

CHAPTER 3

Creating Program Logic Models

This chapter identifies the basic elements of a program logic model. Generally, these models have enough detail to support design, planning, management, or evaluation. This chapter describes a program logic model example and the action steps to create a model with a small group.

Learner Objectives

- Describe the relationship between theory of change and program logic models
- Identify basic elements for a program logic model
- Create a simple model
- Recognize limitations of display

From Theory of Change to Program Models

Theory of change logic models are literally the foundation for program logic models. When well developed, they can ensure intellectual rigor for program logic models. Figure 3.1 illustrates the relationship of a theory of change model (composed of strategies and results) to the primary elements of a program logic model.

Strategies reflect the resources, activities, and outputs needed to achieve results. Results reflect the sequence of outcomes over time through impact. Outcomes (for individuals) are generally progress in changes in awareness, knowledge, skill, or behavior among targeted audiences. There are also outcomes for organizations and systems. Although a plausible and evidence-based connection can be established,

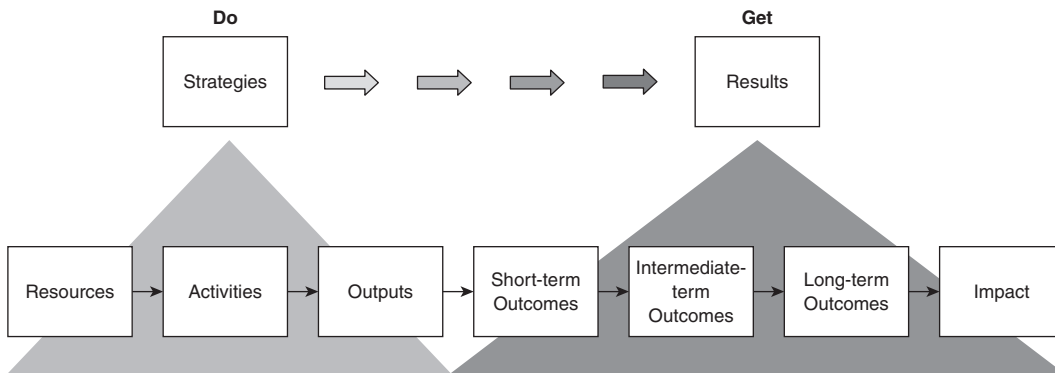


Figure 3.1 Relationship of Program and Theory of Change Models

impact is often well beyond the scope (or feasibility) for the program being modeled. Together, outcomes, which are closer to the effort, of multiple strategies plus impact (further away) make up results. While program logic models are often built on a theory of change, it is also possible to infer a theory of change from a program logic model.

Assumptions Matter

It is important to be aware that specific assumptions are not illustrated in Figure 3.1. Recall that assumptions are informed by beliefs and knowledge. Too often, program models are built without the benefit of explicitly naming the assumptions and underlying theory of change. This omission can help explain why tremendous conflict, even chaos can erupt during program development, planning, implementation, or assessment. In the absence of explicitly named assumptions, either a clear theory of change does not exist or people hold multiple and conflicting variations that reflect their deeply held views about what should/could work and why. This can lead to diffuse or dilute programs that lack the focus and intensity needed to produce intended results. Because of these implications, omitting this “foundation” for your idea, program, or social change effort undermines its potential for success.

As noted previously, conceptualization and learning styles differ from person to person. Organizational culture can also affect how design, planning, monitoring, and measuring occur. Given these practical issues, we strongly suggest that both theory of change and program logic models are eventually created to form the foundation of shared meaning for all aspects of the program. The sequence in which they are developed certainly should and will reflect the stakeholders’ preferences.

Key Elements of Program Logic Models

Program logic models display what an existing idea, new program, or focused change effort might contain from start to finish. The elements in a program logic

model consist of the recipe for a bounded investment of financial and social capital for a specified result. The level of detail increases so that the relationships shown by the model illustrate essential linkages needed to make a plan fully operational for each of the strategy strands identified in the theory of change. The primary elements for each strand of a program logic model include resources, activities, outputs, outcomes, and impact. Figure 3.2 is a template of the elements for most program logic models.

These program logic model elements are defined as follows:

Resources are essential for activities to occur. They can include human, financial, organizational, community, or systems resources in any combination. They are used to accomplish named activities. Sometimes resources are called inputs.

Activities are the specific actions that make up the program. They reflect tools, processes, events, technology, and other devices that are intentional in the program. Activities are synonymous with interventions deployed to secure the desired changes or results.

Outputs are what specific activities will produce or create. They can include descriptions of types, levels, and audiences or targets delivered by the program. Outputs are often quantified and qualified in some way. They simply characterize the application of activities with selected audiences.

Outcomes are about changes, often in program participants or organizations, as a result of the program. They often include specific changes in awareness, knowledge, skill, and behavior. Outcomes are dependent on preceding resources, activities, and outputs. Sometimes outcomes are parsed by time increments into short, intermediate, and long term. Time spans for outcomes are relative and should be specified for the idea or project described. However, short term is often 1 through 3 years, intermediate-term outcomes 4 through 6 years. Long-term outcomes might be achieved in 7 through 10 years. The intervals specified for any given model would depend on the size and scope of the effort.

For example, a small-scale project such as an adult education typing class in one location might produce knowledge and skill outcomes in 6 weeks, where behavior changes such as use or changes in employment might take somewhat longer. Alternatively, a program targeting changes in global water quality might specify changes in the awareness and knowledge of international policymakers within 1 to 3 years; actual environmental improvements might not occur within decades. Typically, dividing the project duration into thirds works pretty well as a starting point. Relying on a literature or other evidence base can help inform what is feasible.

Being clear about timing and expected results is important. The time span for outcomes is project specific. Time is one of several important considerations. The

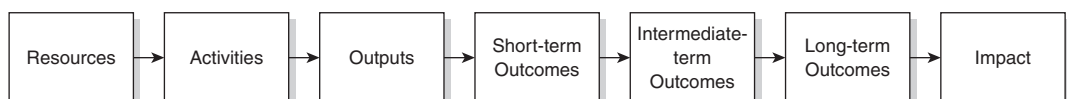


Figure 3.2 A Basic Program Logic Model

logical sequencing of any given outcome chain matters too. Think about what will happen first, then what is likely to happen next. Also keep in mind that the sequence may or may not be lockstep. Under some conditions, there may be different points of entry into a sequence. The important thing is to explore the interconnections and dependencies that do exist among the outcomes and impact you specify.

Impact is the ultimate intended change in an organization, community, or other system. It carries an implication about time. It varies in its relative timing to the actual program or change effort. Sometimes impact occurs at the end of the program, but more frequently, the impact sought is much more distant. For some efforts, this may mean impact can be cited in 7 through 10 years or more. This can have important implications as it is well beyond the funding cycle for many typical grant-funded programs or the patience of many managers or politicians. The logic model is one way to show how the work you can do within these constraints may contribute to a larger, grander impact.

The “planned work” of a program logic model includes resources, activities, and outputs. These are the essential elements that are used to secure results or make change happen. The “intended results” include what the program produces: outcomes and impact.

Nonlinear Program Logic Models

Just as in theory of change models, very few ideas, programs, or projects actually occur in a linear progression. Purposely, to aid learning, we simplified the display of elements as a straight sequence. Reality suggests cycles, iterations (additional attempts), and interactions are very common. This more organic development is shown in Figure 3.3.

In this circular display, there is no specific starting point. Although the logic model elements are constant, the work of design, planning, managing, or evaluating might begin with any element. In addition, this view shows how cycles of the same activity might occur over time. Keep in mind, the illustration groups activities together. A more detailed view could be staggering to portray. Sometimes, capturing reality in a display impedes communication.

Hidden Assumptions and Dose

A program logic model displays the elements that are most critical to establishing and operating a program or social change effort. It specifies the activities and their often interdependent relationship as well as what they are expected to generate. Program logic models do not necessarily include assumptions, but they rely on them. They offer a view of the map that can inform action planning and, later, implementation. Program logic models can also define the “dose” (e.g., number, type, and duration of activities), quantify and describe the effects and benefits of the program for a given dose and the ultimate change expected. Dose is an important concept in effectiveness. A dilute dose can have the same impact as none at all. For example, if your intended result is a large voter turnout in an election, a classified ad

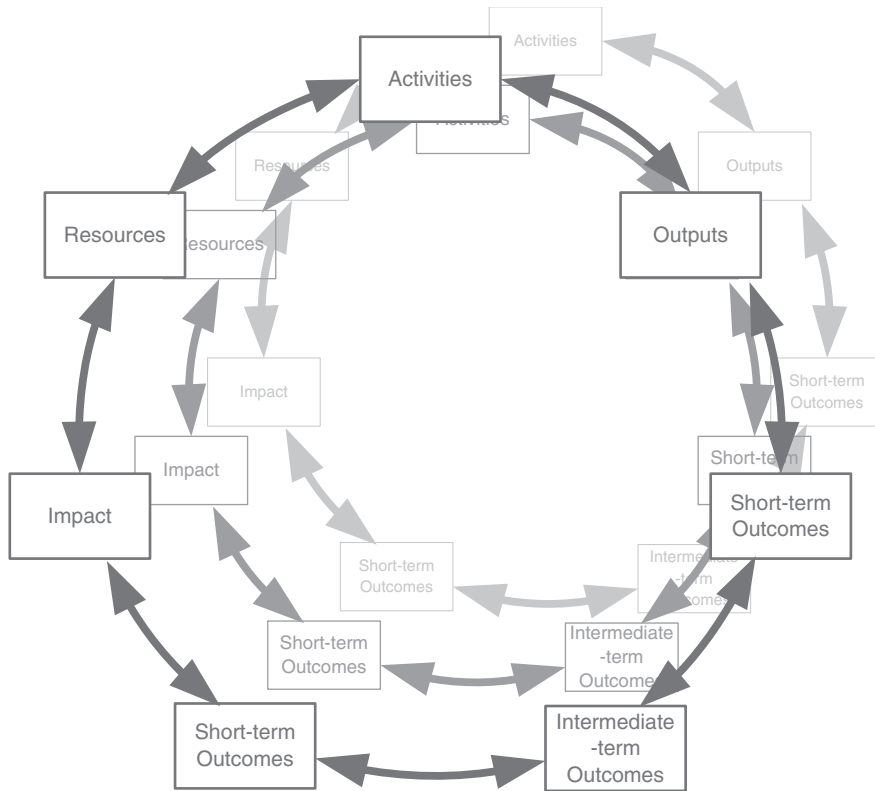


Figure 3.3 Nonlinear Logic Model

is not an adequate communication strategy. A comprehensive media plan coupled with free transportation to the voting booths has greater chances of success. So, it is important to design a program with enough of the right activities to secure the outcome you intend.

Building a Program Logic Model

Program Logic Model Example

An example of a simple program logic model for securing “improved health” is displayed in Figure 3.4. Read from left to right, this program model suggests that if we recruit and retain participants and provide exercise, nutrition, and stress reduction, then we will secure improved health. Note the development of detail connecting strategies to results in this model compared to the theory of change (see Figure 2.3). The program logic model provides detail for the theory of change by explicating the elements from a basic logic model for each strategy strand. In a program model, the details relative to resources, activities, and other elements are named.

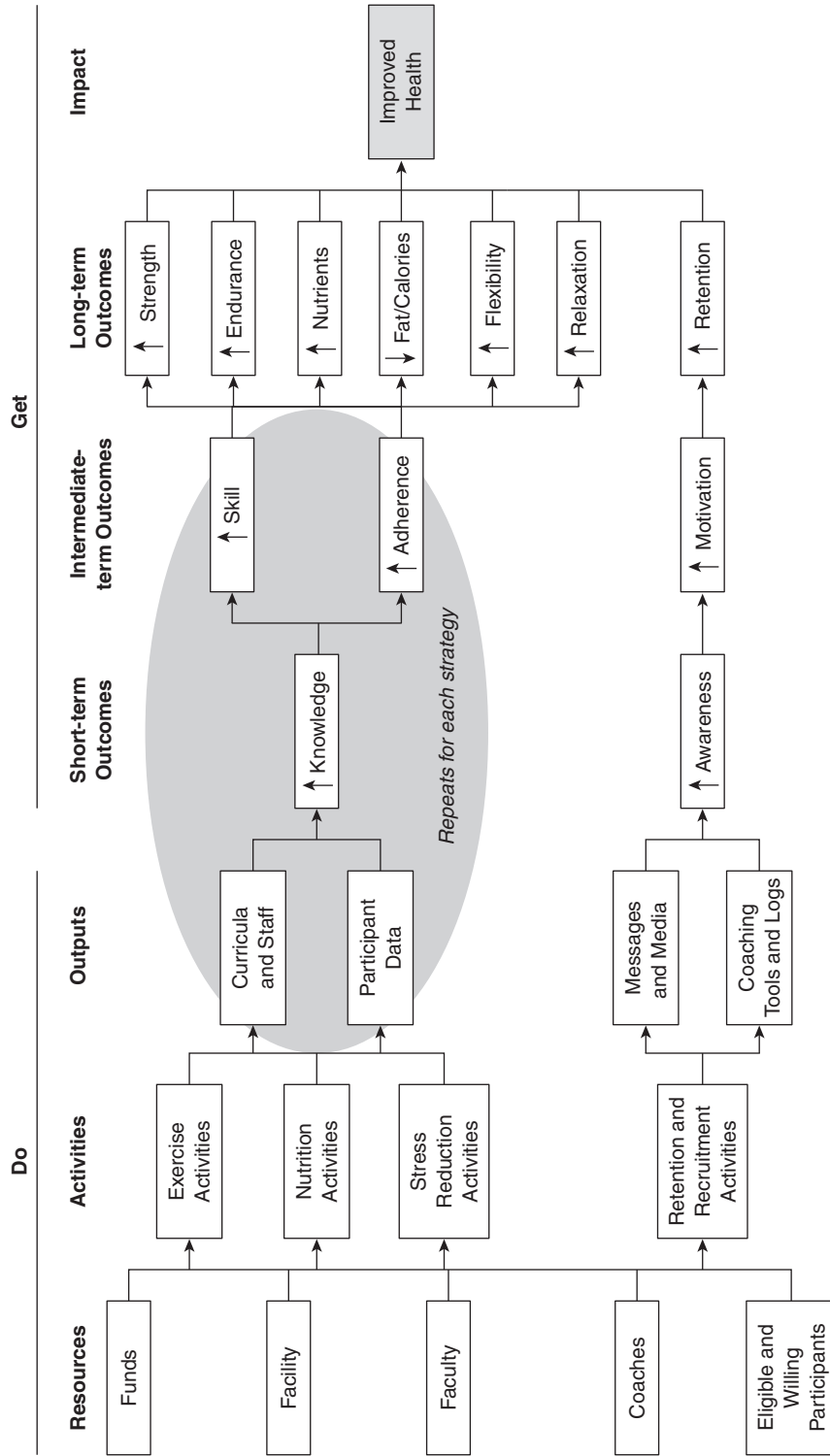


Figure 3.4 Improved Health Program Logic Model

Although still an overview and incomplete, this illustration provides a more detailed view of what this health improvement program wants to do, plans to measure, and hopes to achieve. Beginning on the left with resources, this model includes funds, facility, faculty, coaches, as well as eligible and willing participants among its requisite inputs. To keep it simple, the strategies contain implied clusters of activities in this illustration. The specific activities that contribute to outputs are not named. Outputs from the intervention strategies and associated activities (exercise, nutrition, stress reduction) could be numerous. For this illustration, we show only the overarching categories of information that could be considered. Each category would be repeated for each of the strands. These would include details about the scope, sequence, and quality of the curriculum; staffing qualifications; and information about participants and their participation. Activities “inside” these strategy strands contribute to changes in knowledge, skill, and adherence. Eventually, they can contribute to increases in strength, endurance, nutrients, flexibility, and relaxation. Concurrently, over time, these same strategies also yield reduced fat/calories. The retention and recruitment strategy strand also generates some outputs and outcomes. Aggregated, activities within this strategy secure and keep participants in the program. Note that this model uses arrows to show relationships. Sometimes they reflect a cluster (indicating synergies) rather than just one:one relationships.

As is typical of many programs, several strategies may be shown as contributing collectively to outcomes, rather than each strategy making its individual contribution to distinct outcomes in isolation. Collectively, the long-term outcomes generate improved health, which could be measured in a variety of ways (e.g., blood pressure, blood lipid and sugar profiles, weight).

In contrast to the big picture view that theory of change models offer, program logic models provide a closer, more detailed picture of operations. This view of the program provides adequate detail to then create work plans. Program models can provide a reliable outline for work plans that are used to implement and manage a program or larger change effort. Just like theory of change models, program models are often logical—but here, feasibility, given limited time and resources, is the appropriate standard for assessing their value. A common question about program logic models focuses on their level of detail. Essentially, the level of detail in program logic models should be determined by their intended use and users. Although somewhat situational, program logic models build out strategies to activities. Sometimes they can even get to the fine detail of tasks, although more often that is described in an operations or action plan.

From Strategy to Activities

Some program logic models can be extremely complex, but the steps to create them are generally the same as for more simple efforts (see Figure 3.2). Large-scale programs or multi-year change efforts (sometimes called “initiatives”) often are composed of many strategies aimed at target audiences across many sites over considerable time.

Program logic models usually do not display underlying beliefs or assumptions. They are nevertheless important elements in the conscious exploration of multiple target audiences. Sometimes programs or change efforts are implemented in a cascade with some overlap in time, which requires a particular sequence of strategies and associated activities. When this is the circumstance, it can be helpful to focus on a function, a given strategy, or one partner’s designated work. The task is often simplified by thinking about a single aspect and then connecting it back to the whole with some of the inherent complexity reduced. Ultimately, program execution relies on integrated action—but the work that precedes it may require focused developmental attention on smaller parts.

Using the health improvement program example, Figure 3.5 provides an orientation to how the exercise strategy strand might be reduced to activities. It breaks the strategy into greater detail for the purposes of selection and design.

In Figure 3.5, it becomes evident that exercise as a strategy is made up of several key activities. They include physical exercise (strength and endurance), education, and assessment. Together, all of these activities represent a comprehensive strategy, exercise, that is just one means to improved health. Recall that the whole theory of change for this example also includes stress reduction, nutrition, and retention. It is the combination of strategies reflected in the whole program that is most likely to secure results. Each strand of a comprehensive program logic model needs to illustrate the contribution of each strategy as well as the interdependence.

As you specify the activities content of your strategy you are naming more precisely what makes up the given strategy. Later, the whole model is tested for feasibility—both practically before implementation and literally when the program is evaluated.

In Figure 3.6, we provide a view with greater detail for *only* the exercise strategy.

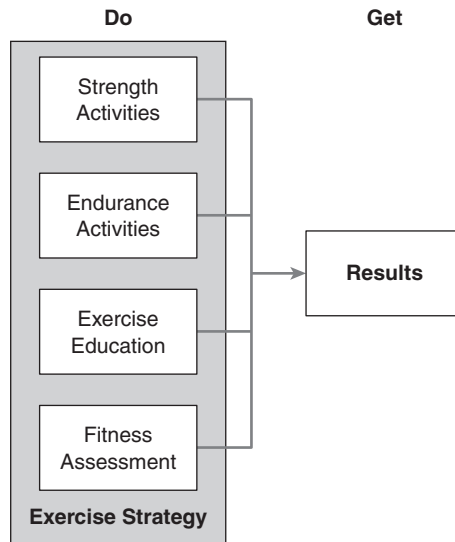


Figure 3.5 One Strategy With Multiple Activities

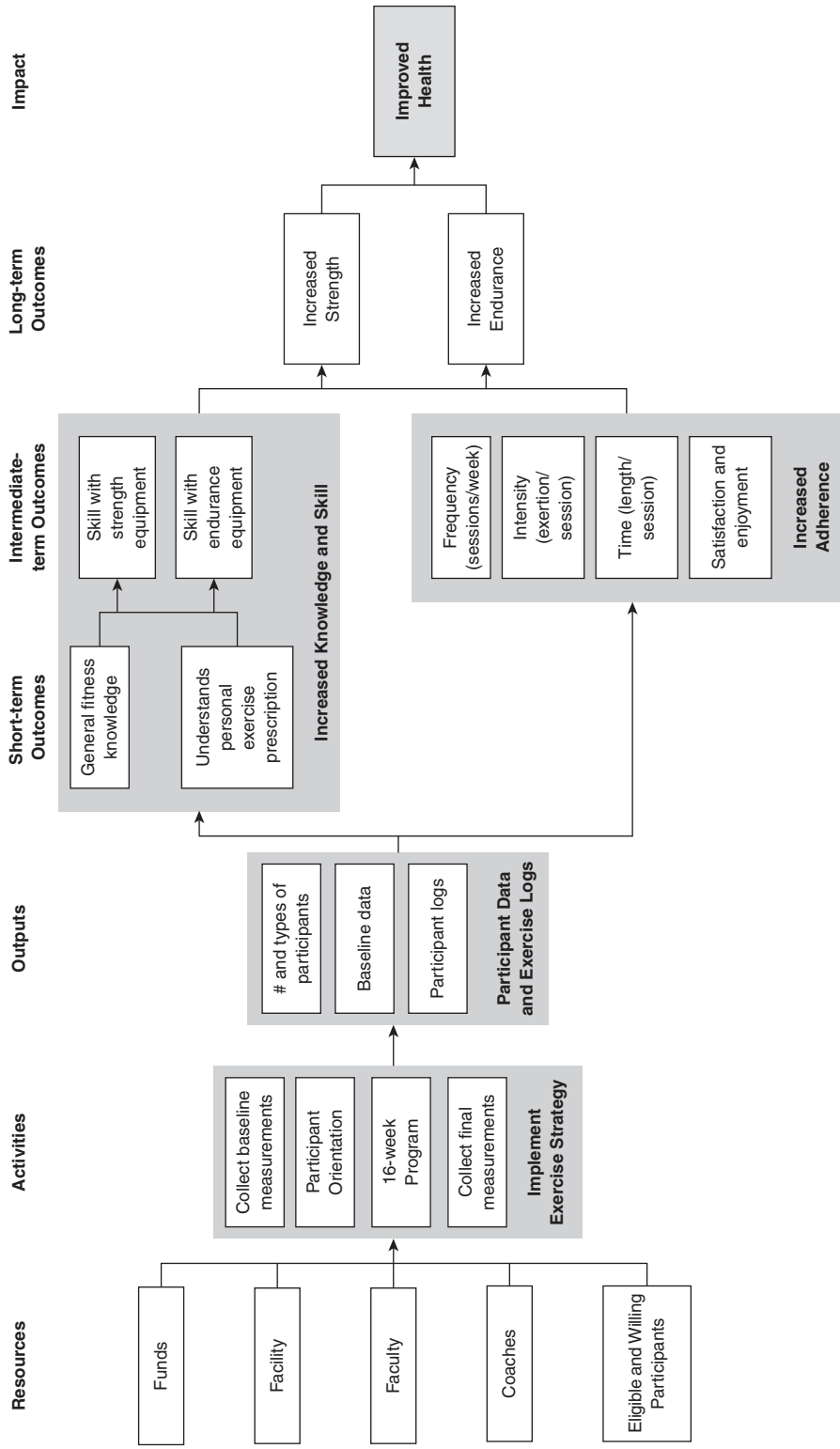


Figure 3.6 Exercise Strategy Detail

In this illustration, we show the detail of activities within the exercise strategy. It also suggests the many decisions hidden in program design and planning. In choosing activities, it is critical that the relationship among strategies and activities is intentional. The strategies and the cluster of appropriate activities should also be chosen with reference to a target audience. Remember, logic models use if-then sequences from left to right in the columns and among the features as you read from left to right.

Action Steps for a Program Logic Model

The practical construction of a program logic model often begins with one or more information sources (e.g., research, interviews, documents). We recommend that people begin both theory of change and program logic models with the named ends. People are most clear about their intended results (outcomes and impact). Our experience is that you do *know* what you want to accomplish. The results sought reflect both the impact intended and the outcomes over time. Next, name the changes or outcomes that will be part of your progress toward impact. Unpacking this sequence is important because it makes it easier to see the strength of the connection between what you do and what you can get. We suggest tackling the activities required to achieve the outcomes you have specified in your third step. Activities are about *how* intended changes will occur. Fourth, resources/inputs become the essential ingredients of activities. And finally, outputs reflect the information needed to verify that activities named earlier in the process reach the right audiences and are of the quality and quantity needed to produce results. So, the steps to draft a program logic model are ordered in this way:

1. Identify the results that one or more strategies will ultimately generate.
2. Describe the stepwise series of outcomes (or changes) that will show progress toward impact.
3. Name all the activities needed to generate the outcomes (for each strategy).
4. Define the resources/inputs that link directly to and will “supply” the activities.
5. Identify the outputs that reflect the accomplishment of activities.

Figure 3.7 illustrates these action steps and their sequence.

Creating Your Program Logic Model

The format of a logic model format helps organize information in a useful way. Think of an idea, project, or program you manage now or want to create and its results. For each strategy, brainstorm elements that might be cited in short-term outcomes first but are clearly linked to your intended results. Do the same for resources, activities, and outputs. It is important to make choices about the outcomes that are feasible with your limited resources. This is discussed in greater detail in the next chapter.

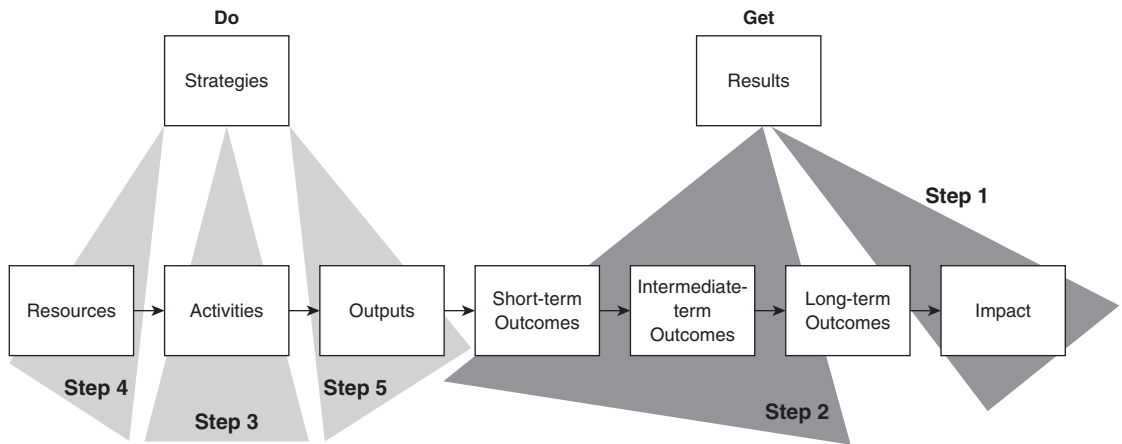


Figure 3.7 Steps in Creating a Program Logic Model

With some experience, you will begin to recognize commonly used strategies that reflect knowledge from your field or discipline. For example, marketing/communication, recruitment, retention, professional development or education, advocacy, and policy are strategies often found in program models. Examples of activities under a marketing/communication strategy could include prepare database of target markets, generate news releases, create and send a newsletter, generate a Web site, and prepare public service announcements. We suggest you tackle one strategy at a time. Aim to define the same level of detail for each strategy. Selected examples of archetypes, or tested recipes, are described in Chapter 7.

Creating a Program Logic Model Together

We think the best method for generating a program logic model engages a small group, especially if the members are stakeholders in the idea or program it represents. Stakeholders are situational, but generally are those with an interest in or people likely to benefit from the program. Logic modeling often includes funders, program staff, and program participants. Intentionally including stakeholders supports best contributions as well as some subsequent benefits relative to implementation. The facilitation of modeling requires some advance planning and a commitment to both discipline and quality during the process. A determination about the status of a shared understanding and meaning for specified results gets your effort started. It is important to note that models may need to be updated to respond to the dynamics of an external environment (context). They also reflect living systems that are not mechanistic but changing. For these two reasons (and others), it is necessary to expect models will be revised. With limits on time, impact and outcomes can be identified and selected. This can be accomplished a number of ways.

We have had success in using the action steps noted, particularly when each participant contributed to the model elements via small sticky notes. This quickly

generates a large number of possibilities for each element. Redundancies should be noted and celebrated as commonly held. Then, the group can sort them: those that *must* be kept, that *could* be kept, and those that are *not relevant*. Once the results are named, then it is relatively easy to specify the other elements. In this disciplined process each stakeholder contributes to the whole and each contribution has the benefit of an internal test relative to design.

We often use Microsoft Visio to construct our models, but many other applications such as Word and PowerPoint have drawing options. These as well as Inspiration software are all readily available. The Supplemental Readings list at the end of the chapter identifies some examples of other free and commercial software applications. Take care in using technology for model creation because it can exclude valuable participation.

In Summary

High-quality program logic models depend on the evidence base found in their parallel but simpler theory of change models. Program logic models display several important elements: resources; activities; outputs; short-, intermediate-, and long-term outcomes; and impact. To create a program logic model start with the intended results: outcomes and impact. Then, activities (which are consistent with strategies in the theory of change model) are selected. Next, resources and outputs are cited. We believe creating models with deep participation of stakeholders improves their quality and encourages their use.

Learning Resources

Reflection

1. What are the implications of a program logic model built without a specific theory of change?
2. Think of a successful business and its product or service. What is the underlying program logic that shows the explanations for profitability?
3. Feasibility relies on several aspects. Can you name some?
4. What are strengths and limitations of a linear or a nonlinear display? Would individuals from different fields (and their relevant cultures) answer similarly or differently? Why?
5. Why is being specific about results important?

Application

Specify the result of a shared program, project, or idea. Draw a theory of change model for the program, project, or idea. Then, attempt a program logic model. Using sticky notes or pieces of paper, brainstorm the outcomes that need to happen to secure the result. Organize them into short, intermediate, and long term. Pick one short-term outcome. Brainstorm what activities are critical to that outcome. Organize the activities relative to a single or multiple strategies. For given strategies and their activities, name the resources needed. From the activities, cite what outputs are possible. Organize these elements as one model.

References and Supplemental Readings

Texts

- Frechtling, J. (2007). *Logic modeling methods in program evaluation*. San Francisco: Jossey-Bass.
- Green, E. L. (2005). *Reinventing logic models: A stakeholder-driven group approach*. Unpublished doctoral dissertation, University of Cincinnati, OH.
- Mayeske, G. W. (1994). *Life cycle program management and evaluation: An heuristic approach*. Washington, DC: United States Department of Agriculture.
- United Way of America. (1996). *Measuring program outcomes: A practical approach*. Alexandria, VA: Author.
- Westley, F., Zimmerman, B., & Patton, M. Q. (2007). *Getting to maybe: How the world is changed*. Toronto: Vintage Canada.
- Wong-Rieger, D., & David, L. (1996). *A hands-on guide to planning and evaluation*. Ottawa: Canadian Hemophilia Society.

Journal Articles

- Cooksy, L. J., Gill, P., & Kelly, P. A. (2001). The program logic model as an integrative framework for a multimethod evaluation. *Evaluation and Program Planning, 24*(2), 119–128.
- McLaughlin, J. A. (1999). Logic models: A tool for telling your program's performance story. *Evaluation and Program Planning, 22*(1), 65–72.
- Millar, A., Simeone, R. S., & Carnevale, J. T. (2001). Logic models: A systems tool for performance management. *Evaluation and Program Planning, 24*(1), 73–81.
- Rush, B., & Ogborne, A. (1991). Program logic models: Expanding their role and structures for program planning and evaluation. *Canadian Journal of Program Evaluation, 6*, 95–106.

Internet Resources

For comprehensive bibliographies and links to additional resources, see:

- Logic model resources*. (n.d.). Atlanta, GA: The Evaluation Working Group of the Centers for Disease Control and Prevention. Retrieved October 25, 2007, from <http://www.cdc.gov/eval/resources.htm#logic%20model>

Jung, B. C. (2007). *Evaluation resources on the Internet*. Retrieved October 25, 2007, from <http://www.bettycjung.net/Evaluation.htm>

For logic model development toolkits, see:

Enhancing program performance with logic models. (n.d.). Madison: University of Wisconsin Cooperative Extension Service. Retrieved October 25, 2007, from <http://www.uwex.edu/ces/lmcourse/>

Graig, E. (n.d.). *Logic models: A tutorial*. Useable Knowledge, LLC. Retrieved October 25, 2007, from [http://www.usablellc.net/Logic%20Model%20\(Online\)/Presentation_Files/index.html](http://www.usablellc.net/Logic%20Model%20(Online)/Presentation_Files/index.html)

Point K: Practical tools for planning, evaluation and action—Logic model builder. (n.d.). Washington, DC: Innovation Network. Retrieved October 25, 2007, from http://www.innonet.org/index.php?section_id=64&content_id=185

Community toolbox: Developing a theory of change or logic model. (n.d.). Lawrence: University of Kansas. Retrieved October 2007, from http://ctb.ku.edu/tools/en/section_1877.htm

For logic model development software, see:

Logic model. (n.d.). Toronto: Performancesoft, Inc. Retrieved October 25, 2007, from <http://www.performancesoft.com/solutions/logic-model/software.asp?Solution=plm>

Outcome tracking software and outcomes-based applications. (n.d.). Durham, NC: Results Technologies Solutions, Inc. Retrieved October 25, 2007, from http://resultstechnologies.net/resources/primary_attributes.php