8

Question #4—How Do We Get There?

Planning is bringing the future into the present so that you can do something about it now.

—Alan Lakein

Identify Action Steps

Objectives	*Success Measures*	*Verification*	*Assumptions*
Goal			
Purpose			
Outcomes			
Inputs			

FIGURE 8.1 The LogFrame Helps Organize Activities, Budgets, and Schedules

The Ancient Project that Saved the World

Do you wish you could have been there long ago when God instructed Noah to build an ark? Maybe the message came via a memo carved in stone or typed on God's personal stationery.

> I have decided to make it rain real hard for 40 days and 40 nights. Noah, I want you to build an ark big enough to hold a pair of all the animals on earth (and people) so you can survive the flood. After the flood, you can restore life on earth and ensure the long-term survival of human and animal life.
>
> Get everything necessary ready before the big rains start in six months. Build a seaworthy ark, bring a pair of each type of animal and people aboard, along with necessary supplies. Here's an advance copy of *Strategic Project Management Made Simple*. Read about and use the Logical Framework for your project plan. Don't feel pressured just because the future of civilization depends on your project management skills. Good luck.

Whew! Definitely a career-defining opportunity—if he could pull it off! Noah realized he needed some high-powered Strategic Project Management concepts and started reading his new book. He flipped to the section about chunking.

Chunking 101

"Chunking" is the art and science of creative grouping. Chunking means breaking down something BIG (problem, strategy, Goal, etc.) into smaller, more accessible "chunks" (phases, components, Outcomes, categories, aspects, etc.). The word chunk doubles as a *verb* as well as a *noun*. The verb expresses the thinking process, while the noun describes the resulting categories. For some projects, the most logical chunks will be obvious, while others will leave you furrowing your brow and wondering where to begin.

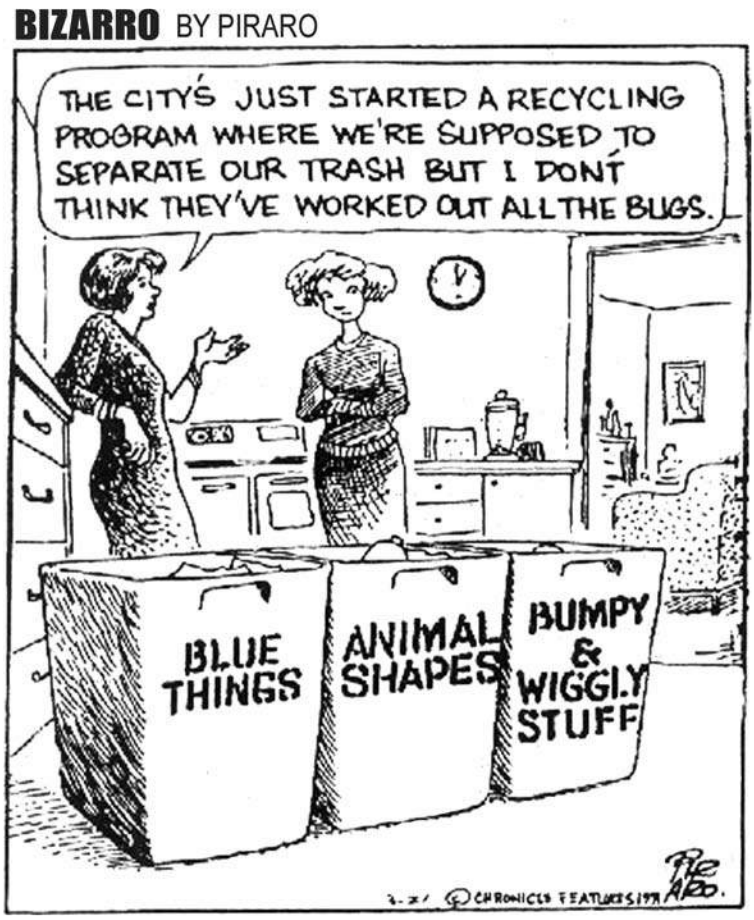

FIGURE 8.2 **Sloppy Chunking Leads to Problems**
King Features Syndicate. Reprinted by permission.

Smart chunking avoids the problems expressed in the Bizarro cartoon shown in Figure 8.2. The bin categories are not sufficiently discrete to enable someone to decide what goes in which container. Can you decide where a blue, bumpy toy elephant belongs? Do you have a clue where a cue ball goes? There are both gaps and overlaps in the bin labels. Worst of all, the categories don't relate to the higher Objectives motivating the project.

Sloppy chunking can jam up your system from the start and make your effort sputter. Smart chunking helps you to:

- Identify logical phases for a multi-phase project.
- Clarify how the diverse program and project elements relate to one another (as shown in the Objective Trees in Chapter 3).
- Define the cluster of project LogFrames that support a larger program or corporate Goal.
- Select the set of proposed Outcomes to achieve a project Purpose.
- Choose the set of Inputs needed to produce each Outcome.

The most common first-order chunking logic is by phases, but secondary chunking can take place within each phase. Use whatever chunking criteria make most sense in achieving your higher Objectives.

Decisions about project chunking begin emerging during Question #1 when you consider whether your LogFrame plan will cover the whole project or just one part or phase.

Noah Chunked Wisely

Noah wisely chunked his project into three phases—(1) Pre-Flood, (2) Flood, and (3) Post-Flood. Each phase shared a common overarching Goal, with phase-specific Purposes and a unique set of Outcomes for each.

Because he was a Strategic Project Manager (among the first ever), he began by creating three scrolls with a LogFrame for each phase.

Starting with the Pre-Flood phase, he gathered his team and tackled Question #1—What Are We Trying to Accomplish and Why? Their vertical logic may have looked like this:

Goal	Ensure long-term survival of human and animal life on earth.
Purpose	Survive the flood
Outcomes	1. Ark built 2. Ark loaded with necessary supplies 3. Animals and people collected and loaded

Noah was tempted to jump to bar charts, but his naval architect spouse Noelle reminded him to answer Questions #2 and #3 before addressing schedule, which happens during Question #4.

So they tackled Question #2—How Do We Measure Success? They began by identifying Measures and Verifications for the Purpose "Survive the flood."

Purpose	**Purpose Measures**	**Verification**
Survive the flood.	1. Ark lands with 100% of animals and humans who boarded still alive, healthy and fertile.	1.1 Review passenger manifest 1.2 Conduct health tests

Note how defining requirements at higher levels sets parameters for what is necessary at the next lower level. In this case, the Purpose Measures of health and fertility remind us that the "Animals collected" (Outcome #3) must be healthy, breedable, and willing pairs. Without being clear about the need to breed, his animal collection team could easily have considered "a pair of each animal" to mean two of each species without regarding gender, thus thwarting survival for that species. (Maybe that's how we lost the unicorns.)

Always progress top-down from Goal and Purpose Measures before setting Outcome Measures. That way you can set the Outcomes Measures at the magnitude needed to reach higher level Objectives.

After Noah's team answered the first three questions, their LogFrame looked like the one shown in Figure 8.3.

Organizing Inputs: Nitty Gritty Project Planning

By now you can appreciate that the LogFrame Input row accommodates tasks along with schedules, responsibility charts, and resource budgets. A wisely chunked activity list is the starting point for all three of these management tools.

The LogFrame Inputs are meant to offer a high level summary, not a comprehensive action plan. Consider Inputs as the jumping off point for more detailed planning, using Work Breakdown Structures (WBS), networks, and my old friend the Gantt chart. Putting too much detail into the matrix defeats the LogFrame's value as a summary document that can concisely communicate the project design to

	Objectives	Success Measures	Verification	Assumptions
	Goal: Ensure the long-term survival of human and animal life on earth.	***Measures of Goal Achievement:*** 1.1 All species propagate within their next gestation cycle and continue to multiply. 1.2 After 100 years, total # of animals is > # before the flood.	1.1 Birth rates. 1.2 Census count after 100 years.	***Assumptions to reach Goal:*** 1. No environmental catastrophes (e.g., tidal wave). 2. Plant life returns after flood. 3. Animals reproduce.
Then / *If*	***Purpose:*** Survive the flood.	***Purpose Measures:*** 1. Ark lands with 100% of animals and humans who boarded still alive, healthy and fertile.	1.1 Review passenger manifest. 1.2 Conduct health tests.	***Assumptions to achieve Purpose:*** 1. Rainfall stops in 40 days; water subsides in 20 days. 2. Severe storms do not damage ark. 3. Food and supplies sufficient. 4. Operations and maintenance systems on ark do their job.
Then / *If*	***Outcomes:*** 1. Ark built. 2. Ark loaded with necessary supplies. 3. Animals and people collected and loaded.	***Outcome Measures:*** 1. By week 20, seaworthy ark is built according to design. Ark capable of holding one pair of all the animals on earth. 2. Ark is loaded with supplies & equipment. Supplies include X lbs. of food per animal per day; medical and maintenance equipment, seeds, etc. 3. At least 1 healthy breeding pair of each species are aboard ark by week 22.	1. Physical inspection and float tests. 2. Compare against supply equipment list. 3. Health tests for all animals on checklist.	***Assumptions to produce Outcomes:*** 1. Rain will not start before six months. 2. Good weather for construction. 3. All animals on board are on checklist.

FIGURE 8.3 Logical Framework for Noah's Ark Project (after Answering the First Three Questions)

stakeholders not interested in the details. You can keep Inputs general and illustrative, or paste a reasonably accurate summary of your detailed work plan generated from the other planning tools back into your LogFrame's Input row, or cite more detailed plans.

Noah's Noisy Planning Session

A noisy discussion broke out among Noah's team about the relationship between Inputs and Outcomes. Take, for example, "Build ark." Is this an Input or an Outcome? It could be both. As grammatically stated, it's clearly an Input activity. But if you are responsible to deliver a completed ark, then it becomes an Outcome and should be restated in past-tense, fully completed language, as in "Ark built".

Let's trace the logic, starting with this If-Then relationship:

Outcome 1. Ark built

↑

Inputs 1. Build ark

The relationship "If we built ark, then the ark is built" is true, but not very useful. Utility comes when you then explode or break out the activity of "Build ark" into its component parts, like this:

Outcome	1. Ark built
↑	
Inputs	1.1 Design ark
	1.2 Hire labor
	1.3 Cut lumber
	1.4 Construct ark

Keep your Inputs at roughly the same magnitude. If the chunks are still too large, you can elevate the Inputs into Sub-Outcomes using past tense verbs. If you "slide Objectives up a level," that creates space for more detailed Inputs such as in this example:

Outcome	1. Ark built
↑	
Sub-Outcomes	1.1 Ark designed
	1.2 Labor hired
↑	1.3 Lumber cut
	1.4 Ark constructed
Inputs	1.1.1 Hire architect
	1.1.2 Develop specifications
	1.1.3 Etc.

When elevating Inputs to Sub-Outcomes, add, delete, or modify your sub-Outcomes as needed so you don't have a confusing mix as in the Bizarro cartoon. You can then identify three to five new Inputs for each Sub-Outcome, which provide finer-grain chunks suitable for building out the action plan.

If this reminds you of work breakdown structures, it's no accident. That's exactly what this process is. And if you are eagerly waiting to crank up your project management software—*Let 'er rip!* You can now lay out a coherent action plan with confidence that you are aiming at the right set of well-tuned Outcomes. Software plays an essential role in project management—but only after sound strategies are in place.

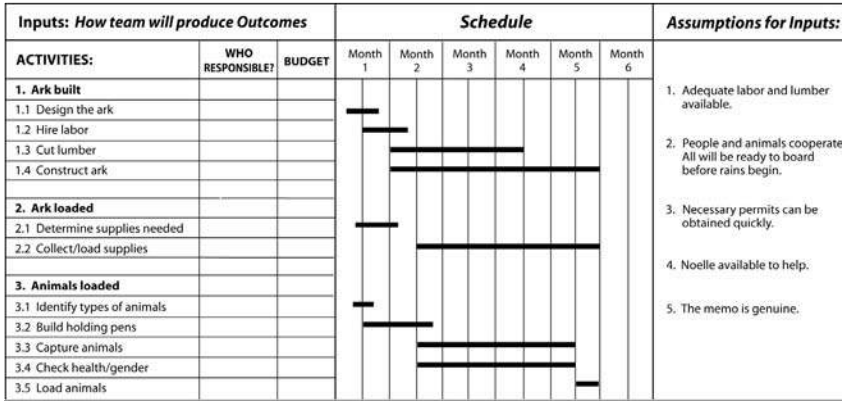

FIGURE 8.4 **Noah's Ark Project Inputs**

At this point, Noah's team descended into a dark tunnel and began sketching a giant grid on the wall with hunks of charcoal (black tape had not yet been invented). Some of his crew held up torches to provide the glimpses of light needed to develop their bar chart, capturing the steps until roll out of this life-saving, breakthrough transportation technology. You can see the result of Noah's Input planning in Figure 8.4.

Clarify Resource Requirements

The LogFrame structure invites innovation and flexibility in planning resources. In any undertaking, the three major resources of interest are time, people, and assets (money and the things money can buy—equipment, materials, and so on).

Start with a solid activity list. For each activity, identify the type of resources and estimate the cost each requires to produce the stated Outcomes. With reasonably good estimates at this level, you will end up with a defensible budget that shows what it will take to deliver your defined Outcomes. The more precision you demonstrate, the better. Smart manager can partially inoculate their project against potential budget cuts by being able to trace back the impact of cuts on the project Outcomes.

Choose the resource categories and formats that are most relevant in planning your project. Noah organized his resource budget

Inputs:	Resource Budget					Assumptions for Inputs:
	A. Materials/supplies needed	B. Cost	C. Manpower needed	D. Cost	E. Total Cost	
Action Steps:						
1.1 Design the ark	Blueprint materials	-	Architect, 2 weeks	-	-	1. Adequate labor and lumber available.
1.2 Hire labor		-	25 strong men	-	-	
1.3 Cut lumber	Saws, trees	-	50 man-days	-	-	
1.4 Construct ark (hull, deck, interior details):	Nails, saws, jigs, lumber	-	260 man-days	-	-	2. People and animals cooperate. All will be ready to board before rains begin.
hull		-		-	-	
deck		-		-	-	
interior details		-		-	-	
2.1 Determine supplies needed		-	2 man-days	-	-	3. Noelle available to help.
2.2 Collect/load supplies	Food, medicine, equip	-	10 man-days	-	-	
3.1 Identify types of animals		-	Zoologist	-	-	4. Necessary permits can be obtained quickly.
3.2 Build holding pens	Nails, saws, jigs, lumber	-	10 man-days	-	-	
3.3 Capture animals	Mating bait and calls	-	Trappers, 20 days	-	-	
3.4 Check health/gender	Vet kits	-	Vet, 20 days	-	-	5. The memo is genuine.
3.5 Load animals		-	10 man-days	-	-	
Total Cost B+D =		**B**	**+**	**D**	**E = Total Cost**	

FIGURE 8.5 **Noah's Ark Resource Budget Details**

by first identifying the materials and supplies needed for each activity, then estimating the manpower requirements, and finally turning these into financial figures. How he did this is shown in Figure 8.5.

Make Responsibilities Clear to All

The Saga of the Confused Project Team

Four people named Everybody, Somebody, Anybody, and Nobody worked together. An important Outcome needed managing, and Everybody was sure that Somebody would do it. Anybody could have done it, but Nobody actually did it. Somebody got angry, because it was really Everybody's job. Everybody thought Anybody could do it, but Nobody realized that Somebody wouldn't. As it turned out, Everybody blamed Somebody when Nobody did what Anybody could have done!

—Author Unknown

Sound familiar? Blame, wasted effort, and sour feelings occur when something important drops through the cracks due to poor communication or faulty coordination. Sorting out roles and

Linear Responsibility Chart General Format

Inputs	Responsibilties						Responsibility Code
Action Steps:	Name	Name	Name	Name	Name	Name	R = Responsible to do P = Participates C = May be Consulted I = Must be Informed A = Approves

FIGURE 8.6 **The Linear Responsibility Chart Shows Actions and Actors**

responsibilities is tricky when tasks involve multiple people, as they usually do. Fortunately, there's a simple tool to assist us: The Linear Responsibility Chart, as shown in Figure 8.6.

The Linear Responsibility Chart identifies project "actions" (tasks or activities) and "actors" (organizations/individuals) in a matrix that shows:

- All persons or organizations involved in the project (along the horizontal line).
- All tasks or activities (along the vertical line).
- The nature of the involvement of all persons in the project task (by code in the matrix).

Use this simple letter code in the cells of the chart itself to identify responsibilities of each player:

R: **R**esponsible to do (but may Delegate)
P: **P**articipates
C: May be **C**onsulted
A: **A**pproves
I: Must be **I**nformed

How to Construct a Responsibility Chart

Ideally, gather together your key players in front of a large whiteboard or blackboard to discuss the project and create your Responsibility

Chart (preceded by a LogFrame). Alternatively, one or two people can develop and circulate the chart (stamped "Draft") to others for review. Either way, follow these steps:

1. Clarify the Outcome or task of interest.
2. Draw a large matrix (on a blackboard, whiteboard, spreadsheet, or oversized paper).
3. List all the activities vertically.
4. List key actors horizontally (leave some blank columns).
5. Discuss each activity and define roles (indicate with a letter code).

Team discussion concerning task roles frequently leads to a redefinition or finer-grained breakout of tasks. For example, "Hire architect" may break out into "Write specifications," "Identify possible architects," "Conduct interviews," "Make selection," and so on.

While you can have more than one person Participate (P), Approve (A), Consult (C), or be Informed (I), only one person gets to wear the Responsibility hat (R). That person can delegate, of course (and the people delegated to would be labeled with a "P"). But one unbreakable rule is that every action only has one "R." Having multiple R's for a single task diffuses accountability and invites multi-directional finger-pointing. You can easily turn Responsibility Charts into conventional job descriptions by reading each column vertically and putting that information into standard narrative format.

Figures 8.7 and 8.8 show examples of two Responsibility Charts: Figure 8.7 identifies the responsibilities that Noah and his team agreed upon.

The chart shown in Figure 8.8 comes from the Caribbean Agriculture Research and Development Institute (CARDI), a 13-nation consortium of small island nations that pool their limited technical resources to tackle problems of common interest.

This one illustrates how responsibility charts can cut across multiple organizations. Major actors included in this example include both internal and external parties who played a role in creating their annual research and development (R&D) plans. External actors include the national Ministries of Agriculture from participating countries, while internal actors are the leaders of CARDI functional groups involved in R & D planning. Note how responsibility shifts by activity, and that some require dual levels of approval.

Noah's Ark Responsibility Chart

Inputs:	Responsibilities								
Action Steps:	God	Noah	Noelle	Ham	Shemp	Moe	Sue	Workers	Animals
1.1 Design the ark	A	P	R	P	-	-	-	-	-
1.2 Hire labor	-	R	-	-	-	-	-	P	-
1.3 Cut lumber	-	-	I	R	-	-	-	P	-
1.4 Construct ark	I	R	A	P	-	-	-	P	-
hull	-	-	-	-	-	-	-	-	-
deck	-	-	-	-	-	-	-	-	-
interior details	-	-	-	-	-	-	-	-	-
2.1 Determine supplies needed	C	A	-	-	-	-	R	-	-
2.2 Collect/load supplies	-	I	-	-	-	-	R	P	-
3.1 Identify types of animals	A	I	-	R	-	P	-	-	C
3.2 Build holding pens	-	-	-	P	R	-	-	P	-
3.3 Capture animals	-	-	-	-	P	R	-	P	P
3.4 Check health/gender	-	-	-	R	P	P	P	-	P
3.5 Load animals	I	A	P	R	-	-	-	P	P

FIGURE 8.7 Noah's Ark Responsibility Chart Pins Down Roles

CARDI Responsibility Chart

Outcome:
CARDI R&D Plan Developed

CARDI = Caribbean Agriculture Research & Development Institute

Code:

R = Responsible to do
P = Participates in doing
C = May be consulted
I = Must be informed
A = Approves

Activities	CARDI INTERNAL					EXTERNAL		
	Board of Gov.	Exec Dir.	R & D Dir.	Budget Dir.	Research Teams	Jamaica Govt.	Barbados Govt.	Belize Govt.
1. Evaluate Prior Year Results		I	R		P	C	C	C
2. Clarify Research Objectives		A	R	P	P	C	C	C
3. Set R&D Priorities	I	R	P	P	P	C	C	C
4. Establish Budget Levels	R	P	P	P	P			
5. Prepare Preliminary Project Proposals		I	A	C	R	P	P	P
6. Review and Rank All Proposals		A	R	C	P			
7. Choose Proposals to Fund		A	R	P	I	C	C	C
8. Prepare Final Plan/Budget	A2	A1	R	P	C			
9. Publish and Communicate Plan	I	R	P	P	P	I	I	I

FIGURE 8.8 CARDI Responsibility Chart Clarifies Internal and External Roles

Applying Schmidt's Law of Planning Density

It may not be written in stone, but you would be wise to consider *Schmidt's Law of Planning Density*, which is as follows:

> *Schmidt's Law of Planning Density* recommends that you plan the upcoming phase/chunk at the level of detail you need to manage it effectively, and simultaneously create less detailed preliminary plans for future phases.

For simplicity, let's assume that you have a one-year project with four phases of three months each and have created a detailed Phase One plan. *Schmidt's Law of Planning Density* suggests that the subsequent Phases Two, Three, and Four should have roughly $\frac{1}{2}$, $\frac{1}{3}$, and $\frac{1}{4}$ the level of detail as Phase One. If, for example, Phase One is a three-month effort with 25 action items, your preliminary plans for the next three phases would have roughly 12, 8, and 4 action items, respectively. A graphic depiction of this concept is shown in Figure 8.9.

Preliminary plans for future phases will have much less granularity and specificity than for your current phase, but include enough detail to spot long-lead items and future issues with present action implications.

Each phase will have its own LogFrame with phase-specific Purposes and Outcomes, but all phases will share a common, overarching Goal. You can create future phase preliminary plans as separate LogFrames. You could also include future phase planning into your current phase LogFrame as an Outcome described as "Preliminary Plans for Later Phases Developed."

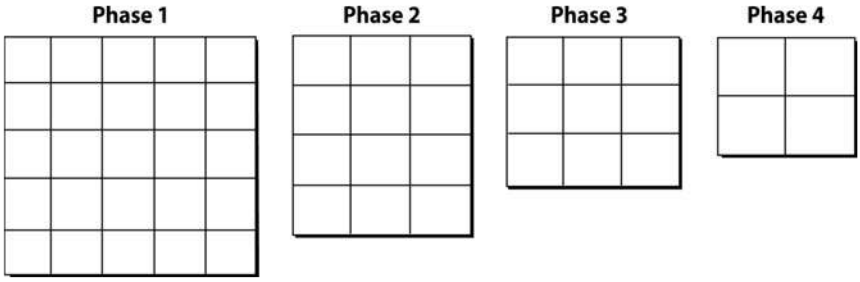

FIGURE 8.9 **Schmidt's Law of Planning Density**

Well-established phases-gate methodologies exist in many industries (e.g., pharmaceutical development, construction, software development, etc.) When standard phase names and chunks already exist (e.g., Design, Build, Test, Operate, and Maintain) use these.

But what about open-ended, non-routine or ad-hoc projects? While ready-made phase names do not exist, you can still characterize and name each phase so you have bounded execution chunks. For example, the sealed-source project team labeled an early phase "Analyze Alternatives" for doing extensive paper studies concerning the characteristics of 15 potential sites.

When you face large issues and don't know where to start, call Phase One "Problem Definition" or even "Figure Out What the Heck This Is All About." Your Phase One LogFrame Purpose might be "Problem sufficiently well understood to define a preliminary path forward." The Outcomes might include things as Stakeholders consultation completed, additional information gathered, options formulated and so on. Realize that your chunking logic will likely change in subsequent project phases (and possibly within a single phase).

Strategic chunking is more essential than you might think. Teams can stay stuck and stagnate (or drown) because they failed to chunk their project meaningfully. Naming your chunks appropriately can lead you and your team along the path more smoothly—no matter how many twists and turns may greet you.

Straight Line versus Curved and Twisty Paths

Let's contrast straight line with curved and twisty path projects. With straight-line projects, you can usually plot a clear path to the end when you start, and use your initial plan and manage, more or less, as a blueprint that won't change very much. When painting your bedroom, for example, after you choose the colors, you buy the paint, then brush, spray, or roll it on. The unknowns are reasonably few and bounded (i.e., Did I buy enough paint?) and less sensitive to environmental factors outside the project. Changes in the price of paint pigment from Peru won't affect your plans to paint your pad in pale pink.

But in curved path projects, you can only see a limited distance ahead. Think of traversing a mountain road with lots of twists and switchbacks. You know the destination and general direction is northwest, but you can't see around the corners. You must round the bend

and see the lay of the land before you scope out the next phase of the journey. Because emergent projects are curved path, chunking and phase-naming are particularly valuable to structure your project into successive phases, guided by a LogFrame for each.

Selective Zooming

At times you'll need to zoom in on a project component for more visibility. For example, here's an excerpt showing Outcome #3 from the Fircrest School LogFrame found in the Appendix. Note the appropriate density of the Input activities, which are descriptive but not overwhelming. The Inputs could then be fleshed out further as needed, but this gave enough description that the team knew the major aspects were covered.

Note that some of the tasks for Outcome #2 are large enough to justify their own LogFrame. Activity 2.2—"Design System"—is clearly a project in itself (see Figure 8.10). While this is condensed to an

Input Activities	*Responsibility*	*Resources*		*Schedule*
		$	People Resources (man-weeks)	
2.1 Assign staff	Superintendent			~~~
2.2 Design System	Expert	$25K	4	~~~
2.3 Purchase Computers	Bus. Manager	$10K	1	~~~
2.4 Create Prototype	QA Team	—	3	~~~
2.5 Collect QA Data	QA Team	—	3	~~~
2.6 Distribute QA Data	QA Team	—	1	~~~
	Total	$35K	12 man-weeks	

FIGURE 8.10 Inputs for Fircrest Project Outcome #2

Input on this Master LogFrame, the expert in charge created a spin-off LogFrame for system design.

Format Variations and Innovations

Because the LogFrame's systems thinking underpinnings are generic and flexible, so is the grid format itself. Be innovative and customize the LogFrame to your needs and add your own categories.

One such hybrid format, shown in Figure 8.11, inserts two new columns at the Input level to capture the information needed and its source. Rather than displaying a full Linear Responsibility Chart, this same Input level variation simply identifies who is responsible and lumps all other actors into an "Others involved" category next to the "R."

Some users add a fifth row beneath Inputs to describe general resource requirements when LogFrames are used for early stage feasibility studies and it's premature to cost out Inputs. Other users simplify by combining the Measures and Verification column.

This system offers you the ability to loosely couple together several LogFrames as "linked clusters," which work together towards a common Goal. The ability to bundle multiple LogFrames around themes that cut across separate organizational units can be very potent because the system flexes to match the team configuration.

You have carte blanche to tinker with the LogFrame. Remember, it's a tool that should amplify your thinking and serve your needs, rather than constrict you like an outgrown pair of pants.

Input Activities	Primary Responsibility	Others Involved	Complete By When	Resources Required	Information Needed	Information Source
1.1						
1.2						
1.3						
etc.						

FIGURE 8.11 Innovative Format for Input Planning

Defining the Next Action Step

Do you know anyone who occasionally procrastinates on key tasks because they seem too big (that you don't even know where to start); so simple (that you feel you could put it off); or too fuzzy (that you aren't sure what needs to be done)? Do you sometimes want to move ahead, but aren't sure about your next step? Your mind can play tricks and bombard you with reasons to not take action.

A professional associate and master of exquisite execution, David Allen, taught me how to solve that problem. He shares this next tip in his must-read book *Getting Things Done*, as paraphrased here:

"Define the next action step" is a success principle that has proven itself enormously valuable. For example, confronting a task like "Improve marketing plan" can lead to getting overwhelmed or stuck because you can clearly see several tasks within the task. So, it starts to look huge every time you try to get going on it.

What is the solution? Define the next discrete, doable step that you can take. What is it? "Review the present plan?" "Locate the present plan under other stuff on your desk?" "Retrieve the plan from the circular file?" Then, define the next action step, i.e., "Read old plan." What then? "Highlight good parts in yellow." And then, oh, "Find yellow highlighter!" (Side Note: This last step suggests that a "get organized" project is required soon in order to get your steps truly in sync.)

By breaking your big task into discrete packets at this level of granularity, you effectively defuse your fear by showing yourself that the big scary task is actually a series of small, simple tasks. You don't need to go atomic on your "What do I do next?" breakdown. Just take it to the level where you can envision yourself doing those necessary, important, mostly ordinary tasks that effective humans do to get things done. Some of the most productive next steps might be to:

- Meet with other people
- Call someone on the phone
- Send an e-mail
- Locate a document on your computer
- Do online to research about some question
- Make a decision
- Create a new electronic file and brainstorm some ideas

To make your actions most productive, ask yourself whether or not there is still some prior step to be done in preparation for the meeting, phone call, or e-mail such as getting some missing information. That is your real next step. Take it now, and take your next step after that, and then the next, and soon you'll be river dancing yourself, and your team, into a flow of true project productivity. There's a psychological lift from getting into this "flow state" because breaking seemingly insurmountable tasks into sure-thing next steps builds inner strength and momentum.

Key Points Review

1. Chunk smart to avoid categorical pitfalls that can roadblock your project from the start. Get the chunks right, and you are on the way. Be explicit and name your chunks, then LogFrame at least the first one.
2. In any undertaking, the three major resources of interest are time, people, and assets. A wisely chunked activity list is the starting point for the schedules, responsibility charts, and resource budgets.
3. Make your LogFrame a high-level summary rather than a detailed action plan. LogFrame Inputs can be illustrative and not definitive. They are simply the starting points for more detailed planning using other task management tools. The LogFrame structure affords you great flexibility.
4. Developing a project Responsibility Chart provides a way to sort out potential coordination difficulties in advance. Turn these into job descriptions by putting the contents into standard narrative format.
5. *Schmidt's Law of Planning Density* recommends that you plan the upcoming phase/chunk at the level of detail you need to manage it effectively, and simultaneously create less detailed preliminary plans for subsequent phases. Like the law of gravity, *Schmidt's Law* is more than a law; it's also a good idea.
6. "Define the next action step" that is discrete and doable is a success principle that has proven itself enormously valuable. Break down tasks into the very next action you need to take.

Application Step #4:

Question 4—How do we get there?

At this point you can use software or continue with old-fashioned technology—pencils!

1. Confirm the Outcomes. Affirm that they are your current best guess as to the necessary and sufficient set needed to reach the Purpose.
2. List key activities for each Outcome, chunked out at roughly the same level of detail. Limit your activities to four to seven per Outcome, so you don't get overwhelmed.
3. Identify tasks sequences by examining predecessor or successor events. Determine what the next step would be after each step as you ask yourself if there's anything else that needs to be done before, after, or in between. Develop a Gantt chart or similar task schedule. (Keep in mind that although this process may be tedious, be thankful you're not working with rolls of black tape.)
4. Identify resources needed for each task.
5. Clarify responsibilities using the Responsibility Chart.

If you've been doing the Application Steps at the end of Chapters 5, 6, 7, and 8 to your project, you have now fleshed out a first draft LogFrame. In the Appendix, you'll find a self-administering quality checklist that you can use to determine how well your project design hangs together. Use this to spot and correct weaknesses. After cleaning up your design, stamp it *draft* and circulate it to a few key players for some live feedback.

This chapter concludes the section about the LogFrame Approach proper. Return to this section again and again for more insights on using it for current and for all future projects, both personal and professional.