

# OPEN Fact Sheet: Getting started with program logic

This quick guide is part of a suite of technical resources developed by OPEN on behalf of The Centre.

For more resources, go to [www.outcomes.org.au](http://www.outcomes.org.au)

# OPEN Fact Sheet: Design - Program Logic

## What is it?

The key tool in designing for outcomes is developing a logic model/map. A logic model makes clear our assumptions about how our actions will cause the outcome we are expecting. It encourages us to reflect on our intent, actions and desired result and to discuss them with others and make them explicit.

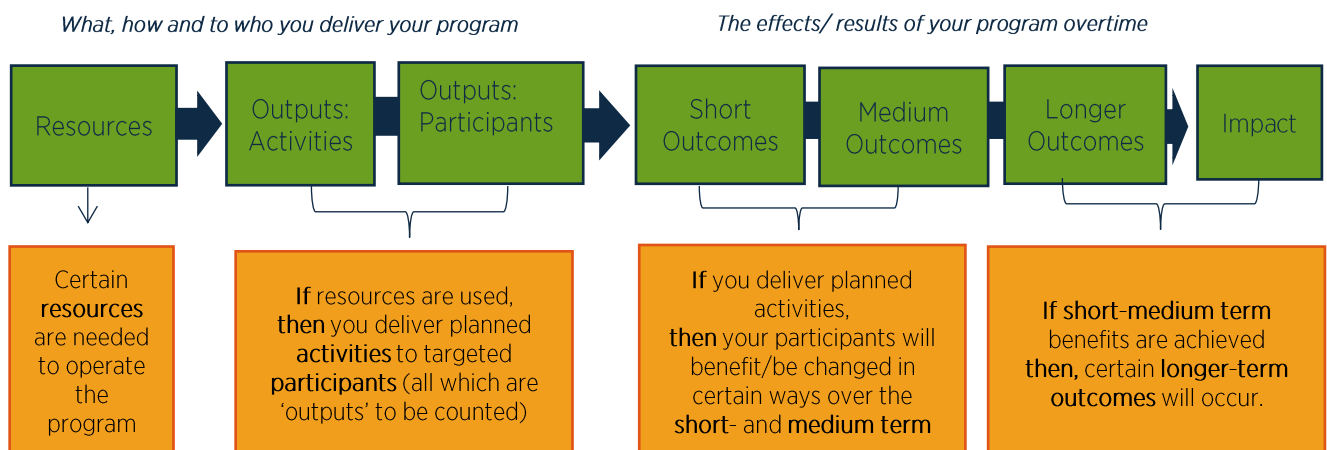
This fact sheet refers to program logic and describes what is helpful at this level. For smaller scale activities, similar key questions and processes can help but these might not be documented as formally or involve the same degree of collaboration with others.

## Logic models can be used to inform the work of the:

- Practitioner in providing a clinical response to a client
- Program manager in designing a program or strategy
- Senior managers in designing outcome frameworks and a measurement approach for a whole service area or organisation.

As shown in Quick Guide: Design Step 1, most commonly, your program logic thinking will be represented as a diagram or map which shows, in a simple form, the implied 'if - then' logic statements that link the components.

A note on terminology: Logic models are represented in a variety of ways, driven by intent, emphasis and often reflecting personal preference. Regardless of terminology or design, logic models seek to represent the causal relationships between action and effect to better unpack and understand these. A variety of terms are also used to describe these models. These include terms such as theory of change, intervention logic, outcome logic, results chain, process map. Diagrams can be left to right, bottom to top, circular or even triangular. *Tip:* Choose one that works for you and helps you organise your thinking.



## Getting started

The key is to keep your program logic as simple as possible and written in language that can be easily understood.

An effective program logic will:

- Use clear, unambiguous, specific language. Avoid jargon and vagueness.
- Be logical. Show the direction of expected changes and sequence of expected changes.
- Present coherent causal relationships showing how the program activities will contribute to outcomes and ultimately impacts.

## Developing a logic model – Fill out each component in this order:

1. Problem Statement	Identify and clarify the problem, issue, concern or need that the program is being developed to address. The ‘why’ is important. You will draw on evidence, practitioner expertise and client perspectives to build your understanding about the problem/issues and its causes. (See <a href="#">Quick Guide – Identify</a> if you need to test or broaden out your evidence.)
2. Longer term outcome or vision	Understand what success for your program looks like – what is the longer term outcome you want for individuals, families, communities, and/or the environment. This will be linked to the problem statement and restates it as an ‘end state’ you are seeking and for whom. You might find you cannot realise your vision within the available time frame but it is important to keep your longer term purpose in view.
3. Activities and Participation	<p>This relates to what you are doing, or plan to do, to contribute to the longer term outcomes defined above.</p> <p><b>Activities</b> – Focus on the key activity features, when, how, and to which targeted client group you will deliver these activities. Think about your activities in order (we do this – then this – then this) actions. Each of these will produce an output (the quantifiable product of your activities) for a specific participant group and each will result in specific outcomes over the short, medium and longer term.</p> <p><b>Delivery or implementation processes</b> – Programs will also often have specific approach to implementation informing <b>how</b> they intend to deliver activities to achieve outcomes <i>for a specific targeted group</i>. Identify if there is a specific delivery model or principle that you see as fundamental to the achievement of your program outcomes. If so, this needs to be documented as it is an important component to monitor and assess going forward.</p> <p><b>Participation</b> – This specifies who will participate in your program. It keeps your targeted client group front and centre. Make this as specific as possible. This will become important when you assess what data you need to collect from whom.</p>

<p>4. Short – medium -long term outcomes</p>	<p>There are shorter term outcomes that will happen before your longer term outcomes or impact. Community service programs that aim to change clients behaviour will probably more immediately report changes in:</p> <ul style="list-style-type: none"> <li>• Understanding, <i>then</i></li> <li>• Knowledge, <i>then</i></li> <li>• Skills, <i>then</i></li> <li>• behaviours or capabilities, and only <i>then</i></li> <li>• broader life circumstances or whole of service and/or system level changes.</li> </ul> <p>These changes may occur over weeks or months depending on the program, while longer term outcomes or impact are more commonly seen as changes for whole client groups over multiple years through persistent effort.</p>
<p>5. Assumptions</p>	<p>Assumptions focus on your beliefs about the program that you know might be a weakness or are untested. To identify assumptions, put yourself in the position of your fiercest critic.</p> <ul style="list-style-type: none"> <li>• What would a critic question, doubt, or challenge?</li> <li>• What do these concerns relate to:             <ul style="list-style-type: none"> <li>○ Delivery – Can you reach your targeted group? Can you deliver what you are planning? Do you have appropriate skills and resources?</li> <li>○ Outcomes – Does your logic map make sense? Is it plausible? Who is the program likely to work best for – Who might it not work for?</li> <li>○ Unintended consequences – What could go wrong? What are the risks?</li> </ul> </li> </ul>
<p>Making it work – causal connections</p>	<p>Finally, define and test the causal connections and order in which you expect changes to occur – building deeper thinking about how change will occur, in what order and what your contribution might be. Stay at a higher level and remember this is a model not a description of reality.</p> <p>Using post-it notes can help. Put one point on each post-it note and move things around to explore your thinking. <b>Draw links and arrows to represent relationships.</b></p>

## Quick tips to get the most out your program logic process

- **Keep it simple.** A program logic represents the bare bones of your program. Don't try to include everything (e.g. program management or administrative processes).
- **Develop your thinking with input from others.** Developing with others can help build common agreement about how your program should work. There are some great

approaches using white boards – or sticky notes on the wall which can be moved around. The process of developing the program logic, and revisiting it regularly, is often as valuable or more so than the actual logic itself. Expect that it will take a number of sessions.

- **Recognise and move on when you are getting bogged down.** While discussion and debate about terminology is part of the process, recognise when it is taking you around in circles. Agree that once these discussions are not productive, you will park the item and return to it at a later time and with fresh eyes.
- **Understand it will never be perfect, and there is no ‘right’.** Judgement is needed in preparing a logic map and knowing how to structure and describe your intent or strategic purpose. As a living document you will gain further knowledge and test and measure your progress as you move through implementation.

## To get your started, find:

Further resources are forthcoming on the topics below

- [OPEN Template: Design - Program logic template](#). This provides guidance on filling out each component of your program logic and some common pitfalls.
- [OPEN Template: Design - Brainstorming your Program logic](#). This provides a tool to use with teams or groups.

## What are some tools that could help?

Designing a Program Logic	
The Australian Institute of Family Studies practice resource <a href="https://aifs.gov.au/resources/practice-guides/getting-most-out-program-logic-models">https://aifs.gov.au/resources/practice-guides/getting-most-out-program-logic-models</a>	This resource provides detailed guidance on creating a Program Logic including a <a href="#">video</a> ‘guided tour’
Social Policy Evaluation and Research Unit, NZ: Making Sense of Evaluation – A Handbook for everyone <a href="https://thehub.swa.govt.nz/resources/making-sense-of-evaluation-a-handbook-for-everyone">https://thehub.swa.govt.nz/resources/making-sense-of-evaluation-a-handbook-for-everyone</a>	Provides a plain English explanation with everyday examples of program logic development. A great place to start.
NPC: Creating Your theory of change: NPCs practical guide <a href="https://www.thinknpc.org/resource-hub/ten-steps/">https://www.thinknpc.org/resource-hub/ten-steps/</a>	Provides a 10-step approach to developing a theory of change. Easy language and instructions.
SNAICC: Aboriginal and Torres Strait Islander child and Family Services Evaluation Readiness Toolkit <a href="https://www.snaicc.org.au/sector-development/monitoring-and-evaluation/">https://www.snaicc.org.au/sector-development/monitoring-and-evaluation/</a>	Has been designed to support Aboriginal and Torres Strait Child and Family Services. It provides an excellent and assessable explanation and step by step approach that many generalist services will value.
Dylomo: online logic model builder <a href="https://dylomo.com/">https://dylomo.com/</a>	Dylomo offers an online logic model builder that can be viewed in “static” or “dynamic” formats and updated in real time. This could be a great addition to a group workshop or development session.