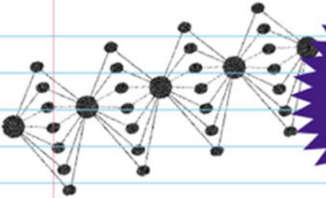

By the Coauthor of the Bestselling HOW TO MAKE MEETINGS WORK

More than 600,000 Copies Sold

# HOW TO MAKE COLLABORATION WORK

Powerful Ways to Build Consensus Solve Problems, and Make Decisions

DAVID STRAUS

An Excerpt From

# *How to Make Collaboration Work: Powerful Ways to Build Consensus Solve Problems, and Make Decisions*

by David Straus

Published by Berrett-Koehler Publishers

# Contents

Chapter 1

# The Process of Human Problem Solving

In 1965, I entered the architecture program at Harvard Graduate School of Design (HGSD). The basic method of teaching design at HGSD, it turned out, was to assign students to design a certain type of building or space and then critique the designs. The critiques were very formal and were modeled after the process an architect might go through in trying to sell an idea to a potential client. The students presented their designs to a panel of faculty members and professional architects. During these critiques, and during class, there was little discussion of the creative *process*—of how we came up with the designs. In fact, there was no accepted language to discuss design methodologies at all. Instead, the professors were mainly interested in the *content* of our designs.

About halfway through my first year, I began to think something was wrong with my eyes. The drawings on my drafting board looked blurred. I had trouble reading. Panicked, I made an appointment with a recommended optometrist.

After examining my eyes, the doctor led me into his office, motioned for me to sit in a comfortable chair, and then posed one of the most insightful questions I've ever been asked. "Your eyes are fine," he said. "What is it that you don't want to see?"

I suddenly realized what it was that I didn't want to see, that I was unconsciously denying. Even though I was paying for a graduate education in architecture, no one was going to teach me explicitly how to design. No one was going to identify and make visible the mental processes of design.

This flash of insight marked, for me, the beginning of my own *process awareness*. This moment was the first step in my lifelong journey to uncover and demystify the processes of individual and group problem-solving, and to transfer these concepts and tools to others.

In this chapter, I hope to shed some light on how individuals solve problems. This information is fundamental to an understanding of collaboration—of how individuals solve problems *together*. It's simply not possible to practice collaborative problem solving effectively if you have no understanding of human problem solving in general. This chapter is a bit heavier on theory than the other chapters in the book, because it serves as a basis for all that follows. To make it as accessible as possible, I use the story of my own experience discovering these concepts and learning their relevance to collaborative action.

## Some Definitions

In the Introduction, I defined *collaborative problem solving* as "the process people employ when working together in a group, organization, or community to plan, create, solve problems, and make decisions." I also talked about what I meant by *collaboration*. Here, I want to say more about the *problem solving* part of the term.

For some people, a *problem* implies something bad, a situation to avoid. In certain contexts, to focus on problems is seen as attending to the negative, the pathological. So some people substitute the word *opportunity* to emphasize the positive, to look at things on the bright side or to look at the possibilities of the future. But *opportunity solving* and *opportunity finding* are clumsy substitutes for *problem solving*, and there is already a whole literature on *creative problem solving*, so I'm going to continue to use the word *problem* in this book.

In any case, I don't view problems as negative. I define a problem as "a situation that someone wants to change." *Problem solving*, therefore, in its most general sense, is *situation changing* or taking action. It includes most of what we do all day long: communicating, learning, planning, working, and making decisions. At work, for example, you may need to make hiring and firing decisions, communicate with employees, fix quality-control problems, sell your products, and so forth. I would call all of these activities *problems*, since they are all situations you need to change—things you need to do something about.

These situations need not be bad. They include positive situations that you may want to reinforce or increase, like supporting employees to continue their education by offering matching funds to attend training programs. Creating a vision for your organization is also a problem-solving activity just as much as analyzing why the assembly line is causing defects in your products.

Also, under my definition, a problem is only a problem if there is an agent present—someone who cares and wants to take action. If you see your kids arguing and it doesn't bother you, you don't have a problem. Your kids may have a problem, but you don't.

Humans are designed for continual problem solving. If all stimuli are removed from a person's environment (as in an isolated prison cell), often that person will go mad. We are constantly making little changes in our environment, from shifting our sitting position to planning for the future. In this book, then, *problem solving* will refer to all the cognitive processes directed to purposeful action, from perceiving and innovating through planning and decision making.

## My Intellectual Quest

Soon after the revelation that came during my eye exam, I set out to teach myself how to design—how to solve the design problems presented by my professors. I could find no useful books about how to design, and my professors were not very helpful. So I started keeping detailed design notebooks, in which I tried to track my own thought processes, to become more aware and at least "consciously incompetent" about the ways I was attacking a project. I found that when I tried a new design strategy, a different way of looking at a three-dimensional structure, I was suddenly able to do things I couldn't do before. For example, l learned how to draw a *section perspective*, which presents a "slice" through a building and a perspective of what you might see from that point. Through this drawing you see your design in a different way. I also learned how to build simple models out of Styrofoam blocks, with which I could arrange spaces in different ways without having to make new drawings.

In the design notebooks, I documented what I was thinking about when I was stumped, when I kept repeating the same mental process without success. Then, when I discovered another strategy from informal discussions with classmates or professors, I could consciously add it to my growing repertoire of design methodologies. I could also retrace my steps in my design notebooks and see how this new strategy might have helped me break fixation—how it might have served as a way out of a trap in which I had found myself.

I saw clearly that there was a relationship between the strategies I learned and my ability as a problem solver. Each design strategy provided a different way to attack an architectural problem, and the more I learned, the better a designer I became. And yet these strategies were not being explicitly taught.

To satisfy my own curiosity about design methodologies and problem solving, I began to audit courses at Harvard in cognitive psychology with professor Jerome Bruner. In these courses, I was introduced to the work of Allen Newell and Herbert A. Simon from Carnegie-Mellon University, as well as that of Ulrich Neisser.

Neisser (1967) was making a case for cognitive structures, or frameworks, about thinking processes. He maintained that it was possible to describe *how* you were solving a problem, and that it was helpful to do so. Without a framework to describe a subject, he said, it's hard to make distinctions and therefore to acquire and retain new information. For example, if you know nothing about general species of birds (e.g., flycatchers, warblers, wrens, thrushes), they all sort of look the same. When you see a small bird you have never seen before, you might not even know you've never seen it and you probably won't remember much about it. The same is true about problem solving. Without a language of process, without knowing something about the different strategies that can be used to solve problems, it is difficult to learn and acquire new ones.

### There Is No One Right Way!

It was the work of Newell and Simon that provided me with the biggest "aha" of that time, however—one that was to guide my work for years to come. Their writings brought out the simple but powerful fact that human problem solving is an educated trial-and-error process (1972). Put another way, *there is no one right way to solve problems.* We can use a variety of strategies, but none of them will guarantee success. Some of them may be more useful in certain types of situations. But there is no single right way. The implications of this realization are profound. Over the years, my colleagues and I used this understanding as the basis for developing approaches to, and teaching, collaborative problem solving.

## Heuristic vs. Algorithmic Problem Solving

What Newell and Simon (1972) did was to clarify the differences between *heuristic problem solving* and *algorithmic problem solving*.

To illustrate, take the example of trying to find a lost contact lens. The algorithmic approach to searching might involve getting on your hands and knees and systematically crawling back and forth across the floor, trying to cover every square inch. If the contact lens is on the floor as opposed to on the sofa or in your clothes, and if you are very sharp-eyed, you will find your lens this way. However, it may take a very long time. The heuristic approach is to try different strategies in succession. You might start with the common "where were you last" approach. Then you might try to retrace your movements, shake out your clothes, get down on your knees and scan the floor, or turn up the lights to see if you can catch a reflection off the missing lens. Usually one of these heuristic strategies will work quite well and save a great deal of time compared to the algorithmic approach. In short, a *heuristic* is a strategy that is flexible and quick but doesn't guarantee success, while an *algorithm* is an approach that is systematic, rigid, and time consuming, but will ultimately guarantee success.

Newell and Simon discovered a great deal about human problem solving by trying to program computers to solve problems that are reasonably easy for humans. To greatly simplify, Newell and Simon found that there were no simple algorithms to deal with challenges like playing chess or recognizing a face. Such problems require heuristic strategies. What seems to characterize the human brain is our ability to think up heuristics and to be flexible and creative in our application of them.

Take, for example, the anagram of "TABLAERY," in which the challenge is to rearrange the letters so that they spell an English word. The algorithmic approach would be to try every combination of letters and test each to see if it is a word. There are 20,160 possible combinations of the letters, so at a rate of one new combination every ten seconds, it would take you up to fifty-six hours to find a solution this way. Using a heuristic approach, however, many people can come up with an answer in fifteen or twenty minutes. Take a moment and play with the problem, noticing what you do. Notice that you try different ways to solve it, different heuristics. Most people try, for example, rearranging the letters by consonants and vowels, looking for smaller words on which to build, avoiding letter combinations that aren't used in English, and even writing each letter on a separate piece of paper and physically rearranging them. Each of these heuristic strategies may lead you to a solution, but none of them will guarantee success. (See page 33 for the solution—but only after you've tried several heuristics!)

## A Simple Model of Human Problem Solving

So Newell and Simon demonstrated that human problem solving is a trial-and-error process involving choosing a heuristic strategy, testing it, and, if it doesn't work, choosing another. This heuristic cycle is illustrated by the model in Figure 2.

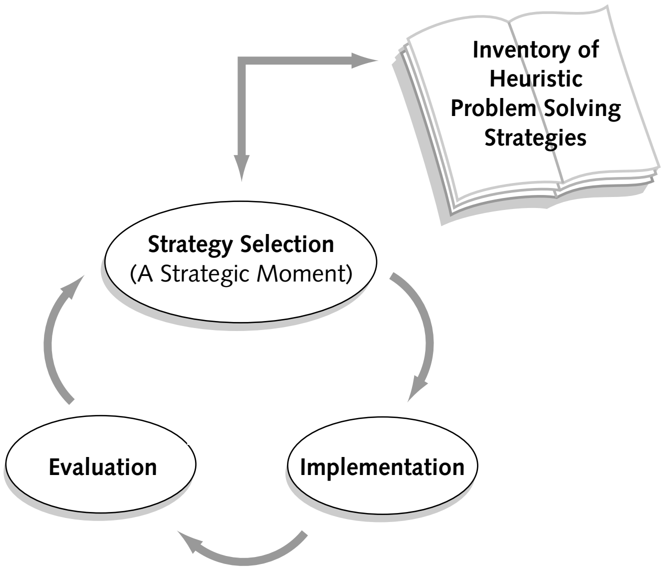


**Figure 2:** The heuristic cycle of human problem solving

The problem-solving cycle begins with what we call a *strategic moment*—that familiar point in time at which whatever you have been trying isn't working anymore. For example, in your search for your contact lens, you may try shaking out your clothes to see if the lens might be caught in the cuffs of your pants. If nothing falls out, you have to try something else. This is the moment at which you consult the repertoire of strategies you have learned, pick one, and implement it. Perhaps, for example, you decide to simply vary the implementation of your current heuristic (e.g., shake out your shirt rather than your pant cuffs), or you may change your approach and select a new strategy. Based on the results, the feedback from your efforts, you evaluate the success or failure of your trial and then you are back to another strategic moment.

This cycle of action/reaction usually happens so quickly that we're not aware of it. It's when we get stuck in a strategic moment that it's helpful to be able to assume conscious control of our problem solving. This is especially true in a group problem-solving situation, as we will see.

So, the great fact I learned in graduate school is that human problem solving is fundamentally a trial-and-error process employing heuristic strategies. There is no one right way. There are no simple algorithms for dealing with most of the open-ended problems we face every day. However, as I was soon to learn, there is a set of very useful heuristics that can be employed.

## A Limited Set of Problem-Solving Heuristics

My own search for heuristics led me to the University of California at Berkeley during my thesis year (1968–69), while I was still registered at the Harvard Graduate School of Design. At this time, the School of Environmental Design at UC Berkeley was a world center for the study of design methodology—of how architects design. Berkeley professor Sim Van der Ryn had received a grant from the National Institute of Mental Health to review the literature on design methodology and try to make some sense of it. He hired me to assist him. It was a perfect opportunity for me to pursue my interest in human problem solving.

I began my research by asking some of Van der Ryn's renowned colleagues to share with me their different design tools. To my surprise, they strongly resisted getting involved in my project. Each of them was sure that he or she had discovered the right approach to design and was not interested in exploring the full range of design methods. So I turned to other sources, reviewing the literature and interviewing researchers from a variety of fields who were exploring the nature of human problem solving.

I began to make a list of the problem-solving methods I uncovered. I discovered that thinkers from very different fields often used essentially the same methods, although they sometimes used different terminology to describe these methods. Indeed, similar methods kept cropping up over and over. I realized that any given problem-solving method could be applied to many different contexts. For example, *brainstorming*, a common way of generating ideas, can be applied to different problems in many different fields. You can brainstorm ideas for creating an ad campaign or solving a calculus problem or finding a place to have dinner.

Like most problem-solving methods, brainstorming involves multiple steps—multiple heuristics. Brainstorming involves, first, *purging* or expressing out loud all the ideas that come into your head; then *listing* or *recording* them on a sheet of paper; and, at the same time, *deferring* evaluation, or not judging them until later. Brainstorming and other problem-solving methods, then, can be understood as "molecules" made up of smaller "atoms." These atoms, or heuristics, can be used by themselves or recombined into many other methods.

Listing, by itself, for example, is a very powerful heuristic. It's the basis of "to do" lists and shopping lists. It's a good way to get ideas out of your head so you can remember them and don't have to keep repeating them to yourself. (As we'll see in Chapter 6, that's why it's so important to record ideas on chart pad paper during meetings—because then people can stop repeating their ideas to each other.)

Each heuristic has many advantages, like any tool. Each also has disadvantages or limitations. While a hammer is great for pounding nails into wood, it's not useful for putting screws in wood, except maybe as a way to get them started. While listing can be useful, it's also sometimes helpful to let an idea germinate—to not express it too quickly.

For my research, I took the list of problem-solving methods and broke these "molecules" into their heuristic "atoms." I gave each heuristic a label or tag—typically an action verb. Wherever possible, I paired each heuristic with its opposite. These heuristic pairs included, for example: "working forward/working backward," "building up/eliminating," and "leaping in/holding back." While my list of problem-solving methods kept getting larger, my list of heuristics grew to sixty-four and then stopped. I decided, using this very simple, unsystematic, and heuristic process, that *there exists a limited set of core heuristics, about sixty-four in total, out of which all of the more complex problem-solving methods can be built.* (A complete list of sixty-four is included in the Resources section at the end of the book.)

## A Pragmatic Theory About Learning Problem-Solving Skills

So in 1969 I came to what was for me a startling and yet reassuring realization: If human problem solving is heuristic and there are a limited number of heuristics, there must be a link between one's repertoire of heuristics and one's ability to solve problems. I began to develop a pragmatic theory about learning and teaching problem-solving skills. It's presented visually in Figure 3.

This theory of learning starts with the premise that (1) your ability as a self-confident, flexible problem solver is dependent on

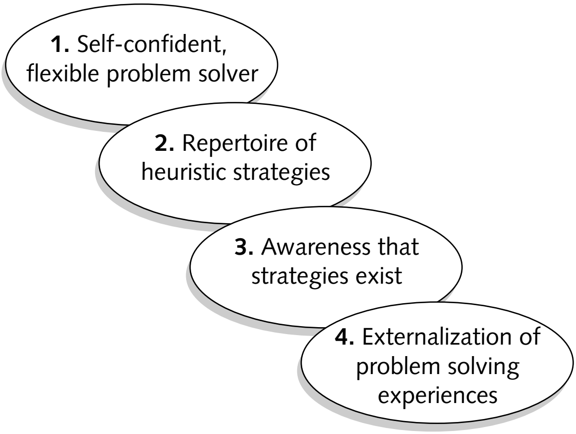


**Figure 3:** A theory about how humans learn problem-solving skills

(2) the repertoire of heuristic strategies that you have learned and are able to apply in a variety of situations. In order to increase your repertoire of heuristic strategies, you must (3) gain awareness that these strategies even exist. And one important way to become aware is by (4) externalizing your problem-solving experiences—that is, thinking out loud—and watching other people externalize theirs. Now, let's look at these elements of the theory in turn, starting with the second one.

### Your Repertoire of Heuristic Strategies

The more heuristics you know, the more effective and creative you will be as a thinker. This is the tool-user analogy: The more tools a carpenter has in his or her toolkit and is able to use skillfully, the more he or she will be able to deal successfully with different types of woodworking problems. The more heuristic thinking tools you have learned and know how to use, the more creative and productive you will be.

## Awareness: The Key to Building Your Repertoire

To increase both the number of heuristics in your tool kit and your ability to use them effectively, you must increase your awareness. Just becoming aware that heuristics exist can help you to demystify individual as well as group problem solving. You can make an inventory of the heuristics you already know and consciously begin to learn new ones. Furthermore, process awareness allows you to become "consciously competent"—to assume control over your choice of problem-solving strategies when necessary.

To help you understand the importance of process awareness, let's consider some sports analogies. Think about playing tennis. Most of the time, when you're playing your best, you're not thinking about your strokes. You are just playing—you're "in the zone"—you're playing with unconscious competence. If your opponent keeps hitting the ball hard to your forehand, however, making you run to reach the shots, and if the only forehand stroke you know is one with topspin, you may have trouble returning these shots with any control. If you have other strokes in your tool kit, however—an underspin slice, for example—then you may be able to handle the pace of the shots and have more control over your returns. But if you don't know the difference between topspin and underspin—if you aren't even aware that there *is* a difference—then you won't be able to make this adjustment in your game. To become a better tennis player, then, you must spend time learning about different kinds of shots—becoming aware of them. Then you must practice them in noncompetitive situations. The more you can groove a stroke so that you don't have to think about it, the more available it will be to you at a crucial moment in a game.

Awareness is critical not only for acquiring new shots, but for keeping yourself in control when things start to go wrong. Watch professional athletes when their game begins to slip. You will see them talking to themselves, consciously trying to analyze what is going wrong and correct it. Tiger Woods, the great young golfer, does this all the time. If he feels his swing is a bit off during a round, he goes completely within himself and works to correct the problem. He does this even if he's winning by a good margin. He'll take practice swings between every shot, stopping the club at the top of the backswing or at the point of impact, to see if his body is properly aligned or if the angle of the club is correct. He often mutters to himself in the process. This constant, conscious awareness, analysis, and correction of small errors is one of the reasons Woods has developed arguably the best swing in golf—and why he has won so often.

Similarly, you need an awareness of your problem-solving process in order to acquire new heuristics and to learn how to become a better problem solver, in general. You also need to be aware of process when you get stuck so you can consult your repertoire of strategies and consciously select a new one. Process awareness is essential for breaking fixation and handling difficult strategic moments.

Process awareness is also essential because we tend to favor certain sets of heuristics based on our personalities. In fact, we often describe each other by our most often-employed strategies. "He is such a *planner*." "She is so *systematic*." "He is always *leaping in* before thinking." Each heuristic in a pair may require the adoption of a different attitude or emotion on the part of the problem solver in order to implement it. Take the heuristic pair of *leaping in* versus *holding back*, and think about how different people learn to use, say, a new remote control for a TV set. Some people leap in and start pushing buttons to discover what each button does. Others hold back and insist on reading the directions before attempting to use the remote. It takes an awareness of our natural process bias in order to consciously choose a different and perhaps uncomfortable approach.

## Externalization and the Need for a Common Language of Process

Finally, step four in this model of human problem solving deals with the importance of *externalization*, or thinking out loud. Externalization is the key to learning and teaching individual and group problem solving.

Think about a time when you learned from someone a new way of approaching a problem. This probably didn't happen just by watching someone work. Watching someone brilliantly think up a new solution to a problem that was stumping you just may have made you feel stupid. More likely, you experienced a key learning moment when someone "thought out loud" in front of you, sharing his or her strategies while working heuristically on a problem. That allowed you to observe the person's mind dancing around a problem, trying a particular strategy, seeing whether it worked, and then trying something different.

My best learning moments at design school occurred when professors came to my drawing table and, rather than just criticizing my work or suggesting a change, designed out loud in front of me. I could observe how their minds worked and how they implemented their strategies through drawing. They might have said something like, "Those two spaces look a bit awkward next to each other. How could we arrange them differently? Let's make a diagram of the main circulation flow. To do this, let's follow a first-time visitor to this building and ask ourselves which spaces we would want them to see, in what order." From this kind of dialogue, I learned about the power of *simplifying, diagramming,* and so forth. And the next time I was stumped I would ask myself, "How would the professor approach this problem?" I began to internalize the voices—and the problem-solving methods—of my professors.

If you don't have the words to describe an experience, it's often hard even to see or observe it. Without a mental framework, or what Neisser calls a cognitive structure, it's difficult to capture and retain related information. In the example of bird identification, having names for species and subspecies helps you distinguish and identify different birds and organize, retain, and access the information you learn about their songs, habitats, flight patterns, and so forth. In the same way, being able to attach a specific name or label to a heuristic strategy allows you to organize and access the information and experience you gather about its powers and limitations.

For example, let's take the pair of heuristic strategies I call *working forwards* and *working backwards*. Working forwards involves starting with what you know and building forwards, step-by-step—for example, writing a book by beginning with the introduction. The strategy of working backwards involves jumping to where you want to end up and building backwards—for example, writing the book's conclusion first, and then figuring out what chapters are needed to build to that conclusion. These two very powerful strategies can be applied in any situation. And you are much more likely to access and use each of these strategies if you learn a general, context-independent term to describe it. If, as so often happens in school, you were only exposed to these heuristics as part of a writing course, and the heuristics were never named, then you might be less likely to be able to apply them somewhere else, such as in a math class or a business situation. So, having a language and a vocabulary to describe various processes is very important for building your personal repertoire of problem-solving strategies.

## Relevance to Collaborative Problem Solving

Hopefully, this chapter has helped you understand more about how individuals solve problems and how you can become a more confident and effective problem solver. These concepts form the basis for everything that follows in this book and will be periodically mentioned again. Before we go on, however, let's look at the major lessons from this chapter and discuss how and why they relate to *collaborative* problem solving.

- **Problem solving is heuristic**. There is no one right way to solve problems. Likewise, there is no one right way to collaborate. At best, collaborative problem solving is an educated trial-and-error process. This is an important realization for groups that get mired in fighting over the right way to approach a problem. Group members must learn that it's more productive to simply select one problem-solving approach and see if it works. If it does not, they can try another. As we'll discuss in Chapter 5, it's the facilitator's job to help a group make conscious choices about which approaches to use in the course of a collaborative process. Thus, facilitators must have command of a whole tool kit of problem-solving strategies.

- **It's important to recognize strategic moments**. A group, like an individual, can get stuck and become fixated. The strategy it has been using just isn't working. The group needs to stop and make a conscious choice about what heuristic strategy it's going to use next. Again, it's the facilitator's responsibility to recognize these strategic moments and help the group make these important process decisions.
- **Problem-solving skills can be learned**. Just as with individuals, a group's problem-solving skill is dependent on the repertoire of problem-solving tools it knows how to use. A group can increase its ability to tackle difficult problems either by consciously acquiring new tools in formal training programs or through just-in-time learning, whereby the facilitator, or someone else in the group, suggests a new problem-solving process at the appropriate moment.
- **Having a common language of process is crucial**. In collaborative problem solving, it's absolutely essential that a group have a common language of process. For example, when a facilitator recommends that a group use the problem-solving method of brainstorming, everyone in the group must understand what brainstorming is and how it works. Effective collaborative problem solving requires members of a group to be able to communicate and agree on common processes, moment by moment. Likewise, an

organization needs a common language of process to work effectively and collaboratively across organizational units. For example, we've found that without a common and clearly understood language for strategic planning, people in different parts of an organization will have different definitions for commonly used terms such as *goals, objectives, strategies,* and *tactics.* And if you are trying to build collaboration between organizations or within a community, all the stakeholders must agree on a language of process in order to be able to design a common way of working together.

*Solution to the anagram puzzle: "BETRAYAL."