream activities such as choosing performance measures or planning next steps.

Stage 1: Collecting the Relevant Information

Building the logic model for a program should be a team effort in most cases. If the evaluation function is external to the program, the evaluator, in collaboration with the program manager and staff, should carry out the process of creating the model. If the program manager does the work alone, there is a risk that she may leave out or incorrectly represent essential parts because she has limited knowledge of the program or its context. There are times when a manager may push back on the use of the logic model. The evaluator should be prepared to talk about the potential benefits of logic modeling such as those described in this chapter. In particular, we advise taking a step back to review the rationale of existing programs. It is often valuable to revisit the underlying assumptions of prevailing practice to explain why a program might be working well or might need improvements. In the end, deep engagement in the process is the best way to demonstrate utility.

In the following stages of building the logic model, we refer to the manager as the key player. However, we also recommend that persons knowledgeable about the program's planned performance, including partners and customers, be involved in a workgroup to develop the model. As the building process begins, it will become evident that there are multiple realities or views of program performance. Developing a shared vision of how the program is supposed to work will be a product of persistent discovery and negotiation between and among stakeholders.

When a program is complex or poorly defined or communication and consensus are lacking, we recommend that a small subgroup or perhaps an independent facilitator perform the initial analysis and synthesis through document reviews and individual and focus group interviews. The product of this effort can then be presented to the larger workgroup as a catalyst for the logic model process.

Whether designing a new program or describing an existing program, it is essential that the evaluator or workgroup collect information relevant to the program from multiple sources (see Box 3.3 for some tips). The information will come in the form of program documentation and from interviews with key stakeholders internal and external to the program. Although strategic plans, annual performance plans, previous program evaluations, pertinent legislation and regulations, and the results of targeted interviews should be available before the logic model is constructed, this will be an iterative process requiring the ongoing collection of information. Conducting a literature review to gain insights on what others have done to solve similar problems and on key contextual factors to consider in designing and implementing the program can reveal important evidence as to whether or not a program approach is correct. All those involved in the process, and particularly the evaluators, should adopt the role of skeptic, repeatedly asking why they should believe that a particular relationship is true, or conversely, why a step in the logic may not happen.

Box 3.3. Tips on Collecting Relevant Information to Build a Logic Model

- Interview people associated with the program, starting with those closely associated with its design and implementation and then moving to others either affected by the program or having a stake in its results. (Evaluators might find the interview guide developed by Gugiu and Rodriguez-Campos, 2007, helpful when facilitating this interviewing.)
- Analyze documents with a small group, perhaps assisted by an independent facilitator, especially for complex, poorly defined programs or where communication and consensus are lacking.
- Stay alert to changes in the context that could influence performance, such as staff turnover, new policies, or changes in the economy.

Stage 2: Clearly Defining the Problem and Its Context

Clearly defining the need for the program is the basis for all that follows in the development of the logic model. The program should be grounded in an understanding of the problem that drives the need for the program. This understanding should be expressed in clear descriptions of the overall problem and any ancillary problems, of who is involved, and of the factors that "cause" the problem. The program will address some or all of these factors to achieve the longer-term goal of solving the problem. (Box 3.4 offers some tips on problem definition.)

For example, there are economic and environmental challenges related to the production, distribution, and end use of energy. U.S. taxpayers face problems such as dependence on foreign oil, air pollution, and the threat of global warming from the burning of fossil fuels. The causal factors that might be addressed to increase the efficiency of the end use of energy include limited knowledge, risk aversion, consumers' budget constraints, lack of competitively priced clean and efficient energy technologies, externalities associated with public goods, and the structure of U.S. electricity markets. To help solve the problem of economic and environmental challenges related to the use of energy, a program would choose to focus on one or more factors related to developing clean and efficient energy technologies and changing customer values and knowledge.

One of the greatest challenges that workgroups developing logic models face is describing where their program ends and others start. For the process of building a specific program's logic model, the program's performance ends with addressing the problem it is designed to solve with the resources it has acquired, with recognition of the external forces that could influence its success in solving that problem. Generally, the manager's concern is determining the reasonable point of accountability for the program. At the point where the actions of customers, partners, or other programs are as influential on the outcomes as the actions of the program are, there is a shared responsibility for the outcomes and the program's accountability for the outcomes should be reduced. For example, the adoption of energy-efficient technologies is also influenced by financiers and by manufacturers of those technologies. Not recognizing these other factors reduces the probability for long-term success.

Box 3.4. Tips on Defining the Problem Addressed by the Program

- Look for what drives the need for the program. Some evaluators put client and customer needs as the first point in the model.
- Define all the major factors that "cause" the problem.
- Define the factors that the program addresses. Factors that "cause" the problem but that aren't addressed by the program are part of the context within which the program operates.
- Determine whether the program can be modified to address or take advantage of the contextual factors identified.
- Identify possible performance partnerships with other programs or organizations whose results affect those of the program.
- If necessary, reflect legislative language, perhaps by adding an additional layer of outcomes.

When defining the problem, it is important to examine the external conditions under which a program is implemented and how those conditions affect outcomes. Such an examination clarifies the program's niche and the assumptions on which performance expectations are set. Understanding program context provides an important contribution to program improvement (Weiss, 1997). Explaining the relationship of the problem addressed through the program, the factors that cause the problem, and external factors should enable the manager to argue that the program is addressing an important problem in a sensible way. Those developing the logic model must not only elicit the key external, or contextual, factors but also develop measures for the most critical of these factors so that these data can be factored into discussions about the program results.

One reason why it is important to recognize contextual factors before the program starts is that the program may be able to do something about them. For example, we once were asked to participate in the evaluation of a preservice teacher training program before it started. When we met with program staff, we began the logic modeling process to get a grasp on how they thought the program might work. One outcome identified was that student teachers would practice technology integration in their practicum sites. We asked if there were any factors that could influence reaching this outcome. Staff members said that participating classroom teachers would have to be skillful in the use of technology. As a result of this interchange, the staff decided to amend their initial logic to include training for classroom teachers who would be working with the preservice teachers.

Many of the problems that programs or organizations address are highly complex, resulting from a number of causal factors. Most programs are uniquely qualified to address a few of these factors, but if the problem is to be solved, then many of these factors must be addressed. We recommend that program staff identify all the factors that need to be addressed and then develop performance partnerships with other programs whose mission is to solve the same problem. Until the performance partnerships are established, all factors that are not under the control of the program fall into the context and may have a negative impact on the program's long-term success. For example, many federal programs depend on state and local programs to carry out policies established at the federal level. One of the performance goals of the U.S. Environmental Protection Agency is to ensure the availability of clean and safe water. This will not happen if states and localities do not develop and enforce guidelines for protecting sources of water.

Stage 3: Defining the Elements of the Program in a Table: Early Sense Making

The purpose of this stage is to uncover all the salient elements of the program. We find it helpful to introduce this stage by comparing it to the first step in completing a puzzle. The first action after opening the box is to dump all the pieces on the table that will make up the puzzle. Then you begin by sorting them into piles of like colors and shapes. The aim is putting the puzzle together but you first have to get an idea of what the pieces look like. This step in building a logic model requires the workgroup to categorize the information collected into "bins," rows, and columns in a table. The manager and other workgroup members review the information and tag each "piece" as a resource, activity, output, short-term outcome, intermediate outcome, long-term outcome, or external factor. Because they are building a model of how the program is intended to work, not every program detail has to be identified and catalogued, just those that are key to enhancing program staff and stakeholder understanding of how the program works.

Just like in puzzle building the modeling team begins to look for relationships between and among the pieces. The group organizes the elements in the table into chains of activities, outputs, and outcomes. (Box 3.5 offers some tips for this process.) An example of a logic model table for a middle school science technology engineering and math (STEM) program is shown in Figure 3.2. In this case the columns display the elements of the logic model and the rows show the program's description of significant program content associated with each element. This is, of course, a simplification of a nonlinear process.

Box 3.5. Tips on Defining the Elements of the Logic Model in a Table

- As you are categorizing elements of the logic model, define the target audiences and expected effects of the program for each.
- Put the outcomes into a sequence.
- Map both forward and backward to develop and check logic and assumptions. Ask questions such as, How do [did] we make this happen? Why do [did] we do this? If this, then that? If that, then what?
- Check for associations with other programs and partners for resources, delivery, or take up.
- Combine and summarize program elements, limiting the number of activity groups to no more than five to seven. These groupings are the program strategies that are expected to lead to results.
- *Avoid* giving the impression that program results occur in a linear process, even though they appear linear in the table format. Showing multiple rows feeding into one outcome and coloring rows to indicate the timing of events are possible ways to do this.

FIGURE 3.2. EXAMPLE OF A LOGIC MODEL TABLE FOR INCREASING THE NUMBER OF QUALIFIED GRADUATES ENTERING STEM CAREERS.

Resources	Activities	Outputs for Customers	Short-Term Outcomes	Intermediate Outcomes	Long-Term Outcomes
Need for increased STEM graduates Evidence-based STEM pedagogy Teachers, scientists, administrators, parents, middle school students Funding from federal and state agencies	Implement professional development for middle school teachers and scientists participating	Trained teachers and scientists	Middle school teachers and scientists have knowledge, skills and values to implement and support new STEM curriculum	Teachers and scientists accurately apply evidenced-based STEM in the classroom and prepare home activities for parents	
	Teachers, scientists conduct orientation sessions for school administrators and parents	School administrators, middle school students' parents complete orientation	Administrators and parents understand their roles in supporting STEM program	Administrators and parents carry out their responsibilities with fidelity	
Technologies	Install middle school STEM program with fidelity	Middle school students take advanced STEM courses, attend camps, clubs	Middle school students improve STEM learning and see value in pursuing a STEM career	Students elect to take advanced STEM courses and opt for STEM career paths in college	The number of STEM college graduates who choose STEM careers increases
External Context: Driving and restraining contextual factors such as state/federal policy, funding priority shifts, available technologies, competing/complementary school programs, need for scientists, changes in school leadership					

As the elements of the logic model are being gathered, the manager, evaluator, and workgroup should continually check the accuracy and completeness of the information contained in the table. The checking process is best done by determining whether representatives of key stakeholder groups can understand the logical flow of the program implementation from resources to solving the longer-term problem. Thus, the checking process goes beyond determining if all the key elements have been identified to confirming that, reading from left to right (or top to bottom), there is an obvious sequence or bridge from one column to the next.

One way to conduct the check is to ask *how* and *why* questions. Start with an entry in any column in the table, and ask, in effect, "How did we get here?" For example, select a particular short-term outcome and ask, "Is there an output that leads to this outcome?" Or, ask "Why are we aiming for that outcome?" The answer should lie in a subsequent outcome in the intermediate or long-term outcome column. Ask such questions at any point in the logic model performance spectrum, from inputs or resources to outcomes or results. The process of asking how and why questions is sometimes called *forward and backward mapping*.

Another good way to elicit information from the workgroup is to ask why an outcome might *not* occur, a reverse logic. What are the non-program factors that will prevent success (where what success looks like has been carefully defined)? Sue Funnell (2000) suggests a *logic model matrix* to capture these aspects of the logic. For example, looking at the first row in Figure 3.2 suggests a number of non-program factors that could prevent accomplishing the desired outcomes, factors such as a lack of scientific and technical personnel in the area. If the program manager and workgroup think of a non-program factor that is particularly critical and know of no one else who is addressing that factor, the design of the program and its logic may have to change to address it.

Last, note that at the bottom of the table the workgroup has recorded factors from the implementation context that might have a positive or negative influence on the success of the project. These influences might be found in searching through previous evaluation studies or the professional experiences of the program designers. The contextual factors are identified throughout the modeling process by asking, "What else must happen to enable this event to occur?" If those factors are not included in the program then they become part of the context and the evaluation should examine the extent to which they actually influenced to program success. Finally, contextual factors can become an influence at any point along the program's performance spectrum.

Stage 4: Drawing the Logic Model to Reveal the Program's Theory of Change

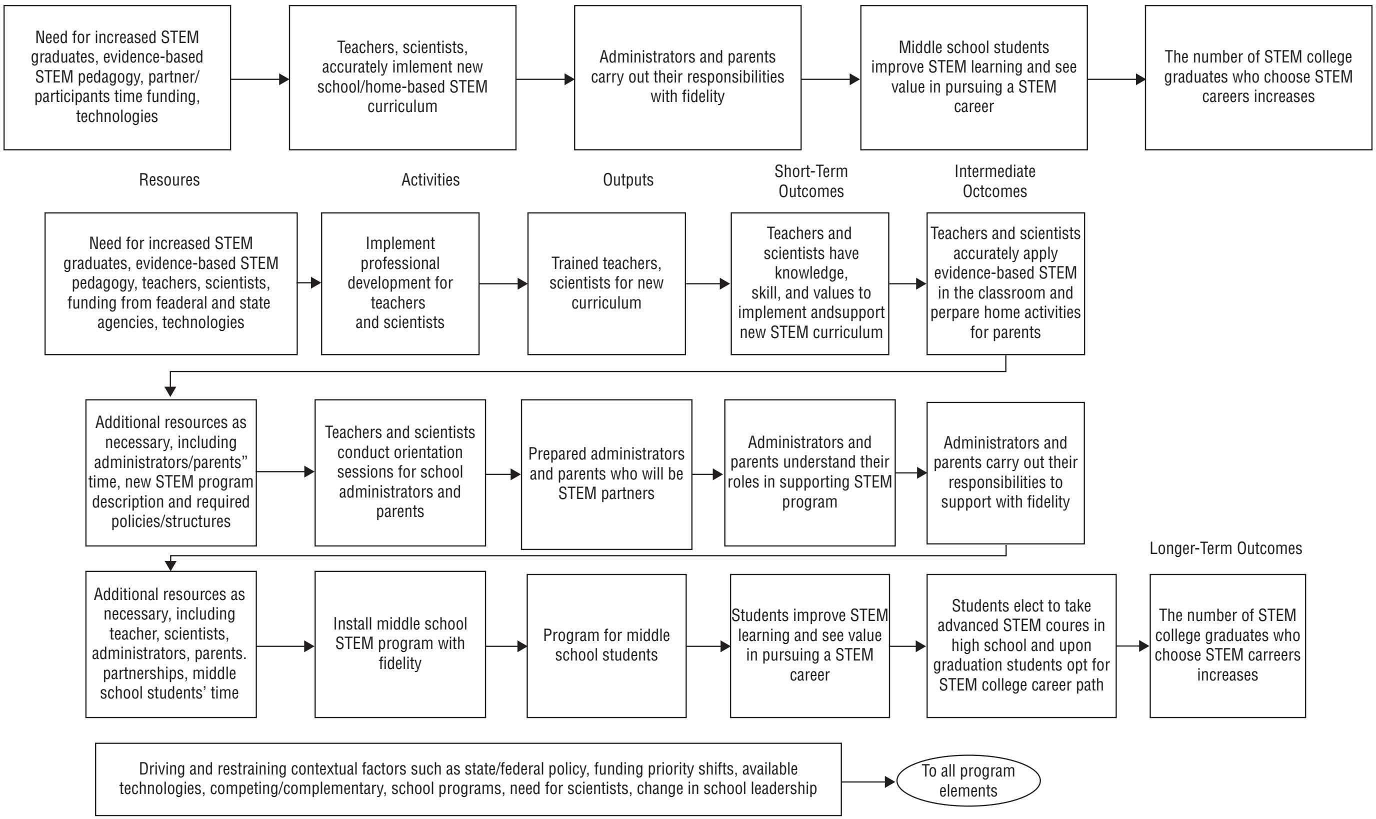
The logic model captures the logical flow and linkages that exist in any performance story uncovering the hypothesized theory of change. Using the program elements in the table, the logic model further organizes that information, enabling an audience to better understand and evaluate the hypothesized linkages. Whereas resources, activities, and outcomes are listed in separate columns in the table, they are specifically linked in the model, so that an audience can see exactly which activities lead to what intermediate outcomes and which intermediate outcomes lead to what longer-term outcomes, or impacts. Remember that this graphic depiction is only a picture of how the program is supposed to work given its implementation context. The text associated with the model will describe the program theory and rationale for believing that certain elements go together in a certain way. The text should not only describe experiences but should relay relevant previous research and evaluation studies.

There are several ways to present a logic model, but usually it is set forth as a diagram with columns and rows of boxes containing abbreviated descriptions and causal linkages shown with connecting one-way arrows. We place program inputs or resources in the far left-hand column and the long-term outcomes and problem to be solved in the far right-hand column. The boxes in the second column show the major program activities. In the subsequent columns the intended outputs and outcomes from each activity are shown, and these boxes may also list the intended customer for each output or outcome. Another common format displays the logic top to bottom rather than left to right, usually with resources and activities at the top and the goals at the bottom of the model.

An example of a logic model for a middle school STEM program is depicted in Figure 3.3. Two levels of the same logic are shown in Figure 3.3: a very high level in the single line at the top, and below that a more detailed version showing the relationship of multiple program elements in a Z-shaped pattern, where one set of activities and outcomes leads to another. The rows are created according to activities or activity groupings. If there is a rough sequential order to the activities, as there is when the accomplishments of the program come in stages, the rows, reading from the top to bottom of the diagram, will reflect that order. When the outcomes from one activity serve as a resource for another activity chain, an arrow is drawn from those outcomes to that activity chain. The arrows represent the expected causal relationship between the program elements. The last in the sequence of activity chains identifies the efforts of participating students to gain STEM-related knowledge, skills, and values leading to developing an increased inclination to follow a STEM career path resulting in an increase in the probability of more STEM professionals available for the workforce.

Rather than using a sequence of activities, a program could take a multifaceted approach, using several concurrent strategies to tackle a problem. The arrangement of the boxes in the model would reflect that. For example, a program might prepare teachers scientists, administrators and parents concurrently and then implement the school/home-based program once the partnership has been established. Although the example in Figure 3.3 shows one-to-one relationships among program elements, this is not always the case. It may be that one program element leads to more than one outcome, all of which are of interest to stakeholders and are part of describing the value of the program. For example, the United Way might have identified infant mortality as a critical problem that needs to be addressed. One of its partners, a school system, discovers that several teenagers in a school are pregnant, and an after-school program is designed to address the needs of these students.

FIGURE 3.3. A Z LOGIC MODEL FOR INCREASING THE NUMBER OF QUALIFIED GRADUATES ENTERING STEM CAREERS, AT TWO LEVELS OF DETAIL.



The outcomes of the program are increasing the participants' knowledge and skills related to prenatal health and caring for newborns. The impact of the program is a reduction in infant mortality in the community.

Activities can be described at many levels of detail. Because models are simplifications, activities that lead to the same outcome may be grouped to capture the level of detail necessary for a particular audience. As mentioned previously, a rule of thumb is that a logic model should have no more than five to seven activity groupings. Most programs are complex enough that logic models at more than one level of detail are helpful. A logic model more elaborate than the simple one shown in Figure 3.1 can be used to portray more detail for all or any one of the simple model's elements. For example, research activities may include performing literature reviews, conducting experiments, collecting information from multiple sources, analyzing data, and writing reports. These can be grouped and labeled as research. However, it may be necessary to formulate a more elaborate description of research sub activities for the staff responsible for research or for a stakeholder group with a specific interest in a research area. For example, funding agencies might want to understand the particular approach that will be employed to answer key research questions.

The final product may be viewed as a network displaying the interconnections between the major elements of the program's expected performance, from resources to solving an important problem. External factors that influence the success of the program may be entered into the model at the bottom, unless the program has sufficient information to predict the point at which they might occur. These external factors serve to record the assumptions that went into the development of the model. They are helpful for people not familiar with the program and for evaluators and staff when using or revising the model. Remember that the logic model is simply a graphic representation of the essential program elements and how they go together. The underlying program theory—why they go together—must be discussed, challenged, and then recorded in accompanying text.

Here are cautions about this step from our experience. Completed logic models are deceptively simple. In reality, it takes many drafts to describe the essence of a program. It can help to plan to have both simple forms of the diagram and more complex diagrams. Stakeholders unfamiliar with a program need a simple version. In this one, limit the words in the diagram. Provide more detail in separate charts or a written narrative. Limit the number of arrows, showing only the most critical relationships and feedback loops. Include outputs to external customers only, collapsing internal outputs such as management plans to one activity group or a separate document. Leave organizational charts separate, but use the same activity descriptions in both. In addition to this simple diagram, consider having more than one model with different levels of detail, different groups of activities, different levels at which performance could be measured, different stakeholder views, and different theories. Whatever the level of detail, avoid even the appearance that this logic model is set in stone by dating the model and including the *current external influences* on the same page.

The process of physically creating the logic model diagram, whatever tool is used, helps focus and organize. PowerPoint is frequently used, even though it is not as automatic as some of the newer tools. Every box and arrow in a model has to be copied from an earlier model or created by selecting a rectangular shape and creating a box of appropriate size. Ideally, the text within the rectangle, again entered with a text box, will display in at least a 10-point font. Other options are just to use a text box with black line selected or a rectangle filled with the allowed text, but these options make it harder to have boxes of a consistent size. Keep words to a minimum within boxes, and group activities into a small number so everything fits on one page. A draft may employ a smaller font and more numerous boxes until thinking is streamlined. Detail can also be moved to separate logic models that look in more depth at one or more aspects of the higher-level logic. Other than with symbols or words, there is no easy way to link models together in a nested fashion in PowerPoint. (Box 3.6 provides tips on tools for developing a logic model diagram.)

Box 3.6. Tools for Drawing Logic Models

- PowerPoint is simple to use and it has the advantage that workgroup members can take ownership of the diagrams.
- Inspiration is inexpensive, easy-to-learn, mind-mapping software that automatically generates arrows and new boxes and also has a feature through which each box can be linked to a nested, expanded logic.
- Flow Charting 6 is similar to Inspiration. These more sophisticated diagrams are easily exported into word processing software such as Microsoft Word. The downside is that to modify the diagram a person would have to have access to or purchase the modeling software.
- More sophisticated drawing tools such as SmartDraw are also available and have features beyond those already discussed here, including the ability to draw very large models that can hang on a wall, but there is a steep learning curve.
- *Easy Outcomes*, available from OutcomesModels.org [www.outcomesmodels.org] is a user-friendly and free version of a more formal approach (systematic outcomes analysis). It can be implemented with DoView software which can be used to build linked models that project well for group work.

Stage 5: Verifying the Program Logic Model with Stakeholders

As the logic model process unfolds, the workgroup responsible for producing the model should continuously evaluate it with respect to the overall goal of representing the program logic—how the program works under what conditions to achieve its short-term, intermediate, and long-term aims. The verification process should engage the appropriate stakeholders in the review process. The workgroup will use the logic model diagram and the supporting table and text. During this time, the workgroup also can address the information needed about performance, setting the stage for performance measurement and evaluation plans.

In addition to why and how questions, we recommend that three measurement and evaluation design questions be addressed in the final verification process:

- Is the program logic theoretically sound? Do all the elements fit together logically? Are there other plausible pathways to achieving the program outcomes?
- Have all the key external contextual factors been identified and their potential influences described?
- Does the program logic suggest the performance measures and major evaluation questions that must be covered?

In our experience, the exercise of defining performance measures makes a draft logic model more concrete to staff and stakeholders and often uncovers elements or relationships that have been missed. Having a logic model in place at the beginning of performance measurement and evaluation is important because it serves as an advance organizer or focusing mechanism for the measurement of key variables and for the evaluation of assumed causal relationships. As noted elsewhere in this volume, performance measurement describes *levels of performance* in relation to some standard and is typically a univariate measure, whereas program evaluation enables the explanation of why certain levels of performance were observed and is thus multivariate, using a number of performance measures to support the explanation. Logic modeling enables the identification of useful performance measures and sets up a pattern for putting them together to test underlying assumptions.

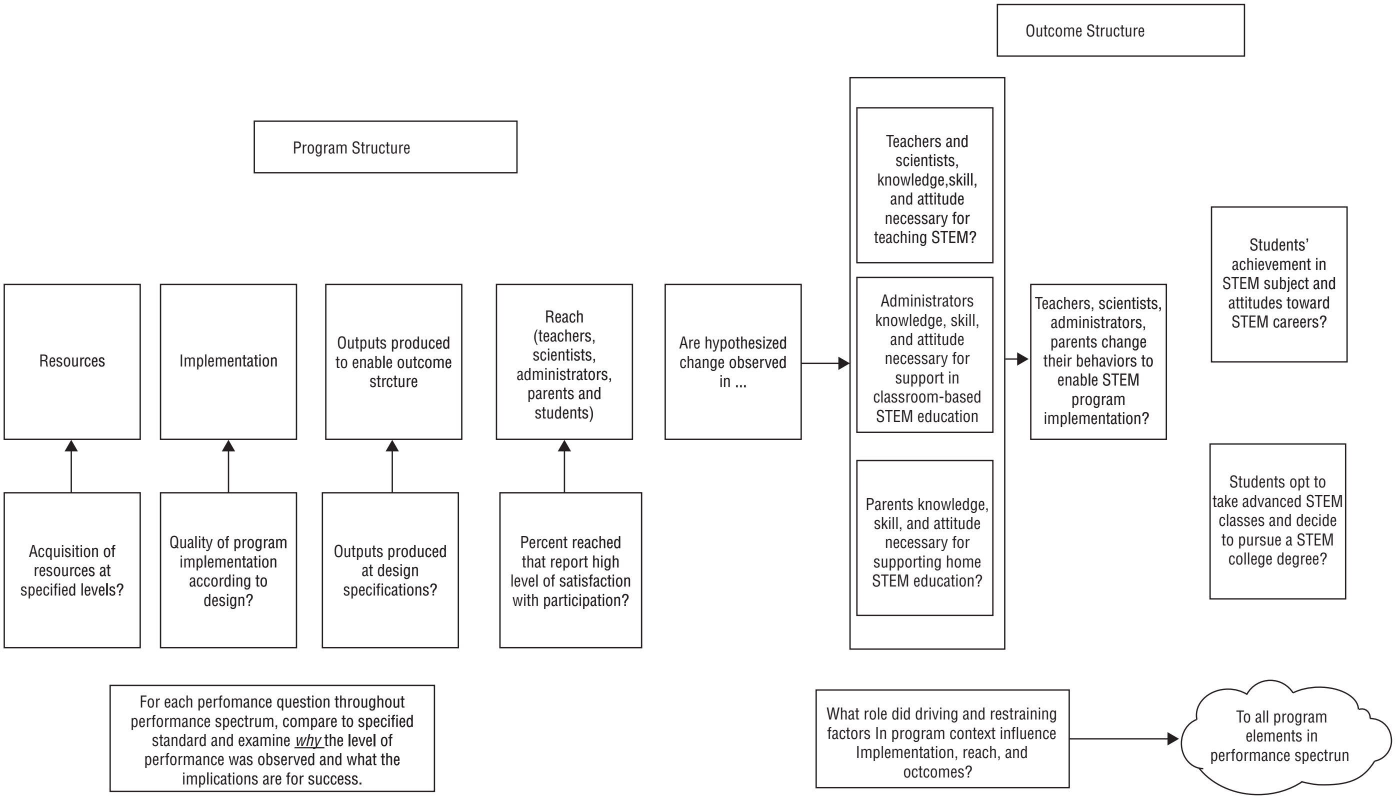
Evaluation should examine or test the underlying assumptions about how the program works to achieve intended outcomes. Weiss (1997), citing her earlier work, noted the importance of not only capturing the program process but also collecting information on the hypothesized linkages. According to Weiss, measurement should "track the steps of the program." Cooksy, Gill, and Kelly (2001) show how logic modeling can be used to focus data collection, organize data for analysis, and guide the interpretation of findings. In the logical model, boxes are the steps that can simply be counted or monitored, and the lines connecting the boxes are the hypothesized linkages or causal relationships that require in-depth study to determine and explain what happened. It is the testing of the linkages, the arrows in the logic chart, that allows the evaluator to determine whether the program is working. Monitoring the degree to which elements are in place, even the intended and unintended outcomes, will not explain the measurements or tell the evaluator if the program is working. It is essential to test the program hypotheses through impact evaluation. Even if the evaluator observes that intended outcomes were achieved, the following question must be asked: What features, if any, of the program contributed to the achievement of intended and unintended outcomes?

Special note should be made here about the relationship of the Z model to performance measurement. As noted previously and presented in Figure 3.3, the Z model describes the elements of the program's system and their interdependence. If one link in the system fails, then the end goal will not be achieved. Therefore, when the evaluator encounters a program in which there are interlocking parts, we recommend that linkage measures be developed to assess the degree to which the dependencies are operating effectively, that is, the degree to which the parts of the Z fit together functionally. In this way the Z model serves as a focusing mechanism for performance measurement and evaluation.

An example of linking performance measurement and evaluation to program logic is set forth in Figure 3.4. The linkage is developed after stage 5 in the logic model development process. The program described in Figure 3.4 is designed to increase middle school students' interest in selecting a career in science, technology, engineering, or mathematics (STEM). The underlying assumptions for the program were that if scientists and teachers collaborated in the delivery of eighth-grade science materials using problem-based challenges in cooperative learning groups and if they talked to students about how such exercises mirrored what scientists do in real life, then students would develop a more positive attitude toward school and a science career, resulting in their taking advanced math classes in high school and pursuing a science career.

Note that in this model, questions are directed at the program structure as well as the outcome structure. Further, questions are aimed at specific elements of the logic model in the program and outcome structures. The diagram also reveals that the evaluation should address potential contextual factors that might influence either the program implementation fidelity or the expected

FIGURE 3.4. SAMPLE LOGIC MODEL WITH PERFORMANCE MEASUREMENT AND EVALUATION FOCUS: MIDDLE SCHOOL STEM PROGRAM.



outcomes. This logic model was used to gain agreement among project staff and other key stakeholders about what would be studied throughout the program implementation, and then used this information for analyses. Remember, the first question seeks to assess levels of performance with respect to a particular program or outcome element. Next, the evaluator asks *why* a particular level of program performance was observed. The answer to the why question comes from a multivariate examination using program performance *and* context measures. Without answers to the why questions the program staff cannot make accurate decisions regarding necessary mid-course corrections in the program to enhance success. Further, the why question results enable staff and managers to communicate more confidently with external audiences regarding program performance. Having the logic model helps focus performance measurement and evaluation strategies and the use of findings.

A logic model is dynamic, so even after the evaluator, manager, and staff agree on the logic and the evaluation questions and data collection strategies are developed the process of verification continues. The logic model is a draft document that captures the program staff's or other stakeholders' concept of how the program works at a point in time. In fact, the program may not work that way at all. Thus the evaluator needs to test the logic model, developing what Patton (2008) has called the *theory in practice*. If discrepancies are found, the evaluator, manager, and program staff should discuss the ramifications of the discrepancies and either redesign the program or increase implementation fidelity to enhance the chance for success.

Stufflebeam (2001) noted that for many programs it will be very difficult to establish a defensible theory of change either because existing social science research has not produced sufficient evidence to support theory development or there is insufficient time to develop the theory. He argues, as we do, that logic modeling is appropriate as long as not too much time is taken for it and as long as the evaluator understands that the model is a draft that needs to be assessed in reality.

Yin (1989) discusses the importance of pattern matching as a tool to study the delivery and impact of a program. The use of the logic model process results in a pattern that can be used in this way. It thus becomes a tool to assess program implementation and program impacts. An iterative procedure may be applied that, first, determines the theory in practice and then moves on to either revision of the espoused theory or tightening of the implementation of the espoused theory. Next, the revised pattern can be used to address the extent to which the program yields desired outcomes and impacts and also the influence of context.

Conclusion

Program managers across the public and nonprofit sectors are being asked to describe and evaluate their programs in new ways. People want managers to present a logical argument for how and why a particular program is addressing a specific problem and how measurement and evaluation will assess and improve program effectiveness. Managers do not typically have clear and logically consistent methods to help them with this task, but evaluators do, and they can bring this tool to managers and help them meet the new challenges.

This chapter has presented the logic modeling process in enough detail that both evaluators and program managers and staff can use it to develop and tell the performance story for their program. The logic model describes the logical linkages among program resources, activities, outputs for customers reached, and short-term, intermediate, and long-term outcomes. Once this model of expected performance is produced, critical monitoring and evaluation areas can be identified. Because the logic model and the measurement plan have been developed with the program stakeholders, the story these tools tell should be a shared vision with a clear and shared expectation of success. Last, we must reiterate that we believe that while it is important for the evaluation to address the question of program results, it is equally important to focus the evaluation efforts on program implementation and early outcomes so the managers and staff know what's working and not working and where necessary make informed mid-course corrections to enhance the probability of longer-term success. The logic modeling process makes this possible.

References

Bickman, L. "The Functions of Program Theory." In L. Bickman (ed.), *Using Program Theory in Evaluation.* New Directions for Evaluation, no. 33. San Francisco: Jossey-Bass, 1987.

Centers for Disease Control and Prevention. *Logic Model.* www.cdc.gov/nccdphp/dnpao/hwi/programdesign/logic_model.htm, 2010.

Cooksy, L. J., Gill, O., and Kelly, P. A. "The Program Logic Model as an Integrative Framework for a Multimethod Evaluation." *Evaluation and Program Planning*, 2001, *24*, 119–128.

Gugiu, P. C., and Rodriguez-Campos, L. "Semi-Structured Interview Protocol for Constructing Logic Models." *Evaluation and Program Planning*, 2007, *30*(4), 339–350.

Frechtling, J. *Logic Modeling Methods in Program Evaluation.* San Francisco: Jossey-Bass, 2007.

Funnell, S. "Developing and Using a Program Theory Matrix for Program Evaluation and Performance Monitoring." In P. Rogers, T. Hacsi, A. Petrosino, and. T. Huebner (eds.), *Program Theory in Evaluation: Challenges and Opportunities.* New Directions for Evaluation, no. 87. San Francisco: Jossey-Bass, 2000.

Funnell, Sue C., and Patricia J. Rogers. *Purposeful Program Theory: Effective Use of Theories of Change and Logic Models.* Vol. 31. Hoboken, NJ: John Wiley & Sons, 2011.

Harrell, A., Burt, M. R., Hatry, H. P., Rossman, S. B., and Roth, J. A. *Evaluation Strategies for Human Services Programs: A Guide for Policymakers and Providers.* Washington, DC: Urban Institute, 1996.

Knowlton, L. W., and Phillips, C. C. *The Logic Model Guidebook: Better Strategies for Great Results.* Thousand Oaks, CA: Sage, 2008, and Second Edition 2013.

Leeuw, F. L. "Restructuring Program Theories: Methods Available and Problems to Be Solved." *American Journal of Evaluation*, 2003, *24*(1), 5–20.

Mayne, J., and Stern, E. *Impact Evaluation of Natural Resource Management Research Programs: A Broader View.* ACIAR Impact Assessment Series Report. 2013, aciar.gov.au/files/ias84.pdf.

McLaughlin, J. A., and Jordan, G. B. "Logic Models: A Tool for Telling Your Performance Story." *Evaluation and Program Planning*, 1999, *22*(1), 65–72.

McLaughlin, J. A., and Jordan, G. B. "Logic Models." In J. S. Wholey, H. P. Hatry, and K. E. Newcomer (eds.), *Handbook of Practical Program Evaluation* (2nd ed.). San Francisco: Jossey-Bass, 2004.

Patton, M. Q. *Utilization-Focused Evaluation* (4th ed.). Thousand Oaks, CA: Sage, 2008.

Stufflebeam, D. L. "Evaluation Models." In D. L. Stufflebeam (ed.), *Evaluation Models.* New Directions for Evaluation, no. 89. San Francisco: Jossey-Bass, 2001.

Taylor-Powell, E., and Henert, E. *Developing a Logic Model: Teaching and Training Guide.* www.uwex.edu/ces/pdande/evaluation/pdf/lmguidecomplete.pdf. 2008.

W. K. Kellogg Foundation. *Logic Model Development Guide.* www.wkkf.org/resource-directory/resource/2006/02/wk-kellogg-foundation-logic-model-development-guide, 2005.

Weiss, C. "Theory-Based Evaluation: Past, Present, and Future." In D. Rog and D. Founier (eds.), *Progress and Future Directions in Evaluation: Perspectives on Theory, Practice, and Methods.* New Directions for Evaluation, no. 76. San Francisco: Jossey-Bass, 1997.

Wholey, J. S. "Evaluability Assessment: Developing Program Theory." In L. Bickman (ed.), *Using Program Theory in Evaluation.* New Directions for Evaluation, no. 33. San Francisco: Jossey-Bass, 1987.

Yin, R. K. *Case Study Research: Design and Methods.* Thousand Oaks, CA: Sage, 1989.