3

The 4S Method

How can we overcome the pitfalls described in the previous chapter to be better problem solvers and solution sellers? In this chapter, we'll introduce a process that can help. We call it the 4S method because it consists of four stages—State, Structure, Solve, and Sell. We'll explore these stages in Chaps. 4, 5, 6, 7, 8, 9, 10 and 11, and we'll apply the method to an actual case in Chap. 12.

Where Does the 4S Method Come from?

Before we introduce the 4S method, we'll explain its origin. If you're only interested in learning about the method instead of where it comes from, you can skip this section.

PSAC: The Problem-Solving Approach of Consulting

The core of the problem-solving method we present here has been developed and refined over many years by McKinsey & Co, one of the oldest and most respected strategy consulting firms, and emulated by other consultancies, such as Bain and the Boston Consulting Group.[1] The Problem-Solving Approach of Consulting (PSAC) isn't usually taught outside of consulting firms, and the ability of strategy consultants to "crack" tough business problems in unfamiliar fields is a large part of the consulting industry's mystique. Perhaps because of this mystique, some components of PSAC have found their way into everyday business lingo. Many executives are familiar with

B. Garrette et al., *Cracked it!*, https://doi.org/10.1007/978-3-319-89375-4_3

terms such as "hypothesis-driven problem solving" or "MECE" problem decompositions (which we'll define in Chap. 5). University and business school students who practice "case cracking" hoping to impress consulting-firm interviewers also strive to become familiar with these tools, sometimes at the risk of focusing on isolated parts of the problem-solving approach without grasping its overall structure.

As its origin suggests, PSAC is a practical approach, and rarely makes its intellectual underpinnings explicit. These foundations are based on Aristotelian logic and Cartesian method and on their modern incarnations, including the practice of "critical thinking" taught in many universities. PSAC is grounded in pure logical reasoning: for instance, it assumes that facts are part of an objective reality, and that all honest observers will agree upon that reality once presented with the evidence. It assumes one thing can't be true if its opposite is true. It also presumes that causal links between events can be established and verified, and that the necessary and sufficient conditions for a proposition to be true can be identified (we'll discuss these in Chap. 5). These are principles with which the overwhelming majority of executives agree. Some epistemologists and cultural anthropologists may object, but, in our experience, very few CEOs do.

Logical thinking is a powerful tool in any setting, and business is no exception. By pushing their clients to formalize their reasoning, by rigorously testing the links in the causal chain of that reasoning, and by insisting on evidence to back up each link in that chain, strategy consultants—or any skilled user of formal logic—can sometimes overturn preconceived ideas or challenge accepted practices. Often, it's possible to conclusively prove, using logic and facts, that one course of action is inappropriate, or (less often) that one recommendation is demonstrably the best. Logic is a powerful way of communicating conclusions to a rational audience. This idea is reflected in the "pyramid principle" of communication, which we'll present in Chap. 10. In sum, PSAC is powerful, pragmatic, and tested. It provides a sound foundation for a general-purpose problem-solving approach.

PSAC does, however, suffer from some limitations. Two of these limitations have been the focus of serious challenge from other schools of thought. We'll briefly examine each.

Hypothesis-Driven Thinking and Cognitive Science

One of the tenets of formal logic is the concept of formulating and testing hypotheses. It is how scientists conduct scientific research: develop a hypothesis, grounded in theory; design an experiment to test it; and subject

the results to challenge and debate with peers. PSAC borrows a page from the scientist's handbook and suggests that the method of choice to move the problem-solving effort forward efficiently is to formulate hypotheses, then test them. This approach is logically defensible and highly efficient: by zeroing in quickly on a possible solution, consultants avoid the painful process of exploring all the possible (but unlikely) answers, a trap they deride as "boiling the ocean." This is "hypothesis-driven problem solving," an approach that newcomers to consulting firms (and their clients) initially regard as disconcerting, but quickly find addictively powerful.

There is, however, a difficulty that practitioners of hypothesis-driven problem solving must overcome. Logic dictates you should be able to test hypotheses independently of the way they're formulated: it should not, in principle, make a difference whether you hypothesize that "this product will sell more than $100 million" or "this product will sell less than $100 million." Regardless of the formulation, the same evidence should lead you to the same conclusion. Proving a hypothesis and disproving its opposite are logically equivalent and should be practically identical.

Unfortunately, this neutrality is very difficult to achieve. As Francis Bacon put it in 1620, "Man prefers to believe what he prefers to be true."[2] William James echoed this when he wrote, "A great many people think they are thinking when they are merely rearranging their prejudices." This is what cognitive scientists call the *confirmation bias.*[3] Because of confirmation bias, we are much more likely to seek evidence that confirms our hypotheses and much more likely to believe such evidence once we find it, than we are to search for, and pay attention to, *disconfirming* evidence. The confirmation bias explains, for instance, why voters who support opposite parties can watch the same political debate and confidently conclude that "their" candidate has won: they unconsciously pay more attention to the points their preferred debater has scored and ignore the ones made by her opponent. This can severely distort our assessment of facts and mistakenly lead us to conclude our hypothesis was supported.

Consultants aren't immune to this bias, but they've developed at least three safeguards against it. First, as outsiders, they have, in principle, no vested interest in the recommendation: there may be a hypothesis on the table, but it's not *their* hypothesis. Being neutral doesn't eliminate confirmation bias, but being politically or financially biased would make it worse. Second, consultants work in teams and are trained to challenge one another. McKinsey's core values, for instance, include a "non-hierarchical atmosphere" and the "obligation to dissent." These guiding principles help ensure that a team member blind to his own confirmation biases will be called out by a colleague.

The spirit of collaboration that such social norms foster is an integral component of the problem-solving apparatus. Finally, consultants learn and practice PSAC continually. Rookies are taught that their task, once a hypothesis is formulated, is to work diligently to *either prove or disprove* it, with no preconceived idea of what the answer will be. Experienced consultants are regularly reminded of this. These safeguards don't guarantee that consultants won't fall in love with their own hypotheses. But the risk is much greater when the safeguards are removed. Our experience shows that when corporate executives attempt to apply this method in their companies—because they are former consultants or they've been trained by them—they often struggle with the challenge of fighting confirmation bias and its ramifications.

Consider, for instance, the case we introduced in the previous chapter involving Danone and Grameen, and put yourself in the shoes of an executive tasked with implementing the vision of the two CEOs. Can you see the flaw in the candidate solution? Maybe … but the odds are stacked against you. First, you're probably not a member of that team by accident: as a longtime employee selected for this high-visibility project, you may also believe in the vision of fresh dairy products as the remedy for child malnutrition. Second, it doesn't take a Machiavelli of corporate politics to realize that the CEOs' vision isn't a mere "hypothesis" to be skeptically challenged. In most organizations, there is no such thing as "a hypothesis to be proven or disproven"—proposals have proponents, precedents, and histories, and those who evaluate them know it. Third, as someone tasked with implementing the vision your bosses have put forward (and achieving its life-saving benefits), you're under considerable pressure, a condition that reduces your ability to think creatively.[4] Fourth, if you disproved the hypothesis, what would you do next? You'd have no report to write, no action to recommend, and perhaps no way of achieving the objectives assigned to you. Your criticism of the solution might even be regarded as an attempt to find excuses for not reaching the new business' growth targets. Finally, even if you were the lone skeptic on a team of true believers, you'd probably quickly decide that it's not your job to challenge other team members' confirmation biases, and just "go with the flow" despite your doubts—a powerful phenomenon sometimes called *groupthink*.[5]

The upshot is that hypothesis-driven problem solving is a powerful tool—so powerful it can be dangerous. It is, by design, a risky method that pushes us in a direction we are already prone to take. Outside of consulting firms—and sometimes within them—there is a real risk that it will lead to some of the pitfalls we described in Chap. 2, especially the "solution confirmation trap."

This is a difficult challenge to overcome, but there is an antidote—which is also part of PSAC. Alongside *hypothesis-driven* problem solving, many strat-

egy consultants practice *issue-driven* problem solving. This version of logical problem solving eschews the formulation of hypotheses, and treats problems as "open questions." This isn't a guarantee against confirmation bias: your hypotheses may creep into your definition and structuring of the problem, leading you to define your problem narrowly, which constrains the solution. This is the "narrow framing trap" of Chap. 2. However, when you avoid the explicit formulation of hypotheses, you prevent some of the worst cases of the "solution confirmation pitfall." Therefore, this approach, while more difficult to master, is often preferable. We'll discuss in Chap. 5 when to use an issue-driven approach, and when it's safe to rely on the more expeditious hypothesis-driven approach.

Solution Generation and Design Thinking

The second important limitation of PSAC is that some problems just seem to resist it. Some critics of the consulting approach (and the consulting profession) see this as the result of a lack of creativity in the problem-solving method (and in those who deploy it). PSAC, they claim, is a mechanical process, routinely applied by like-minded consultants who come from the same backgrounds, wear the same gray suits, and produce the same uninspired, formulaic solutions. No wonder the flashes of insights that produce novel solutions never seem to emanate from consulting firms.

This criticism is, in our view, misguided—or at least too broad. While there are brilliantly creative consultants, many problems don't require much creativity. Although contemporary business culture sometimes seems to view innovation as an absolute good to be pursued at all costs, creativity in problem solving isn't universally desirable. We expect experts such as doctors, air traffic controllers, or auto mechanics to identify and solve problems, but we don't expect them to be creative. Sometimes, creativity is even illegal, as the sarcastic phrase "creative accounting" suggests. Much business problem solving consists of identifying tried-and-tested solutions to complex problems, not in finding novel solutions to them. Experienced practitioners of PSAC call this temptation "reinventing the wheel": the tendency to look, at all costs, for a new, out-of-the-box answer to a problem that has an acceptable off-the-shelf solution.

Critics of PSAC, however, have a point: some problems don't lend themselves to being disaggregated and solved by logic and facts in the linear way consultants prescribe. Some problems are hard to state precisely, because problem solvers don't understand them well enough. Some are hard to structure

logically because problem solvers don't understand their causes. And some are hard to solve using facts alone because they require active idea generation. This is likely to be the case when the problem is complex and poorly understood, and when the solution must be designed for and used by people. The challenge of redesigning the customer experience at J.C. Penney, described in the previous chapter, is a good example of such problems.

Over the past 20 years, the school of thought known as "design thinking" has proposed a compelling approach to address problems of this kind. While various flavors exist, the core process is most closely associated with the Silicon Valley design firm IDEO and the Hasso Plattner Institute of Design at Stanford University (commonly known as the Stanford d.school). Design thinking has emerged as a powerful problem-solving toolkit that integrates both creative and analytical approaches. Although developed for the design of physical artifacts, design thinking has evolved to address intangibles such as services, processes, and larger organizational systems and strategies.[6]

Why has the reach of design thinking expanded? There are a few reasons. One is its growing use by a variety of organizations, including for-profit, nonprofit, educational, governmental, and non-governmental. Reflecting its broad applicability, problem solvers in these organizations are finding new and useful applications of human-centered design beyond material artifacts. For example, the Designing Out Crime research center, a partnership in Australia between the New South Wales Department of Justice and the University of Technology Sydney, uses design thinking to help solve complex crime and social problems. A related reason is that design thinking works—it helps problem solvers tackle highly complex and poorly understood problems in ways that more traditional analytical methods don't. The intensive use of design thinking by companies also improves their financial performance. The Design Management Institute's Design Value Index shows that a stock portfolio of 16 design-centric companies outperformed the Standard & Poor's (S&P) 500 by 228 percent over the period 2006–16. Many organizations have established design thinking service units and some, such as IBM and Intuit, are reinventing their cultures and operations around design thinking. Management consulting firms are also investing in this capability by acquiring design services firms. In the past few years, Deloitte bought Doblin, Accenture acquired Fjord and 2nd Road, and McKinsey acquired Lunar.

Design thinking is concerned with how people experience human-created artifacts because these artifacts represent solutions to problems. At their most basic, problems signal unsatisfied needs and wants. They arise from dissatisfaction with existing solutions. In *The Sciences of the Artificial*, his landmark book on problem solving and design, Nobel laureate Herbert Simon

argued that design is concerned with how things ought to be and with devising artifacts to attain goals.[7] Designers devise courses of action aimed at changing existing situations into preferred ones by deliberately creating new artifacts as solutions. For a problem to be solved, a solution must be designed.

From a logical and philosophical standpoint, design thinking is distinct from the other approaches we've discussed. Hypothesis-driven and issue-driven problem solving are both forms of *deductive reasoning*. Both approaches require the problem solver to have a theory—an understanding of cause-and-effect relationships—about the general causes of the problem being solved. In contrast, design thinking is a form of *abductive reasoning*. As defined by the American philosopher Charles Sanders Peirce, abduction takes place when you use a limited set of observations to generate the most plausible and parsimonious explanation for the data—an explanation which may be incorrect, and must be tested and validated. Design thinkers do this when they suppress their assumptions about the problem, and instead, use insights generated from observations of users to develop hypotheses about solutions. These hypotheses are then iteratively tested in the form of prototypes to converge toward the best-fitting solution. As Roger Martin argues in *The Design of Business*, "Designers live in Peirce's world of abduction; they actively look for new data points, challenge accepted explanations, and infer possible new worlds."[8]

Design thinking is a powerful approach to solve *some* problems, but not *all* problems (just like the traditional PSAC method is effective in many, but not all situations). The key is knowing which approach to select and when. The 4S method provides a guide.

An Overview of the 4S Method

Figure 3.1 summarizes the 4S method. The flowchart may look complex at first, but it's easy to follow. It combines the approaches we introduced above in a pragmatic manner. It will help you decide which of three possible paths to take to solve a problem: the hypothesis-driven, issue-driven, or design thinking path.

A simple example will help illustrate this reasoning.

A new CEO—let's call her Tracy—has been appointed at Solar, a multi-business, family-owned industrial company that sells packaging products to large corporate customers. Over the years, Solar acquired firms that

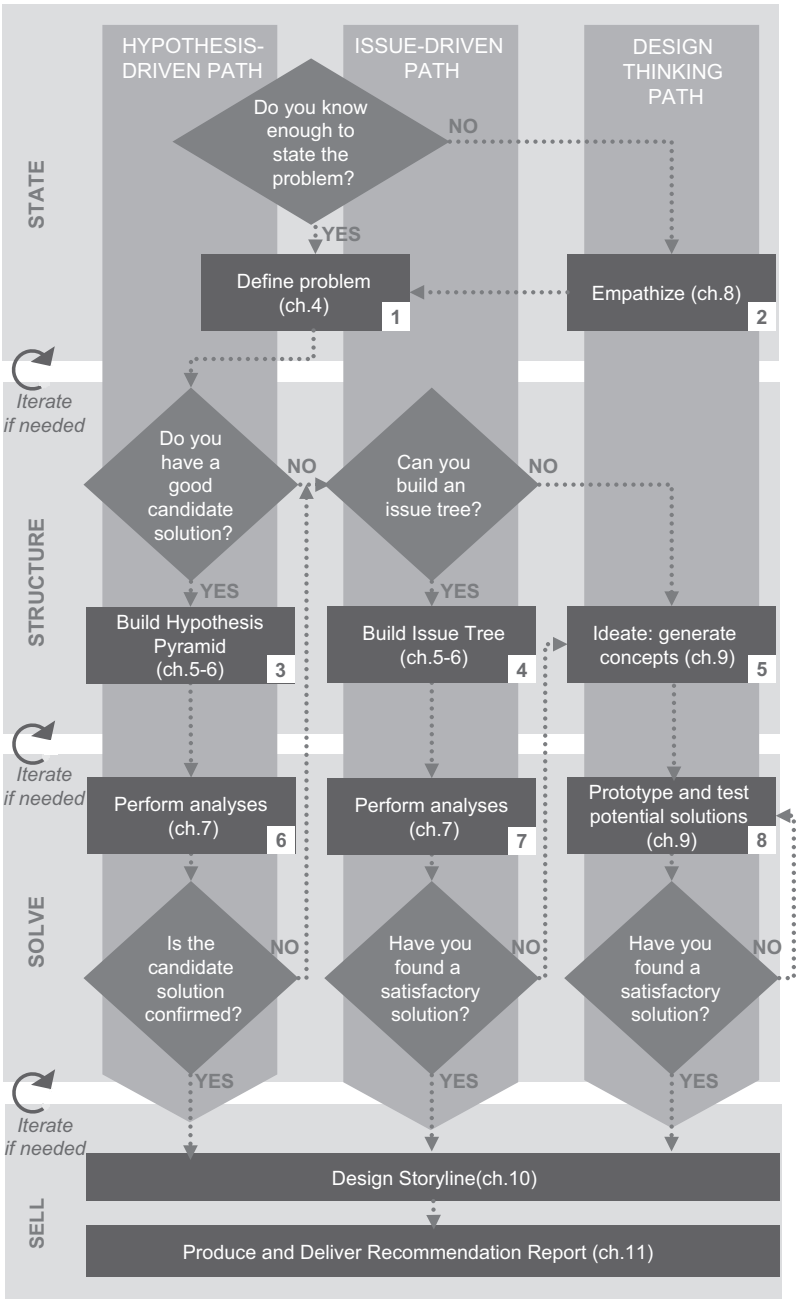

Fig. 3.1 The 4S method

specialize in different packaging technologies to provide a full range of packaging solutions to its corporate customers—a classic example of customer-centric diversification. When Tracy joins, the corporation is profitable overall, but two of its business units—Pluto and Uranus—are in the red. Members of the management team have differing viewpoints about the money-losing businesses. Several of them believe that their problems are due to a temporary decline in market demand, and believe the problem will resolve itself as soon as the economy improves. A few claim that the problems are purely operational, and assert that better manufacturing effectiveness could restore profitability. Others, however, believe such a turnaround to be impossible: they suggest selling or even closing Pluto and Uranus—despite the risk of labor unrest associated with such a move.

Tracy must decide on the most appropriate course of action (a problem-solving task) and sell the solution to the Board (a communication task). How could she think about the steps in that process, using the 4S method?

State: A Problem Well Posed Is Half-Solved

The importance of a good problem statement can't be overstated. Einstein said, "If I had an hour to solve a problem, I'd spend 55 minutes thinking about the problem and 5 minutes thinking about solutions." As we saw in Chap. 2, the "flawed problem definition" is a pitfall problem solvers should be aware of.

The first step for Tracy is to state the right problem. At this stage, Tracy has a clear question ("What to do with Pluto and Uranus?"), a set of symptoms (losses), and even some proposals for possible answers. This may seem like it's enough to state the problem. But it's not. Many more elements are needed for a complete problem statement. The first question may not be "What is the problem?" but "Do I know enough to state the problem?"

In Tracy's case, what don't we know, but need to know to define the problem? For instance, we don't know the extent of the bleeding. The problem isn't the same if the losses of Pluto and Uranus threaten the viability of the corporation and are an urgent problem, or if they are only a minor concern. Likewise, we don't know what Tracy's objectives are: perhaps the family owners expect her to create maximum shareholder value, even if it means shrinking the company, or maybe they're more concerned about keeping the corporation intact for reasons of social responsibility, prestige, or other considerations.

These are just examples. As we'll see in Chap. 4, writing a full problem statement requires you to examine five elements of the problem—abbreviated in the acronym TOSCA (Trouble, Owner, Success criteria, Constraints, Actors). If Tracy tries to write a full-fledged problem statement (Box 1 in the flowchart in Fig. 3.1), she'll find she doesn't know enough about the problem yet.

This realization should lead Tracy to discover more about the problem. There are two ways to do that. Sometimes, basic data gathering and rigorous thinking may be sufficient to define the problem properly. In this case, some data on recent business results are needed. Tracy will want to spend time at the production sites of Uranus and Pluto, meet with customers, and talk to board members about their objectives. This is the minimal level of inquiry required to state almost any problem of reasonable complexity.

Frequently, however, this won't be enough to develop a full problem statement. It will then be valuable to borrow from the design thinking toolkit and use techniques known as *empathy* (Box 2). In the design thinking path, a problem solver invests in *empathizing* with users by observing them, engaging with them, and immersing herself in their situation. This helps her discover their needs and learn how they experience the problem: how they think and feel about it, the context in which they experience it, and the constraints they face. During this phase, problem solvers develop rich insights about users, which they can use to state the problem better, reframing how they understand it by viewing it from different users' perspectives. We'll explain these techniques in Chap. 8.

Structure: The Architecture of Problem Solving

As soon as we state the problem**,** it's tempting to list actions Tracy could take to solve it immediately. Not surprisingly, this is what members of her executive team propose. Business people have knowledge and judgment, and they typically look at situations through the lens of their experience.

Tracy could decide that one idea her colleagues propose is an attractive candidate solution, and treat that idea as a hypothesis to be tested. That is, she can adopt a *hypothesis-driven* approach (pictured in the leftmost column of the flowchart).

The hypothesis-driven approach starts from an idea—a hypothesis—about what the solution might be and then tests it. Suppose, for instance, that Tracy hypothesizes that Solar should sell Pluto and Uranus. Logically, for this hypothesis to be true, many things must be true. For instance, the unit to be sold must be readily separable from the rest of the business, there must be a

buyer at an acceptable price, and so on. If she chooses this approach, Tracy should list these conditions, and disaggregate them into smaller requirements. This is what we'll call a "hypothesis pyramid" (Box 3 on Fig. 3.1). Tracy and her team can then move to the "solve" stage to perform the analyses to test all these hypotheses.

However, as we've seen, the hypothesis-driven approach increases the risk of confirmation bias and exposes you to the solution confirmation pitfall. Before using this approach, Tracy must ask herself whether she has good reason to be *highly confident* in a given hypothesis. In our example, this doesn't appear to be the case. The fact that different colleagues have radically different hypotheses should give Tracy pause.

In such a situation, the alternative is to go for an "issue-driven" path, following the second column in the flowchart. An issue-driven approach requires you to break down the problem into smaller components with an *issue tree* (Box 4). An issue tree is a way to structure the problem thoroughly, to look at its different facets systematically, without a preconceived idea of what the solution might be. The benefit of this approach is that you can avoid the solution confirmation pitfall.

The downside is that building an issue tree is more difficult and more time-consuming than building a hypothesis pyramid. In Chaps. 5 and 6, we'll introduce techniques that will help you overcome this challenge, and especially the role of *frameworks* in building issue trees. Frameworks are essential shortcuts for someone building an issue tree because they provide pre-packaged decompositions of typical, recurring business problems. For example, Tracy's issue tree may include a branch that asks whether Uranus operates in an attractive industry: that is a question that Porter's five forces framework[9] can help address. Likewise, if Tracy must analyze the business portfolio of Solar, a strategic business portfolio matrix can be useful. Frameworks enable you to think faster. However, as we'll see in Chap. 6, this convenience has a cost: frameworks also constrain your thinking. A framework encapsulates a theory to solve a class of problems, and to use a framework is to espouse a theory and the assumptions that underlie it.

If you succeed at decomposing the problem and turning it into an issue tree, then you can move onto the "Solve" phase and analyze the issues. You may, however, struggle to build an issue tree. Unlike Tracy's problem, which she could readily turn into an issue tree (we'll show how in Chap. 5), not all problems are amenable to decomposition. Solving a product design problem, imagining a new advertising campaign, or designing a bold new strategy for an ailing company are tough business problems. But you'll probably find them difficult to break down with a hypothesis pyramid or an issue tree.

Instead of a traditional decomposition, these kinds of problems call for an *ideation* phase (Box 5). Design thinking practitioners use what they learned about *what is* (in the "State" phase) to then imagine *what should be*. They begin this effort by understanding the problem space. They then build on this understanding to synthesize an essential set of *design imperatives* that an effective solution must address. These imperatives represent the benefits a solution must provide users. They serve as a guiding vision for the ideation, prototyping, and testing phases. Together, the imperatives constitute a model of what should be, guiding the search for solutions. Once imperatives are established, designers generate a wide variety of ideas for solutions.

Solve: Between Analysis and Creativity

We've now seen three routes Tracy can follow to get to the "Solve" stage: hypothesis driven, issue driven, and design driven. The next steps will depend on her entry point.

Let's first assume that Tracy adopted a hypothesis-driven approach. Her next task is to test her hypothesis with the requisite analyses (Box 6). If, for instance, she hypothesizes that Uranus should be sold, she'll need hard facts to back up each statement into which her central hypothesis breaks down. Two outcomes are possible:

- Tracy may reject the hypothesis altogether if she finds clear evidence it was wrong. In that case, she'll naturally be tempted to formulate a different hypothesis: "If X doesn't work, let's try something else." Given the dangers of hypothesis-driven reasoning, however, that is a risky approach. A better strategy is to switch gears and adopt an issue-driven approach (moving to the middle column in the flowchart). Sometimes, it may even be necessary to return to the first stage in the problem-solving process and revisit the problem statement.
- The other possibility is that the hypothesis will be confirmed. Wholesale confirmation is, however, rare. More often, analyzing the hypothesis will cause changes or refinements to it. For instance, Tracy may conclude that Uranus should be sold, but Pluto shouldn't. The question Tracy must answer eventually is whether this resulting, refined hypothesis meets the success criteria she specified in her initial problem statement. If it doesn't, she'll consider the hypothesis rejected, with the same consequences as in the previous paragraph. If the resulting solution meets success criteria, she'll proceed to the "Sell" stage.

The second case we must consider is the one where Tracy chose the issue-driven route from the onset. She, therefore, enters the "Solve" stage with an issue tree that breaks down her problem into discrete parts to be analyzed separately. For instance, her issue tree may include the question, "What are market prospects for Uranus?" This question calls for a specific piece of analysis, probably using market research. It may ask, "What would be the value of Uranus as a stand-alone entity?" This question calls for some financial valuation work. In general, an issue tree will lead you to do all the same pieces of analysis as a hypothesis pyramid, and more: unlike the hypothesis pyramid, which zeroes in on the key analyses that prove or disprove your hunch, a good issue tree leaves no stone unturned. Box 7 in the Fig. 3.1 flowchart shows the step of conducting these analyses.

All these analyses may point Tracy toward a solution. Suppose, for instance, that the data unambiguously show that Pluto's problems are caused by declining quality. It takes little effort to propose that Pluto needs a quality improvement program. Once proposed, Tracy will ask herself whether it meets the success criteria she had predefined; if it does, she'll need to sell the program to her board.

However, with an issue tree, there is no guarantee the solution will just emerge like this. Issue trees disaggregate problems, but not all problems can be solved by being disaggregated. For instance, disaggregation may reveal that Pluto's difficulties stem from growing customer dissatisfaction, resulting from unattractive products, changing preferences, and the launch of superior competitive products. In such a case, the issue tree yields a complete diagnosis, but doesn't, by itself, suggest a solution. What can Pluto do to make its products great again? You won't answer this question by splitting the problem into smaller problems. You need fresh ideas.

Fresh ideas are precisely what the design thinking approach can generate. If the problem-solving process gets "stuck" on the issue-driven path, it may be useful to move back from Box 7 to Box 5 in the flowchart. Generating solution imperatives and solution ideas will allow Tracy to propose how Pluto's offering can be improved.

Once ideas are generated, they must be tested. This is when the "designer" (i.e., the problem solver on the design thinking path) leaves the realm of abstraction and returns to the concrete world to translate ideas into tangible prototypes to be tested with real-world users. In this *prototyping and testing* phase (Box 8), designers choose promising potential solutions and represent them in tangible form so users can interact with them. Prototypes embody designers' hypotheses about desirable solution characteristics, which are then tested by users in the final phase. User feedback about prototypes helps designers choose

the final solution for implementation. Tim Brown, CEO of IDEO, summed it up this way in his book *Change by Design:* "The mission of design thinking is to translate observations into insights and insights into products and services that will improve lives."[10] Ultimately, one or several solutions should emerge that satisfy the problem statement's success criteria. If that doesn't happen after a reasonable amount of trial and error, a rethink of the problem statement may be in order.

Some proponents of creative problem solving might argue these last steps are the only valuable ones, and that building the issue tree was just a waste of time. We disagree. First, when Tracy started building the issue tree and analyzing issues, she had no way of knowing whether an issue-driven approach would generate an acceptable solution (but as we showed, it often does). The analysis Tracy conducted based on her issue tree informed her thinking. It narrowed down the solution space for her creative search: if the challenge is to make the product great again, she'll get creative about that, not about boosting employee motivation. And the analysis may have planted the seeds of ideas that will be helpful in the creative stage. Rather than substitute for each other, the analytical and creative approaches are complements.

To summarize, Tracy has three paths to a potential solution, which form the three columns in the flowchart:

- She can have a hypothesis from the beginning and test it to confirm its validity.
- She can start without a hypothesis, and find after she disaggregates the problem with an issue tree, that a viable idea emerges from her analysis.
- Or she can generate ideas using specific ideation techniques, because she couldn't build an issue tree or because the issue tree she built didn't generate a solution.

Sell: Choose the Approach That Suits Your Audience

Regardless of the path taken, Tracy has now settled for a solution she believes is the best. Her next task is to convince the Board to approve her plan. It is time to switch gears from problem-solving mode to communication mode.

As we mentioned in the discussion of the "miscommunication pitfall" in Chap. 2, this switching of gears raises an important question. If communication

is a different exercise from problem solving, why are we covering both in this book? There are innumerable books and training programs on how to communicate your ideas effectively that don't cover the problem-solving stage, and some are excellent.[11] So why not just stop at a "3S" method? The pragmatic answer is that almost anyone who reads this book looking to become more effective at solving problems will also need to sell the solution, which makes a "one-stop shop" approach appropriate. More importantly, while solving a problem and selling the solution are distinct stages, they should be integrated for at least two reasons.

First, business problem solvers usually can, and should, interact with their audience throughout the problem-solving process. In our example, Tracy must talk to the board not only when she has found a solution, but from the very beginning of the problem-solving process, to state the problem correctly. She may well involve board members in the "Structure" and "Solve" stages of the process too. It's likely she'll give the board interim progress reports in which she presents emerging findings, shares ideas on possible solutions, and gets feedback about the board's thoughts and concerns. In short, her communication task starts at the very beginning of the problem-solving process. Sometimes, a fully baked solution must be sold to a "virgin" audience—for instance, when an advertising agency pitches a campaign to a new client. But communication approaches that are based on this premise are not generally appropriate for a problem solver within an organization.

There's a second reason to consider communications in the problem-solving phase, and vice versa: it ensures the two are sufficiently distinct and communication concerns don't "contaminate" the search for a solution. Finding the best solution is one task, selling it is another, and shouldn't start before you're sure the problem is solved. With the fat vs. sugar debate we described in the previous chapter, Ancel Keys did exactly the opposite: because he thought he knew the solution, he didn't use his empirical work as a problem-solving device to test his hypothesis. Instead, he used it as a sales pitch to convince decision-makers. This aggravated the analytical mistakes he made in the Seven Countries Study, by leading him to focus on countries that supported his view and disregard those that could have proven it wrong. The confirmation bias led him to confuse problem solving and solution selling. This confusion is summed up by the adage: "Never let the facts get in the way of a good story."

Letting the need for a good story pollute the search for a solution frequently traps business problem solvers. Anyone who's been tasked with finding a solution to a tough problem is, understandably, anxious about the "selling" part—the

final "pitch" that completes the effort. Whenever you discover something, like an untested candidate solution, it's natural to ask yourself "How will I explain this?" But, as the example of Ancel Keys shows, jumping to selling an untested idea can result in the adoption of a poor or even harmful solution. Contrast this with what Lustig did in "Sugar: The Bitter Truth." Unlike Keys, he didn't let the need for telling a good story hinder finding the best answer. But unlike Yudkin, he didn't try to walk his audience through all the steps of a complex problem-solving effort. In Chaps. 10 and 11, we'll explain how to switch from problem-solving to solution-selling mode, and how to develop efficient and persuasive presentations of your solutions.

* * *

Like any linear description of a complex process, the overview we've presented is a simplification. The four stages of the 4S method are sequential, but, in practice, you won't work through them rigidly. Sometimes you'll double back to previous stages. The problem-solving process is inherently iterative. For example, you'll often revise a problem statement based on what you learn during your efforts to structure it. Refining the problem statement is a crucial part of structuring and solving a problem. Likewise, the way you structure the problem is likely to change after some analysis. In a hypothesis-driven approach, the facts will lead you to revise your hypothesis, while in an issue-driven approach, they'll lead you to think of different ways to slice and dice the issues. The 4S flowchart (Fig. 3.1) shows additional feedback loops.

Fundamentally, the 4S method encourages you to switch from an intuitive, informal, and automatic approach to solving and selling problems, to a reasoned, structured, and manual approach. Rather than solve problems based on what may instinctively come to mind, which leads to the pitfalls we discussed in the previous chapter, you deliberately and rigorously attend to stating, structuring, and solving them and selling your solutions.

Our goal in this book is to help you become better at solving challenging business problems and effectively selling your solutions to those who need them. In the chapters to come, we'll show you how the 4S method works, and how to use it to achieve this goal.

Chapter 3 in One Page

- The 4S method is based on the problem-solving approach of strategy consulting (PSAC), but addresses its two limitations:
  - *"Hypothesis-driven" in corporate setting → risk of confirmation bias, groupthink*
  - *Not all problems are amenable to disaggregation → some require creative thinking*
- The 4S method has three paths: hypothesis driven, issue driven, and design thinking.
- Each path covers the four stages: State, Structure, Solve, and Sell.
- *State* the problem, after learning enough about it:
  - *When you don't know enough, this requires using empathy techniques.*
- *Structure* the problem, depending on the path you're on:
  - *With a hypothesis pyramid (if you are highly confident in the candidate solution)*
  - *With an issue tree (if you don't have a good candidate solution, but can decompose the problem)*
  - *With ideation based on solution imperatives (if decomposing the problem is ineffective)*
- *Solve* the problem:
  - *By performing the analyses required (first two paths)*
  - *By prototyping and testing solutions (design thinking path)*
- *Sell* the solution, focusing on the answer and your audience, not on how you solved the problem.
- The 4S method is iterative and not rigidly sequential.

Notes

1. This doesn't imply that the approach applies only to strategy problems. Indeed, much of the work "strategy" consulting firms do isn't concerned, strictly speaking, with strategy. The term "strategy consulting" is shorthand for "CEO-level, premium management consulting," and "strategy" consultants use essentially the same problem-solving approach when solving organizational, operational effectiveness, or marketing problems.
2. Bacon, F. (1620). *Novum Organum,* XLIX.
3. Nickerson, R.S. (1998). Confirmation Bias: A Ubiquitous Phenomenon in Many Guises. *Review of General Psychology*, *2*(2), 175–220.
4. Friedman, R.S., & Förster, J. (2001). The Effects of Promotion and Prevention Cues on Creativity. *Journal of Personality and Social Psychology, 81*(6), 1001–1013.
5. Janis, I.L. (1982). *Groupthink: Psychological Studies of Policy Decisions and Fiascoes*. Boston, MA: Houghton Mifflin.
6. Liedtka, J. (2017). Evaluating the Impact of Design Thinking in Action. *Academy of Management Best Paper Proceedings.* Beckman, S.L., & Barry, M. (2007). Innovation as Learning Process: Embedding Design Thinking. *California Management Review, 50*(1), 25–56.
7. Simon, H.A. (1969). *The Sciences of the Artificial.* Cambridge: MIT Press.
8. Martin, R.L. (2009). *The Design of Business: Why Design Thinking Is the Next Competitive Advantage*. Cambridge, MA: Harvard Business Press.
9. Porter, M.E. (1980). *Competitive Strategy.* New York: Free Press.
10. Brown, T. (2009). *Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation.* New York: HarperBusiness.
11. Heath, C., & Heath, D. (2007). *Made to Stick: Why Some Ideas Survive and Others Die.* New York: Random House.