

# Ten Problem-Solving Tips

## Introduction

Problems come in all sizes, shapes, and colors. There is no single or simple step-by-step process guaranteeing us we will solve every problem we encounter. We are faced instead with the requirement to configure or adapt our problem-solving processes to fit the problem at hand. As problem solvers, we have more in common with the cabinet-maker than the assembly-line worker. What we need are plans and blueprints, high-quality materials, a decent place to work, a well-stocked toolbox, and the skill and knowledge necessary to properly select and use the tools in it. Here are ten problem-solving tips—ideas for beefing up the tools in your problem-solving toolbox. (You can skip around from tip to tip if you like, by clicking on the links in the list below. You can return to the list by clicking on the back button.)

1. [Focus on the solved state.](#)
2. [Be clear about \*all\* your goals and objectives.](#)
3. [Expand your definition of "Define the Problem."](#)
4. [Think of problem solving as a cover-the-bases activity.](#)
5. [Draw diagrams and otherwise picture the structure of the problem.](#)
6. [Take the concept of cause with a grain of salt.](#)
7. [Watch out for "disconnects."](#)
8. [Be aware of your own blinders.](#)
9. [Develop your own system for solving problems.](#)
10. [Research the subject matter.](#)

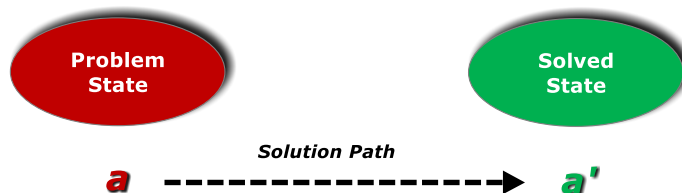
## Focus on the Solved State

[BACK](#)

Pay at least as much attention to the solved state as you pay to the problem state. As Robert F. Mager's (1962) fable of the sea horse reminds us, "if you're not sure where you're going, you're liable to end up someplace else—and not even know it (p.vii)."

When solving a problem, we typically wish to do more than simply rid ourselves of some unacceptable situation. More often than not we are trying also to achieve some other, more desirable state of affairs.

Theoretically speaking, we're trying to move from the problem state to the solved state. We do so by traversing what is called "the Solution Path" (see below).



It seems obvious that if we do not focus some of our attention on the solved state, the likelihood of attaining it is diminished. Unfortunately, the problem state typically attracts all our attention. The

squeaky wheel gets the grease. On occasion, this is an appropriate response. If the roof is caving in, then discussions about where to go can wait until we're safely outside.

But, if we're not in an emergency situation and if we still have nothing more in mind than doing something to rid ourselves of the problem state, we can create situations where the solution to one problem creates one or more new problems. Chester Barnard (1938) labeled solutions that create new problems as "inefficient" solutions. An "efficient" solution, of course, creates no new problems.

# Ten Problem-Solving Tips

There are several ways of focusing on the solved state. One is to define it the same way we would define the problem state (more about that under Tip #3). Another is to list possible measures or indicators of its attainment. Ask yourself questions like these: “How will I know the problem has been solved? What will I accept as evidence? What does the solved state look like?” Yet a third way is to be clear about *all* the goals and objectives of the problem-solving effort. (This last point is so important that it constitutes a tip all its own—the next one.)

## Be Clear about All Your Goals and Objectives

BACK

Ultimately, the aim of problem solving is action. To engage in problem solving is to search for a solution. To actually solve a problem is to implement the solution that has been found and demonstrate that it works. Solving problems requires intervention as well as investigation.

Intervening in complex organizations requires of us that we carefully think through the likely effects of any actions we are contemplating. Actions in an organizational context often “ripple” outward from the point of intervention, sometimes having unforeseen and unintended consequences. Our goals and objectives, therefore, are typically multi-dimensional; that is, we seek to eliminate some conditions, and to achieve others. There also are conditions we seek to preserve or avoid.

<b>No</b>	<p><b>I</b></p> <p><b>Achieve</b></p>	<p><b>III</b></p> <p><b>Avoid</b></p>
<b>Yes</b>	<p><b>II</b></p> <p><b>Preserve</b></p>	<p><b>IV</b></p> <p><b>Eliminate</b></p>
	<b>Yes</b>	<b>No</b>
	<b>Do We Want It?</b>	

One way of examining the multi-dimensionality of our goals and objectives is to compare and identify any disparities between our perceptions (what we have) and our preferences (what we want). This comparison is shown in The Goals Grid on the left.

If we don’t want something that already exists, our goal is typically one of eliminating it. If we want something that doesn’t exist, our goal is ordinarily one of achieving it. Four categories of goals and objectives can be derived from the interplay of our perceptions and preferences: Achieve, Preserve, Avoid, and Eliminate (Nickols, 1992).

For any problem situation, it is useful to ask the following questions as a way of clarifying *all* your goals and objectives:

- What are we trying to *achieve*?
- What are we trying to *preserve*?
- What are we trying to *avoid*?
- What are we trying to *eliminate*?

## Expand Your Definition of “Define the Problem”

BACK

Perhaps the best-known step in the problem-solving process is the one most people think of as the first step: “Define the Problem.” This is probably the most misunderstood and poorly executed step in the

# Ten Problem-Solving Tips

process. For many people, "Define the Problem" means simply to provide a written definition or statement of the problem. There is much more to it than that.

To define means to establish boundaries, to encompass, to enclose, to locate, to isolate, to distinguish, to differentiate, to set apart. To define the problem state (or the solved state) means, at the very least, to do the following:

- To establish boundaries; to delineate (*Locate*).
- To give distinguishing characteristics; to differentiate (*Isolate*).
- To state the nature of; to describe precisely (*Articulate*).
- To state the meaning of; to provide a definition (*Explicate*).

Rarely are definitions of the problem state or the solved state crystal-clear up front. Clarity typically develops over time. In many cases, the definition of a problem may be considered complete only after the problem has been solved. Until then, it is a shifting, evolving, changing part of the process. Thus, although "Defining the Problem" is a good step with which to begin the problem-solving process, it is only a starting point and it must be revisited on a regular basis. This also is true of any definition of the solved state.

## Think of Problem Solving as A "Cover-the-Bases" Activity

[BACK](#)

Information does not make itself available to suit the requirements of anyone's problem-solving process. Solving a problem in a complex organization has much in common with detective work. We are forced to follow leads and unearth clues. Further, it is generally the case in complex organizations that no one individual possesses all the information necessary to solve a given problem. Vital information appears in bits and pieces. We have different backgrounds, perceptual filters, and value priorities. Different people seek and assimilate information in different ways.

Consequently, if you listen carefully to almost any discussion of a problem in a group setting, what you'll hear is conversation that shifts from problem to symptom to cause to solution and back again, often in no particular order, a fact much lamented by Charles Kepner and Benjamin Tregoe (1965) in their classic little book, *The Rational Manager*. Such "bouncing around" is natural. Don't worry about it. Above all else, don't try to force yourself (or others) to follow some lock-step, linear process. The task of problem solving is very much a type of intelligence work, a matter of piecing things together.

A systematic approach is necessary but the point of having one is to make sure you tend to all the things that need tending to, that you "cover the bases," not trot around them in a 1-2-3 fashion. Here is a list of twelve "bases" to be covered or tasks that typically need tending to in the course of solving a problem:

1. Define the problem.
2. Specify the solved state.
3. Build Consensus and support.
4. Troubleshoot the problem.
5. Design a solution.
6. Identify the means of change.
7. Settle on a course of action.
8. Reconcile restraints and constraints.
9. Prepare plans and schedules.
10. Take action.
11. Assess effects and consequences.
12. Adjust future actions.

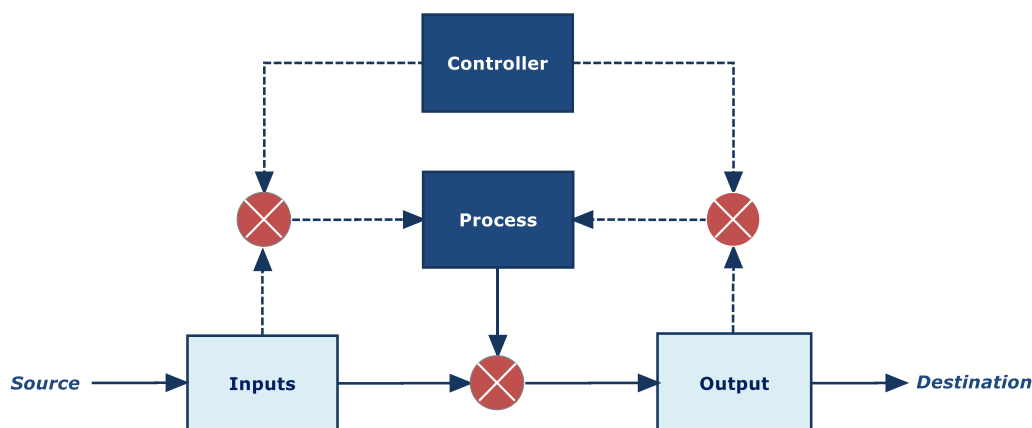
# Ten Problem-Solving Tips

Ordinarily, steps 4 and 5 are mutually exclusive; you do one or the other but not both. If you're dealing with a problem where something has gone wrong, then your best bet, at least initially, is to focus on finding and fixing the cause of the problem. On the other hand, if you're out to achieve some state of affairs never before attained, or if the cause of the problem has been found but can't be corrected, then you'll have to design a solution to the problem. In either case, you'll have to settle on a course of action and carry it out.

## Draw Diagrams and Otherwise Picture the Structure of the Problem

[BACK](#)

A picture or model of the elements and relationships in a problem situation will help you to more quickly and more completely grasp the situation and figure out what to do about it.



Consider, for example, the diagram shown above. It depicts the structure of a general-purpose work system. The elements of this system include inputs, a processor, outputs, a controller, and two control loops. On the front end of this system is a task initiation loop and on the back end is an evaluation and termination loop (the dotted lines). The relationships among these elements are such that inputs to the work system interact with the processor. The interactions between inputs and processor, which typically consist of prefigured routines, are referred to as "processes." These processes produce the work system's outputs. All this occurs under the watchful eye of the controller.

If the outputs of the work system are faulty, several possibilities are suggested by the structure of the diagram above. The inputs might be faulty. The processor or the controller might be malfunctioning. Perhaps one or the other or both of the control loops is open, and no information is getting through. Whatever the contributing factors, the diagram provides guidance regarding places to look for what might be causing the problem and for what might have to be changed in order to solve it.

The use of diagrams or schematics as an aid to problem solving is not new. Technicians have been using schematics as troubleshooting aids for years. Computer programmers and systems analysts are familiar with, if not dependent on, flowcharts and data structure models. Industrial engineers have relied on process flow diagrams ever since the days of Frederick Winslow Taylor. Diagrams and schematics should be found in your problem-solving toolbox, too.

# Ten Problem-Solving Tips

Most important, get in the habit of visualizing the problems you tackle.

## Take the Concept of Cause with A Grain of Salt

[BACK](#)

If ever there was a time-waster in problem solving, it has to be the search for the cause of the problem.

Don't misunderstand—the concept of cause is frequently relevant, but its usefulness depends on the kind of problem being solved. It's not relevant all the time and, for some problems, it's never relevant.

For certain kinds of problems, mostly in contrived physical systems like the work processing system depicted in Figure 3, the concept of cause makes sense. Things are going along just fine, something happens, and matters take a turn for the worse. A component in a piece of equipment burns out. A fuse blows. A tire picks up a nail and goes flat. In such cases, and countless more of a similar nature, the search for cause is indeed relevant.

But not all problems can be said to be caused. And not all causes can be corrected.

The stock market crashed in October of 1987. The stock of many companies plummeted in value. Why? Well, not even the SEC can provide an answer to this question. Some attribute it to computerized trading and faulty buy-sell algorithms. Others chalk it up to a loss of investor confidence. Whatever caused the crash of '87, no one has yet found the cause and fixed it.

At a more mundane level, consider the employee who doesn't know how to perform a certain task. Suppose this person was never trained to perform the task. Suppose the task itself was only recently made a part of the person's job, the result of a reduction in force in response to straitened economic circumstances. What's the "cause" in this case? Is it the employee's lack of knowledge? Is it the fact that she was not trained? Is it the newness of the task? Is it the reduction in force? Or is it the economic conditions that led to the reduction in force?

The point of this tip is a simple one: The concept of cause is relevant in some cases but not in others. Take the time early in your problem-solving efforts to decide if the concept of cause is or isn't relevant before you spend time and energy looking for something that doesn't exist or that can't be corrected.

## Watch Out for "Disconnects"

[BACK](#)

The term "disconnect" refers to the uncoupling of solutions from the problems they are intended to solve. This commonly happens when one person or group of people defines the problem, a second person or group of people is assigned to figure out what to do about it, and yet a third person or group of people is tasked with actually implementing the solution.

Vertical disconnects occur frequently. An executive spots a problem. A middle manager gets the job of coming up with an action plan. And a line manager or supervisor gets the unenviable job of doing the unthinkable for unfathomable reasons. No one clarifies, questions, or challenges the definition of the problem (which might or might not have been sharply defined to begin with). As a result, the solution has no bearing on the original problem and might have very undesirable side effects as well.

Horizontal disconnects also occur frequently. These regularly surface in efforts to develop computer-based information systems. This scenario starts with business people laying out the problem for business analysts. The business analysts translate and relay the problem to the systems analysts. The systems

# Ten Problem-Solving Tips

analysts translate and relay the problem to the programmers. A system gets built. Then, one of three things happens:

1. The system is scrapped because it's taking too long and costs too much to develop.
2. The system doesn't work as intended because no one knows what was intended in the first place, or because the definition of the problem was lost in repeated translations, or because it was left at one of the relay stations.
3. The system works as intended, but all it accomplishes is the automation of what was being done manually, making the underlying processes all the more difficult to get at and improve. In effect, resistance to change is mechanized.

Beware of disconnects!

## Be Aware of Your Own Blinders

[BACK](#)

We all wear a custom-tailored set of "blinders," that is, perceptual and value-based filters that lead us to see or not see certain things, or to interpret them in certain ways. These blinders are sometimes useful, sometimes not. In all cases, it pays to know what they are.

When it comes to problems and problem solving, there are several ways we can get a peek at how we're placing our own personal "spin" on things. One is the way we label the problem.

Labels invoke what Allen Newell and Herbert Simon (1972) termed the "problem space." If I label a problem a "personnel" problem, for instance, I invoke one set of models and concepts, and call into play a certain realm of knowledge. If I label that same problem a "business" problem, or a "production" problem, or "an attitude" problem, I invoke different sets of models and concepts, and different realms of knowledge. Problem labels influence the way I frame a problem; they lead me to look in different places for different things. It is useful to pause during the course of working on a problem and ask yourself how you've got it labeled. Deliberately changing the label can also be useful.

Another way of gaining insight into the way we frame various classes of problems, and thus invoke certain families of solutions, is to examine the solutions we propose. If we think training is the answer, we have framed the problem one way. If we recommend a reorganization, we've framed it a very different way. If we think replacing certain people will do the job, we have yet a third view of the problem. Looking at the solutions we propose—and prefer—will tell us a lot about the way we classify and frame problems as well as our underlying biases and filters.

Yet a third way of getting at our own blinders is rooted in our goals and objectives. Thinking about what we wish to achieve, preserve, avoid, and eliminate—and why—can help us sort out how much of the analytical axe we're grinding is personal and how much of it is rational problem solving.

## Develop Your Own System for Solving Problems

[BACK](#)

This amounts to recommending that you reinvent the wheel. Be that as it may, the point is that you simply cannot pick up someone else's conceptual tools and make immediate use of them. A certain amount of adaptation is unavoidable.

As a problem solver, you must develop your own system for solving problems. You have to develop a scheme whereby you can tell if a given problem solving tool is useful or not and put it to use accordingly.

# Ten Problem-Solving Tips

An earlier example of a problem-solving tool that might or might not be useful in a given situation was the concept of cause. Below are three more. These are techniques.

- **Brainstorming.** Coming up with ideas is important, and brainstorming is a good way to do that. But brainstorming won't take the place of detailed financial analysis. And brainstorming won't tell you who's going to oppose your definition of the problem. And brainstorming won't tell you how to persuade that person to support your definition—or lead you to acknowledge that there may be merit in her position. Brainstorming is a useful technique, but not for all occasions. The definitive book on this subject is *Applied Imagination*, by Alex Osborn.
- **Force-Field Analysis.** This, too, is a useful technique—in certain situations. But, think about that flat tire mentioned earlier or, better yet, that piece of equipment with the burned-out component. Force-field analysis isn't likely to be of much use in determining which component burned out (and neither is brainstorming). In general, force-field analysis is a useful technique for examining psychological and sociological factors or forces, but there are better tools available for diagnosing problems in physical systems. A good summary of this technique can be found in the second edition of *The Planning of Change*.
- **Root Cause Analysis.** This technique has its roots in quality control. It is most clearly associated with famed Japanese quality guru, Kaoru Ishikawa, and it typically makes use of a diagramming method yielding a product known as an "Ishikawa" or "fishbone" diagram. It is a structured method for identifying the factors that contribute to the effects observed in a given situation, but it is not a means for identifying the structure of the problem. Those interested in the technique of root cause analysis can find adequate explanations in *Kaizen*, by Masaaki Imai, *The Customer Driven Company*, by Richard C. Whiteley, and in Section 22, of Juran's *Quality Control Handbook*.

## Research the Subject Matter

[BACK](#)

There is no easy way, no shortcut to becoming a first-rate problem solver. It takes years of work and study and practice—and researching the subject matter.

Many books have been written about problem solving and solving problems. It would take a great deal of time to read them all—if you were so inclined. The good stuff is scattered about in several books. A tidbit here, a morsel there, a snack now and then, but hardly ever a full meal let alone a feast.

Below is my personal list of favorites, what I view as the five best books ever written about problem solving, with a few comments regarding each.

- *How We Think* (1910). John Dewey. Heath: Boston

Dewey's book addresses problem solving in the larger context of reflective thinking. Dewey's explication of the problem-solving process is as relevant today as when he first wrote it, including his introduction of now commonplace concepts such as the definition of a problem situation as a discrepancy or deviation. If you can't find it in the library, try an antiquarian book store; I found a first edition in first-rate condition for \$25.00.

- *The Functions of the Executive* (1938). Chester I. Barnard. Harvard University Press: Cambridge

# Ten Problem-Solving Tips

Barnard's book is a classic in management literature. Of particular interest to problem solvers are his notions of the efficiency and effectiveness of solutions as well as of the problem-solving process that produces them. Barnard's book is extremely relevant to anyone wishing to grasp the nature and structure of "business" problems.

- *The Rational Manager* (1965). Charles H. Kepner and Benjamin B. Tregoe. McGraw-Hill: New York.

Messrs. Kepner and Tregoe's little book is a classic too. It is the bible for many organizational problem solvers. However, the problem-solving process it presents rests squarely on cause-effect thinking and, in my opinion, applies chiefly to problems in physical systems. It is perhaps the best of the troubleshooting genre of problem-solving books. The Kepner-Tregoe Corporation in Princeton, New Jersey still sells the second edition of this little gem.

- *Analyzing Performance Problems* (1970). Robert F. Mager and Peter Pipe. Fearon: Belmont.

This is for many people the definitive book on the subject of analyzing human performance problems. It is an easy read, as is the case with all of Bob Mager's books.

- *Human Problem Solving* (1972). Allen Newell and Herbert Simon. Prentice-Hall: Englewood Cliffs.

This book is more than 900 pages in length, and it is not an easy read. But this same book is also a gold mine. I've been extracting high-grade ore out of it for the past 20 years. Hint: Try reading just the chapter summaries first then delve deeper as you are so inclined.

## References

Some of the references cited in this article are contained in the list of books immediately above and it would be pointless to list them again. The other references are as follows:

1. *Applied Imagination* (1953). Alex Osborn. Charles Scribners & Sons: New York.
2. *Preparing Instructional Objectives* (1962). Robert F. Mager. Fearon: Belmont.
3. *The Planning of Change*, 2nd Edition (1969). Warren Bennis, Kenneth Benne, and Robert Chin, Editors. Holt, Rinehart & Winston: New York.
4. *Kaizen: The Key to Japan's Competitive Success* (1986). Masaaki Imai. Random House: New York.
5. *Juran's Quality Control Handbook*, 4th Edition (1988). Joseph M. Juran and Frank M. Gryna, Editors. McGraw-Hill: New York.
6. *The Customer Driven Company: Moving from Talk to Action* (1991). Richard C. Whiteley. Addison-Wesley: Reading.
7. "Objectives, Systems, Patterns, Politics, and Conflict" (May 1992). Frederick W. Nickols. *Performance & Instruction*. NSPI: Washington, D.C.

## Note

This paper was first published under the pen name of Peter W. Taylor and appeared in the July 1994 issue of *Performance and Instruction* with the title, "Ten Tips for Beefing Up Your Problem-Solving Toolbox." I have since retitled it, lightly edited it, and added the ability to skip around in the document.



# Ten Problem-Solving Tips

## Further Reading about Solution Engineering

The papers listed below can be found in the articles section of my web site.

1. [Choosing the Right Problem-Solving Approach](#)
2. [Forget about Causes, Focus on Solutions](#)
3. [Reengineering the Problem-Solving Process](#)
4. [Solution Engineering in Action: A Really Good Example](#)
5. [Solution Engineering: Some Basics](#)
6. [Three Cases of Figuring Out What to Do](#)
7. [What's Your Intervention Logic? – The Links to the Bottom Line](#)

## For More Information

Contact Fred Nickols by [e-mail](#) and visit the articles section of his web site by clicking [here](#). There, you will find more about problem solving and Solution Engineering.