Realizing Radical Improvements in Performance through Solution Engineering

Fred Nickols

8/13/2012

DISTANCE CONSULTING

Many organizations with a history of success now find themselves outclassed by their competitors and rapidly falling behind. Catching up will require radical improvements in performance, especially the performance of key processes. Realizing radical improvements in performance calls for organizations to shift their approach to problem solving from one of "find the cause and fix it" to one of "specify the result and achieve it." This is a shift in focus from finding causes to configuring solutions and a shift in methodology from troubleshooting to Solution Engineering. Three examples of Solution Engineering in action are described in this article. Five sources of problems are identified and briefly reviewed. So is the concept of cause. Contrasts are drawn between troubleshooting and Solution Engineering. The companies, people, problems and events described in this paper are real. Only their names have been changed. The problems presented are on a small scale. The dollar amounts involved range from $250,000 to $1,000,000. There is no instance of a $60 million savings or an example of technology eliminating 5,000 jobs. Although small in scale, the cases presented are easily grasped. They also are clearly members of the same classes of problems found on a much larger scale. Thus, the cases presented can be generalized to situations involving much higher stakes.

SECTION I: Rationale

The game of business is won or lost on playing fields known as markets. The outcome is determined by the comparative performance of the players' workforces and processes. The performance of many organizations leaves them hopelessly outclassed by their competitors. These outclassed organizations have no choice except to embark upon a course aimed at realizing radical improvements in workforce and process performance. As will be shown, this requires a very different approach to problem solving.

It should be made clear right away that this paper is not a call for reengineering, although radical improvements in process performance will require a certain amount of process redesign and reengineering. It is, therefore, appropriate to say a few words about the reengineering craze of the 1990s.

Reengineering burst upon the corporate scene in 1990 and was "the management nostrum of choice" for several years afterward.[1] Its clean-slate design approach and its promise of order-of-magnitude payoffs had obvious appeal. Its advocates were adamant that it was not a fad. One critic suggested that it was not likely to be around long.[2] Another asserted that reengineering was "a repackaged product that does more harm than good."[3] For a while, all this remained to be seen. To paraphrase Cervantes in *Don Quixote*, "The proof of the reengineering pudding was to be in the eating."

Since then, reengineering has fallen into disfavor if not disrepute, seen mainly as a pretext for downsizing and, worse, a pretext that, once undertaken, has an alarming failure rate. The term itself has fallen into disuse. However, when you contemplate the task of realizing radical improvements in workforce and process performance, what better terms are there than "process redesign" and "process reengineering"? And so the terms will be used in this article. The claims for and against reengineering will be left to history and historians to prove or disprove on the basis of contributions over time.

If efforts to realize radical improvements in workforce and process performance are to succeed, they must employ a problem solving process very different from the find-the-cause-and-fix-it approach currently so prevalent in organizational problem solving efforts. This is true whether the improvement effort is an instance of the all-or-nothing, bet-the-business, throw-the-baby-out-with-the-bath-water kind articulated and advocated by Mike Hammer[4] or "kinder, gentler" version espoused by Thomas Davenport[5] or the highly disciplined and data-oriented find-the-root-cause-and-fix-it problem-solving approach used in Total Quality Management (TQM) projects. Radical improvements in workforce and process performance will not be realized by finding and fixing causes or by fine-tuning existing processes.

Problem solving is likely to play a central role in efforts to greatly improve process and workforce performance for other reasons as well. Process improvement efforts target key business processes; for example, order fulfillment, product development, and purchasing. These processes lie on the surface of a business. There is a deep structure to organizations as well as a surface structure. If an organization's deep-structure processes do not support the redesign and reengineering of its surface-structure processes, the effort will fail. In such cases, the choices are simple: forego any significant process improvement, or redesign and reengineer the deep-structure processes, too. Process improvement is truly revolutionary when it focuses on deep-structure processes.

Problem solving is a deep-structure process. So is decision making. Performance appraisal is another, and it exemplifies deep-structure processes. On the surface, and taken at face value, it appears to be an assessment or appraisal process. Upon closer scrutiny, it turns out to be one of the chief means whereby a vertical or hierarchical system of authority is maintained.[6] It is the "stick" in a carrot-and-stick management philosophy.

Decision-making offers additional insight into the distinction between surface and deep structures. How decisions are made is one thing, where they are made is another. To shift from autocratic to participative decision making, or even to shift the locus of decision making from managers to self-managed teams, are both changes in the surface structure of decision making. The deep-structure process at work here is the behind-the-scenes decision-making process that determines how and where other decisions will be made. This "back room" process is rarely discussed and generally left unexamined and untouched. Thus, although the locus of specific decisions might be moved about, the process that controls the locus of decision-making remains unchanged. In a very real sense, nothing changes at all.

As Chester Barnard pointed out, it is useful to conceive of an organization as a sphere.[7] At its core are deep-structure or control processes. On its surface are operating or business processes. Tying the two together are linkage processes, what some would call "command-and-control" processes.

Deep-structure processes are typically concerned with the control or governance aspects of organization; for example, allocating and maintaining authority, establishing the boundaries of organizational units, exercising dominion over them and the previously mentioned situating of decision making. Deep-structure processes must be examined, changed and occasionally overridden in the course of reengineering surface-structure processes. Problem solving and decision-making rank high among the deep-structure processes that might have to be altered.

With the possible exception of decision-making, problem solving is *the* core process in all organized endeavor. Problem solving is the process of figuring out what to do when what to do is not immediately apparent. In successful organizations, people easily forget, overlook or never realize that established routines are solutions to problems solved long ago. The fact that such routines become standard operating procedure is testimony to their effectiveness as solutions to the original and sometimes long-forgotten problems they were intended to solve. It also gives rise to a legitimate view of organizations as "a collection of solutions to old problems."[8]

As time passes, things change. If the pace of change is slow and steady, the amount of change to be accommodated is incremental. These incremental changes are no more than minor perturbations in an otherwise stable scheme of things. Accommodating such changes is a matter of evolution, not revolution.

The longer original solutions withstand the test of time, the more likely the focus of control will shift away from the original goal toward compliance with the patterned routines whereby the original goal was initially achieved.[9] In turn, the focus of problem solving shifts from finding ways of achieving goals to identifying and correcting deviations from accepted practice. In this way, original solutions become encrusted with the results of attempts to restore the *status quo ante*. These accretions obscure the original goal and, like barnacles on the hull of a ship, must be periodically scraped away. If not, goals become bifurcated and maintaining the *status quo* becomes an end unto itself.

If the pace of change picks up sufficiently, the requirements of successful adaptation move from evolutionary to revolutionary. The degree of change required escalates from incremental to radical. When radical changes are necessary, the nature of problem solving must shift from troubleshooting existing processes and systems to specifying, designing, building, installing, operating, and maintaining new processes and systems. This Solution Engineering view of problem solving must prevail in support of the efforts necessary to effect radical improvements in workforce and process performance. A troubleshooting view simply won't do.

A primary aim of this article is to illustrate an approach to problem solving called "Solution Engineering." Solution Engineering is a goal-centered, not a cause-centered process. It focuses on the solved state and on ways of achieving it. Its emphasis is on engineering solutions, not on finding and fixing causes.[10] A chief means of improving workforce performance is to improve its problem solving capabilities and that can be accomplished by adopting a Solution Engineering approach to problem solving.

A secondary aim of this article is to contrast Solution Engineering with the more commonplace find-the-cause-and-fix-it problem-solving process. For the sake of brevity, this latter, cause-centered approach will be termed "troubleshooting."

At this point, readers may choose between two approaches to reading this article. Those who prefer to begin with examples can read on without interruption. Those who prefer their concepts served up right away can skip ahead to Section III, beginning on page 12, and then return to Section II, which begins next. A brief recapitulation can be found in Section IV on page 18.

SECTION II: Examples

Solution Engineering Example # 1: The Database "Scrub" Problem

Company A, a $300,000,000 service bureau business, was feeling the pinch from increasingly successful competitors and heightened demands from customers for improved service at lower costs. As part of its effort to improve service to customers while reducing costs one of the company's computer-based program support systems was being moved from a minicomputer to a local area network (LAN). This transition required moving from a flat-file database structure to a relational database architecture. Approximately 1,000,000 records were involved.

In the existing flat-file structure, each record contained customer and transaction data. The new relational-database architecture was envisioned as having a one-to-many relationship between the records identifying customers and the records of their business transactions. In other words, information identifying customers would be listed once in the customer database, but the transaction database could have many records for a single customer.

Customer data often varied from record to record in the flat file. Joan Barnes, for instance, might be listed as J. Barnes, Joan C. Barnes and, in the case of data entry, gridding or scanning errors, as Joan Batnes or Joan Baqnes. Unless the customer data were first made consistent, many customers would be treated as different people in the new database.

To make the customer data uniform, a review of the flat-file records was proposed. The systems shop would print out a hard copy or "green-bar" listing of the entire database, and it would be subjected to clerical review. Corrections would be noted on the hard copy and, later, these would be keyed into the existing system via its update function.

The proposed approach was unacceptable to the product manager. It would take too long (at least six months), cost too much ($360,000), and, most compelling, it wouldn't yield a new database of sufficient quality. This last shortcoming arose from the indexing scheme or "key" used in the flat file. The flat file used a key consisting of the customer's last name, first initial, and date of birth. Coupled with the data variations mentioned earlier, this meant that many records for the same customer would be so far apart in a printout of the flat-file database that reviewers would fail to recognize they were dealing with the same customer. At an estimated six months and $360,000 each, multiple hard-copy reviews based on different indexing schemes were clearly out of the question.

Stymied, the database conversion team, including the operations managers, solicited the counsel of the general manager responsible for the division where the customer service work was performed. Upon being briefed, he said, "We can start by cutting the $360,000 in half. If we download the flat-file records, load them into PCs (personal computers), and do the clerical review using PCs instead of paper, we can eliminate the duplicate effort of first writing and then keying the corrections, plus do a better job to boot."

"That's still way too much," responded the product manager.

"Okay, how about $90,000?" answered the general manager, halving the estimate a second time. "But I'll not go below that until I can get my hands on a sizable sample of the database to play around with, and see just exactly what cleaning up the database will take."

The product manager authorized a download of 7,500 records for the general manager's use. The general manager spent several hours poring over these records during the next few days. At the end of his investigative period, the general manager informed the product manager that the flat-file database could be cleaned up or "scrubbed" for a maximum cost of $40,000.

What was now identified as "database editing work" was outsourced on a fixed-price basis to a nearby data entry vendor. The flat-file database was downloaded and transferred to the vendor where portions were assigned to individual operators. Using the vendor's PC-based data entry system, each operator made three correction passes through each portion. The first pass, based on the flat-file key, enabled the operators to add some social security numbers and achieve improved consistency in customer names. A second pass, with the file indexed on social security number and name, facilitated additional corrections. A third pass, with the file indexed on last name and street address, permitted even more corrections. After the vendor returned the data files, they were passed through a final "scrubbing" process that eliminated stray characters, such as asterisks and slash marks. The process spanned six calendar weeks. The clean up of the flat-file database was completed in time for the planned database conversion and at a cost of $32,000, less than one-tenth the original estimate.

Review of the Database "Scrub" Problem

The objective was to transform a flat-file database from its current state into a form that would facilitate separating customer data from transaction data. The definition of "root" cause is instructive here. "Root cause is that most basic reason for an undesirable condition or problem which, if eliminated or corrected, would have prevented it from existing or occurring."[11]

The customer data in the flat-file were inconsistent from record to record for many reasons. These reasons were irrelevant to the task of making the customer data consistent. A search for the causes of these inconsistencies and attempts to fix them would have been irrelevant, too. Indeed, the shift to a relational database architecture was itself a solution to the inconsistent data problem. It is very difficult to have inconsistent customer data in a single record. The general manager focused instead on the goal or solved state, namely, identifying all customers in the flat-file database in precisely the same way so that when the data were moved to the relational database system no customer would be listed more than once, and all transaction records would be linked to the correct customer.

The operations managers were experienced and accustomed to providing intense clerical support for system changes and rollouts. However, they also were accustomed to thinking of clerical work as "pushing paper." Moreover, they viewed the work of making corrections on green-bar printouts as distinctly separate from the work of entering corrections via terminals. The company's job classification system (another element of deep structure) made a distinction between the people who performed the two kinds of work. Even though the "functional silos" that fragment work processes existed on only a small scale, they were very real in this division.[12]

The systems staff also saw the operations people as "paper pushers." They too were as blind to the use of PCs as a means of cleansing the flat-file database as the operations managers.

The general manager, a former management consultant, was relatively new to the company. He did not have to "break set" or "get out of the box" to see that using PCs to review and correct the flat-file database offered an efficient and effective means of cleaning it up. His mindset was not colored by years of indoctrination in "the way things work around here." To him, the use of PCs was an obvious possibility.

Engineer is both noun and verb. Solution Engineering makes use of engineer chiefly as a verb, as in "engineer a solution." However, someone must do the engineering, someone must be the engineer. The general manager was the solution engineer in this case. The significance of his contributions cannot be overlooked. Neither can his focus on the solved state.

Two more examples will illustrate other important aspects of the Solution Engineering approach.

Solution Engineering Example # 2: "Is There An Algorithm?"

Jimmy "Spider" Webb, vice president of Operations at International Assessment Services (IAS), walked into Robert "Bobby Ray" Floyd's office, sat down and, when he was sure he had Bobby Ray's attention, posed this question: "Given some number of candidates to be assessed, a set of rooms of known or determinable capacity, and a set of staffing guidelines, is there an algorithm that can be used to determine the optimum staffing level for any and all assessment situations?"

Without batting an eye, Bobby Ray, one of Webb's division chiefs, said, "Sure."

"What makes you so sure?"

"There has to be an algorithm. There are no vague or ambiguous variables in the problem."

"Okay. Would you work on it for a while?"

Bobby Ray set to work immediately. Two days later, he had the basic framework. Two weeks later, he had the algorithm. Two months later, the algorithm was implemented, but only after repeated attempts to "break" the algorithm as a result of testing it with a wide range of assessment center staffing conditions and circumstances, and considerable effort expended in selling it to those who had to buy it. The algorithm took the form of a simple five-step procedure any assessment center director could carry out. Two years later, it was clear to all interested parties that IAS was saving about a million dollars a year in assessment center staffing costs and, better yet, would continue doing so for the foreseeable future.

IAS had been in the assessment business more than 40 years when Bobby Ray was asked to tackle the staffing algorithm problem. Why hadn't an algorithm been developed long before? Some say IAS's best mathematicians had tackled the algorithm problem before and failed. How in the world was Bobby Ray able to engineer a solution so quickly?

The staffing rules were a piece of the puzzle, as indicated by Bobby Ray's confident response to the initial framing of the problem. More important, Bobby Ray moved quickly to specify the solved state. He asked his vice president two pointed questions.

"What do you mean by 'algorithm'?"

"A reliable method or procedure. It doesn't have to be a decision tree or a computer program."

"Okay. And what do you mean by optimum staffing level?"

"The least costly staffing arrangement given the number of candidates to be assessed and the constraints posed by the facilities and the staffing guidelines."

Questions regarding why there wasn't already such a staffing method, or what had caused previous efforts, if any, to founder or go awry, did indeed cross Bobby Ray's mind, but he discarded them as irrelevant. His task was to create the specified method, not figure out why it didn't exist. And he certainly wasn't going to waste time trying to prove it couldn't be done. He set about engineering the solution that had been requested.

Given the goal of "least costly staffing arrangements," Bobby Ray focused on costs, and staffing guidelines. What he found can be summed up as follows.

The staff at an assessment center includes Directors, Supervisors, and Associates. These are not IAS employees but, rather, employees of the institutions where the assessments are conducted. The monies paid for administering the assessment activities on behalf of IAS are honoraria, not wages. Although the honoraria rates vary somewhat, they fit this general pattern: Directors are paid $100 per assessment, Supervisors are paid $75, and Associates are paid $50.

The staffing guidelines stipulate that one Director is required for each assessment. Each room used during an assessment must have a Supervisor present. In addition, each room must have one Associate for every 50 candidates in the room. Finally, for every four rooms used, one Associate must be deployed to monitor the hallways. An exception to this otherwise logical arrangement is that the first Associate is required in a room when the number of candidates in the room reaches 35. Then the "rule of 50" takes effect, that is, another Associate is required when the number of candidates in the room reaches 50 and at every increment of 50 thereafter.

Bobby Ray constructed a model of the costs and cost drivers for an assessment that looked very much like the one shown in 1. From it, a number of observations can be made.

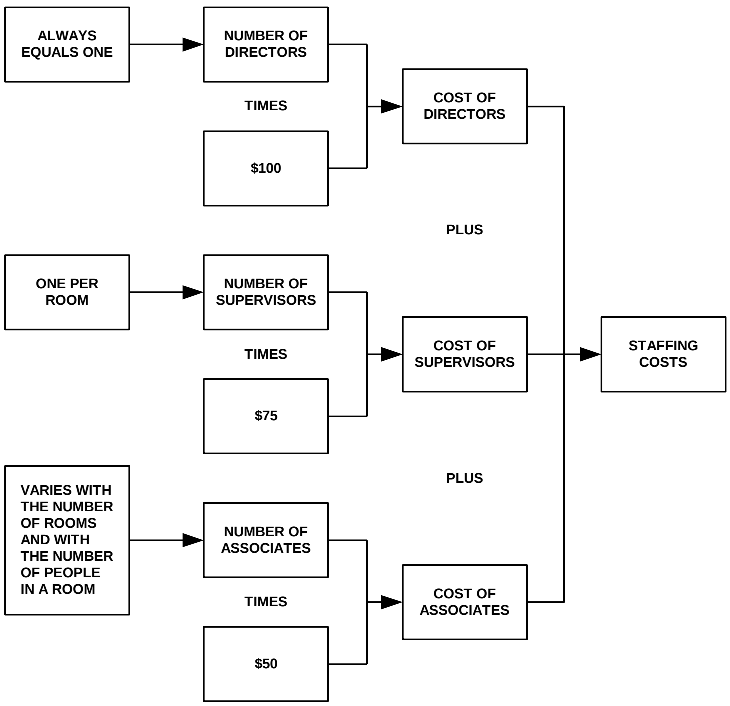

Figure 1 - Test Center Staffing Costs

The cost of the Director is a given. All assessments must have a Director, but only one.

Supervisor cost varies with the number of rooms used. This suggests a strategy of reducing the number of rooms used. This strategy is particularly appealing in light of the fact that Supervisors cost half again as much as Associates.

Associate cost varies with the number of rooms used and with the number of assessment candidates in a room. This suggests a trade-off between minimizing the number of rooms and maximizing the number of candidates in a room.

Bobby Ray was forewarned by his vice president that simple strategies, such as minimizing the number of rooms used or spreading the assessment candidate population equally across all available rooms occasionally work, but they don't yield the optimum or least costly staffing arrangements in all cases. Indeed, in some cases, such rule-of-thumb measures inflate the costs. For instance, cramming a large group of candidates into two large rooms can reduce Supervisor costs and eliminate Associate costs for hallway monitoring purposes, but it also increases the number of Associates required in the two rooms being used.

For the next few days, bobby Ray set up, and attempted to resolve, a wide range of assessment center staffing situations. These were submitted to Bob Henson, the director of assessment center operations. Henson could always determine the optimum or least-cost staffing configuration for any given assessment situation, but he was not able to articulate the method by which he did so. From Henson's corrections to the test cases Bobby Ray worked up, emerged the insight needed.

What "popped" into Bobby Ray's mind, so to speak, was a completely different picture of the problem. The picture was that of a large pitcher full of water and several smaller containers of various sizes. The volume of water in the pitcher, measured in ounces, represented the number of candidates to be assessed. The smaller containers represented the rooms available for assessment purposes. On the sides of the smaller containers were marks numbered 34, 50, 100 and so on. These, too, represented ounces and, at the same time, a number of assessment candidates. Some containers were smaller than 34 ounces and had no marks on their sides.

The trick, as Bobby Ray now saw it, was to empty the pitcher into the smallest number of containers, and to then pour water from one container to another to minimize the number of marks crossed. This, he further saw, could be accomplished by selecting a set of containers beginning with the largest available, then the next largest, and so on, until the capacity of the containers selected would accommodate the amount of water in the pitcher.

Bobby Ray's vision of the solution to the problem also entailed pouring the water from the pitcher into the containers in successive passes. On the first pass, he would fill the container to the first mark, or to capacity if the container held less than 34 ounces. On the next pass, he would fill to the next mark or to capacity, whichever came first. Then, with the pitcher emptied, he could pour water from one container to another to minimize the number of marks crossed. If necessary, he could even pour water from the initial set of containers into a new one.

At this point, an example of the staffing method in action will no doubt help. (The actual "algorithm" or staffing procedure is shown in Inset 1 at the end of this paper.)

Given a set of 156 assessment candidates and a set of rooms consisting of one large room capable of holding 80 candidates, a room half that size, and a number of rooms capable of holding 30 candidates each, the initial allocation of candidates to rooms would appear as shown in Table 1. (Note that only the first four rooms are used because their total capacity will accommodate the assessment population.)

Table 1: The First Pass

Available Rooms	Room Capacity	Candidates Allocated
1	80	34
2	40	34
3	30	30
4	30	30
5	30	0

Table 1 displays the results of the first pass. It also suggests that the second pass will consist of placing six more candidates in Room 2 and the remaining 22 in Room 1. The completed initial allocation of candidates to rooms is shown in Table 2.

Table 2: Initial Allocation

Available Rooms	Room Capacity	Candidates Allocated
1	80	56
2	40	40
3	30	30
4	30	30
5	30	0

The staffing requirements for this initial arrangement are one Director, four Supervisors (one for each room used), and four Associates (two in Room 1, one in Room 2, and one for the hallways). Total staffing cost is $600.

Moving 6 people from Room 2 to Room 1 eliminates the requirement for an Associate in Room 2 without incurring any additional costs in Room 1. This reduces the total staffing cost by $50. The revised arrangement is shown in Table 3.

Table 3: Revised Allocation

Available Rooms	Room Capacity	Candidates Allocated
1	80	62
2	40	34
3	30	30
4	30	30
5	30	0

Now, another opportunity presents itself — in the form of opening a new room. Moving 28 people from Room 1 to Room 5 eliminates the requirement for two Associates in Room 1 in exchange for one Supervisor in Room 5. The net savings is $25. The final allocation of candidates to rooms, reflecting an "optimum staffing level" or cost of $525 is shown in Table 4.

By the way, if the $75 total savings realized in the example above seems a piddling amount, try multiplying it by 15,000 assessments annually.

Table 4: Final Allocation

Available Rooms	Room Capacity	Candidates Allocated
1	80	34
2	40	34
3	30	30
4	30	30
5	30	28

Review of "Is There an Algorithm?"

Bobby Ray knew something about engineering solutions to business problems. He knew that solving any problem requires changing something. The search is always first for what to change and in what ways, and then for the means of changing it. He knew also that the search for solutions takes place in the structure of the problem to be solved. And he knew that having or developing a diagram of the structure of the problem to be solved is essential to its fast, efficient, and effective resolution. So he developed such a diagram. From it he was able to identify the variables he was trying to affect and then look for ways and means of affecting them. This knowledge, coupled with his experimentation, and the feedback he was receiving from Bob Henson, gave rise to the insight necessary to engineer the solution. Not once, however, did Bobby Ray look for causes.

Solution Engineering Example # 3: Staying On Top of Things

When Louise "Lou" McCoy assumed responsibility for the Special Services Division in her company, it was apparent the division could benefit from additional work. The sale of a line business had created excess capacity in the form of people who, lacking sufficient paid project work, charged part of their time to overhead accounts. This increased the "load" or indirect expenses for the division and, at the same time, reduced the hours charged directly to revenue-producing projects. The division was only two months away from a scheduled move to more expensive space as part of a larger corporate relocation. This meant more expenses would have to be recovered from a smaller base of hours, driving up the rates the division charged for the work it performed.

Being new to the company at the time, Lou took advantage of her "honeymoon period." She went to her colleagues, the other division managers, explained the problem, which they understood better than she, and asked if they had any work they would be willing to transfer to her division. Two of them graciously offered to assign some work to Lou's division and the rates were held in check — for a while.

Work comes and work goes. Things change. A year later, Lou was faced with the rates problem again. This time the culprit was neither excess labor capacity nor a move to more expensive space. The difficulty was rooted in indirect expenses; more specifically, in charges for space.

When Lou's division moved into its new quarters, the additional work assigned to it provided a broader base of hours and thus softened the effects of the increase in space costs. However, during the year, the division again lost some work. Further, as a result of internal transfers, retirements, resignations, and Lou's firm refusal to fill open positions, the division staff had been reduced, and productivity increased, by almost 20 percent.

As a result of the loss of work, the reduction in staff, and the increased productivity, the base of productive or directly chargeable hours had again grown smaller. Unfortunately, there was no corresponding decrease in indirect expenses. Of particular concern was the cost of the new and more expensive space to which all had moved. Lou's clients were once more concerned about her division's rates.

Although being overstaffed clearly wasn't the source of the division's difficulty this time, some well-meaning people recommended reducing staff, a move Lou knew could result in not being able to do the work at all. This time, there was no work to be transferred to Lou's division. To keep the rates in check required reducing indirect expenses. Based on an analysis of a model of the division's cost-accounting structure (see Figure 2), Lou targeted an amount of $250,000.

At the time of this "rate problem," as it was called, Lou's division occupied space in two buildings at the new site. The larger portion of Lou's division occupied 16,000 square feet in one building and the smaller portion occupied 5,000 square feet in an adjacent building. The annual cost of these 5,000 square feet was $225,000, roughly half consisting of charges for the space itself, and the other half consisting of allocations for common or shared space, facilities staff, security systems, and other space-based allocations.

Given the loss of work since the merger of the units, and the reduction in staff achieved through attrition, consolidating the two operational areas in a single, smaller space was an obvious and appealing option. Unfortunately, the primary work area of 16,000 square feet could not accommodate the 5,000 square feet occupied by the smaller unit. The primary work area had about 3,000 square feet available.

Lou had to figure out how to fit the 5,000 square feet in the adjacent building into the 3,000 square feet available in the main work area. This called for a 40 percent reduction in the space occupied by the smaller unit.

The manager and the supervisors of the smaller unit chuckled when Lou informed them of her aim. They said people had been trying to do that for years, and no one had yet succeeded.

What made the space problem so intractable was the nature of the work performed in the smaller unit.

Each year, thousands of candidates for cooperative job training programs submit applications and related materials such as references and transcripts. These materials trickle in over time. The materials are assembled in folders for review by selection committees. The folders are, or were, kept in open boxes on tabletops where they are easily accessed and readily available for transport to and from the sites nearby where the selection committees meet. Conventional filing cabinets would not do.

When the peak processing period arrived, Lou surveyed the scene. She saw 5,000 square feet of space, 3,000 of which was occupied by 40-50 tables covered with boxes of folders. The tabletops constituted a single, horizontal plane. If some way could be found to stack the boxes in a vertical plane, they would require much less space. Stacked four-high, they would require one-quarter the floor space they presently occupied. Lou had the general form of the solution to her problem.

"Why not," she asked, "move to a vertical filing system?"

The answer was that this had been thought of before, in the form of placing the boxes on shelves or using drum files. But the use of shelves prevented ready access to the boxed files and, owing to the weight of the boxes, increased the risk of spills. Drum files are huge, chain-driven devices, not easily moved about, and much too heavy for the floor load limits of the buildings Lou's division occupied.

Despite assurances that it was a waste of time, Lou asked one of her assistants, Rick Edell, to contact office equipment and supply firms to see what options might exist. Rick, to everyone's surprise, took the obvious course — he resorted to the Yellow Pages. Two of the firms he contacted claimed to have the solution to the problem — vertical filing systems. These use folders with end tabs for easy identification, and small removable containers to facilitate access as well as reduce the risk of spills. Moreover, they are lightweight and would not exceed the floor load limits.

Site visits were arranged, studies conducted, presentations made, and a deal struck. The times were hard. So was the bargain struck by Rick. The winning vendor wanted $25,000 for its vertical filing system. Rick agreed to pay $16,000. At this point, several others became involved.

The $16,000 represented an unplanned, unbudgeted capital expense. This required approval by Lou's vice president. He was concerned that Lou would spend $16,000 to make use of less space, but the costs of the space vacated wouldn't really go away. The 5,000 square feet vacated would move into the corporate pool of unoccupied space where its cost would be allocated to all cost centers based on the amount of space they occupied. Some of this cost would even make its way back to Lou's division.

Lou's vice president showed her how the expenditure could be justified on the basis of the future value of vertical filing systems in making better use of space throughout the corporation. The value to Lou's division of shedding 5,000 square feet of unwanted space was real enough to her, and to the programs her division supported, but it offered no real economic benefit to the company. Making better use of space did offer such a benefit.

Complicating the implementation of this now financially-approved solution was the fact that the larger of the two areas Lou's division occupied didn't have 3,000 square feet of contiguous space sitting vacant and ready to be occupied. The main workspace had to be rearranged. Moreover, the people in the work unit being moved had to be convinced that the new vertical filing system would satisfy their requirements. The vendor installed one system to use on a trial basis while the larger space was being rearranged.

The facilities staff was prevailed upon to rearrange schedules and priorities to accommodate the timing of the moves. Telephone hook-ups, data links, terminal and LAN connections were put in place. These costs, by the way, were not out-of-pocket costs but, rather, internally allocated costs that would have been incurred anyway. Thus, Lou was able to argue successfully that they did not offset the savings she claimed for her proposed course of action.

A solution that looked simple enough in conception proved quite complicated in execution. Nevertheless, it was carried out, and the two shops were merged. The vertical filing system worked. As Lou's vice president suggested, it is being applied elsewhere in Lou's company as a general strategy for reducing space requirements and fixed costs. More important, at least from Lou's perspective, her division's indirect expenses were reduced by $225,000. The rates were once again okay — for a while.

A Review of the Staying-on-Top-of-Things Problem

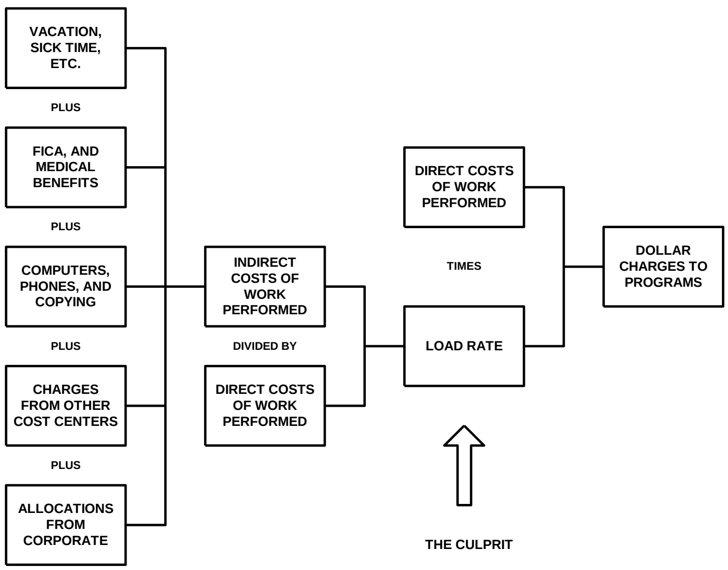

Figure 2 - The Structure of Load Rate

Reflect upon the incident just related. What was the problem? Rates too high? The pressure exerted by those who had to pay them? Too much space? Not enough work? Too many people? Moving to expensive space? Inefficient use of space? Poor planning? A lack of imagination? A failure to investigate? A tendency to confuse rates with charges? Ask six people and at least six different answers will be given. That happens when attention focuses on problems and their causes. But, focus on the solved state, and on the task of engineering a solution, and things become clearer.

The structure of Lou's problem was arithmetic. The problem was embedded in that portion of her company's cost accounting system related to cost center charges and the recovery of cost center expenses (see Figure 2).

The load rate was going up again. Unless it was offset by productivity gains, charges would go up, too. Upward pressure on the load rate was exerted by the relationship of indirect to direct expenses.

Lou's short-term objective was to bring the load rate down, to keep it at or near the preceding year's rate. This objective was itself a means to more distant ends, for example, managing the costs of work performed, controlling charges to the programs, and helping the company maintain a competitive posture.

At the time of the move, owing to more expensive space and a decreased workload, total expenses increased and indirect expenses constituted a larger portion of the expense picture than before. Spreading expenses over a smaller base of productive hours drove up the load rate, concerning those whose programs were charged for work performed by the division. The general form of the solution was to obtain more work for the division, increasing the base of productive hours, reducing the load rate, and reducing charges for work performed.

Later, owing to another decrease in workload, and a sizable reduction in staff, the base of chargeable hours had once again decreased. Indirect expenses again dominated the expense picture. This time, the general form of the solution was to reduce the amount of space occupied.

The rate problem just discussed is embedded in a larger structure consisting of the elements, connections, and relationships making up the cost-accounting system in Lou's company. Only a very small portion of this structure is shown in Figure 2. The complete model of the company's cost-accounting system constitutes the problem space and the search space for solutions to rate problems and other problems of a financial nature as well.

All cost-accounting systems are linked to non-financial but measurable factors; for example, square feet occupied, hours worked, and materials consumed. These non-financial factors are, in turn, tied to specific production processes and the resources required to support them. These linkages between resources and results make it possible to connect actions taken with their impact on the bottom line. They also make it possible to move from a sought-after bottom-line result to a range of actions that will produce it.[13] The ability to move back and forth between results required and the actions that will produce them, means it is feasible to engineer solutions to business problems.

Changing the elements in the cost-accounting structure in which the rate problem was embedded is an indirect process. None of the elements shown in Figure 2 can be directly manipulated. The search for solutions to problems embedded in the structure of any cost-accounting system is a search for actions that can be taken elsewhere and then ripple through the structure of the cost-accounting system, producing the effects desired. In Lou's case, corporate allocations for space were reduced as a result of decreasing the amount of space occupied by and charged to the division. As was pointed out at the time, space charges also could have been reduced by changing the rate charged for space. Lou's problem could have been solved by "a stroke of the pen."

The model in Figure 2 made possible several key steps in engineering a solution to the rate problem. First, Lou's analysis of the financial data for each element in the model focused attention on space charges as a leverage point. Second, the fixed, mathematical relationships in the model enabled her to determine the dollar amount of the savings required to reduce the division's rates to an acceptable level. Third, the model helped explain what was being done and why, thus contributing to a consensual view of the problem and its solution. Fourth, the model offered a useful way of visualizing the arithmetic structure of the financial aspects of the problem, just as visualizing the boxes of files in the vertical plane helped focus the search for a vertical filing system. Ultimately, fixed expenses were reduced by $225,000 at a one-time cost of $16,000.

What's the point? What's the moral? Where's the magic? There isn't any magic, that's the point. There is instead a methodical way of getting from where you are to where you want to be, of being able to figure out what to do about situations that demand action, of being able to engineer solutions. However, there is a moral. In organizations facing technologically turbulent or hotly competitive business environments, problems are rarely solved in any permanent sense. Situations are continuously changing and evolving and must be continuously managed to maintain some approximation to a goal or solved state. It is useful to think more in terms of staying on top of things than in terms of solving problems permanently. There are few, if any, permanent solutions.

SECTION III: Conceptual & Theoretical Considerations

The Message: Focus on Solutions, Not Causes

Many people believe that solving a problem is a matter of finding and fixing the cause. This is true in some cases but not all. To solve a problem is to realize the solved or goal state. This might or might not entail a search for cause. In many cases, a search for cause is futile; it can’t be found or, if found, it can’t be fixed. In some cases, it is counterproductive. In all cases, however, a solution — a course of action that leads to the solved state — must be developed. More important, it must be implemented. No solution is ever really a solution until proven to work. Developing and implementing solutions to the kinds of problems encountered in an organizational setting requires considerable technical and political skills, and no small amount of artful contrivance. One of the chief meanings of engineer is "to arrange or bring about through skillful or artful contrivance." It is this sense of engineer that underlies the concept of "engineering solutions" and the process of Solution Engineering.

Almost 20/20 Foresight

This article was originally undertaken with three objectives in mind.

One was to elaborate upon the commonplace notion of a problem as a discrepancy in results, a gap between what is and what should be. The intent was to review the origins of what some call "gap theory," describe five different ways in which gaps in results can occur, and then show how only a Solution Engineering approach can respond to all five.

The second objective was to change the readers' thinking about the concept of cause. Readers were to come away from the article viewing the concept of cause as occasionally but not always relevant to their problem-solving efforts.

The third objective was to interest readers in thinking more in terms of how they might engineer solutions instead of finding and fixing causes.

What was originally intended as a brief article regarding a few key concepts has clearly become a much longer exposition. However, the original objectives are still valid. Chief among them is a brief review of the origins of "gap theory."

The Murky Origins of "Gap Theory"

It is not clear who originated the concept of a problem as a discrepancy between what is and what should be.

Charles Kepner and Benjamin Tregoe lay claim to it in their 1965 book, *The Rational Manager*.[14] In its annotated bibliography, they credit Herbert Simon with independently developing the same concept at about the same time.[15]

The reference to Simon leads to research that Allen Newell, J.C. Shaw, and Simon were conducting at the Rand Corporation in the 1950s. Newell, Shaw and Simon were bent on developing a "general problem solver," a computerized problem-solving program. Their research culminated in Newell and Simon's 1972 book, *Human Problem Solving*.[16]

John Dewey, the great educator, alluded to discrepancies as an element in the definition of a problem in his 1910 book, *How We Think*.[17] Dewey's formulation of the problem-solving process was so sound that psychologist J. P. Guilford once observed that, with minor modifications, Dewey's steps have been rather persistent over the years.[18]

Roger Kaufman, the undisputed champion of needs assessment, has been writing about needs as discrepancies in results for at least 40 years. Kaufman's view differs from those cited above mainly in that he defines a need as a discrepancy in results and a problem as a discrepancy that has been selected for resolution.[19]

In this writer's opinion, it is impossible to give credit to a single party for creating the concept of a problem as a discrepancy between what is and what should be. So, with credit given where credit is due, the ways in which gaps in results come about can now be examined.

Discrepancies between *what is* and *what should be* come about in four different ways, two of which can combine to form a fifth. These are briefly discussed next and illustrated in Figure 3.

Gap # 1: Something's Gone Wrong

This gap in results happens when things are going along just fine, what is and what should be are in alignment, and then — Wham! — something changes and performance deteriorates, usually in a hurry. Blown fuses and tripped circuit breakers offer examples close to home. So do flat tires. "The computer just went down," is a phrase that should be familiar to many readers. This something's-gone-wrong problem is the kind on which Kepner and Tregoe lavished most of their attention. It exemplifies the category Harvey Brightman labels "operating problems."[20] An important characteristic of this kind of problem is that *the deviation in performance is from a previously attained level or standard of performance.*

Not all discrepancies come about because of equipment, component or personnel failure, or other unwanted or unforeseen changes, and not all performance standards have been previously attained. There are other classes of problems to be considered.

Gap # 2: Changed Expectations

A discrepancy in results also occurs when expectations are raised. Suppose a firm has been enjoying a steady but unspectacular rate of return on investment (ROI) of seven percent. Suppose a new CEO takes the helm and commands a doubling of this figure. A discrepancy in results exists. There is definitely a problem to be solved. But it owes to raised requirements, not because something's gone wrong. Any search for cause will be futile — unless someone is willing to point the finger at the new CEO — yet, the task of achieving the new goal remains.

This class of problem is neatly captured in Rajat Gupta's response to questions regarding the kinds of changes he would make when he took over as the managing director of McKinsey & Company: "Nothing is broken. But our aspirations are higher."[21]

An interesting and challenging variation on this class of problem occurs when expectations or standards are in a state of flux. In such circumstances, the goal is a moving target.

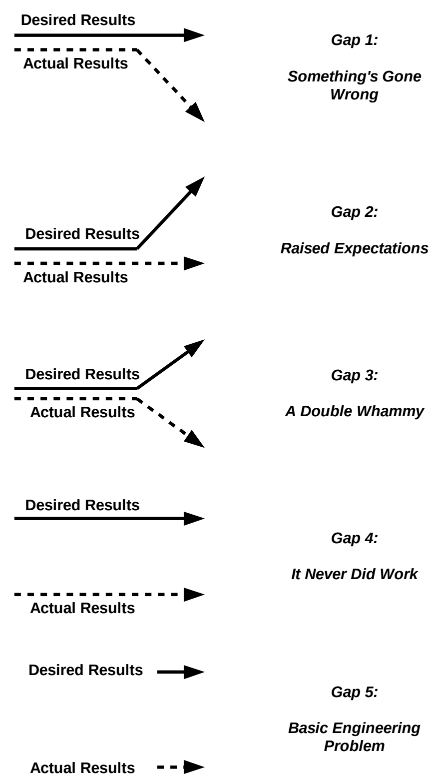

Figure 3 - The Five Gap Types

Gap # 3: Scylla and Charybdis (The Man-Eater and the Whirlpool)

Cost center and division managers are intimately familiar with this class of problem. It is referred to in some circles as "a double whammy" and in others as "caught between a rock and a hard place." A new system is rolled out, it doesn't work and service delivery suffers as a result. At the same time, budgets are cut and demands for improved service are made. On a grand scale, new competitors gobble up market share, eroding revenues and profits, and casting doubt on the value of the existing product line. Meanwhile, shareholders and analysts press for improved financial performance. (It is tempting to say that managers and executives who haven't faced and surmounted this class of problem haven't been adequately inducted into the mysteries of management.)

Gap # 4: It Never Did Work Right

More than one new system has failed to perform to expectations or requirements. The same can be said for many new products or procedures. Many problems can be cited as examples of what Kepner and Tregoe called "a Day One problem," a gap in results that has existed since the system, product, or procedure was first introduced.[22] Perhaps the best-known example of this kind of problem is "the lemon," a brand-new automobile plagued by one problem after another. The process called "systems development" has produced its share of computer-based processing systems deserving of the label "lemon." Some of these are now known as "legacy" systems. Poor specification, poor design, and poor execution are generally the culprits here.

The different classes of problems presented so far have two qualities in common: a history, and an existing system. The next and last kind has neither of these qualities.

Gap # 5: The Basic Engineering Problem

A distinguishing feature of this class of problem is its lack of history. In addition, there is no existing system to troubleshoot or diagnose. Results have been specified for the very first time. The goal is to design, build, install, operate, and maintain a new system that will produce the desired results. Any solution to this class of problem must be engineered — in every sense of the word.

According to Brightman, this situation calls for "strategic problem solving." Clearly, solving this kind of problem is not a matter of figuring out what went wrong or of figuring out how to cope with changed expectations. It most certainly is not a combination of the two. There are no "root" causes to be excavated.

It is this class of problem that most clearly calls into question the concept of cause.

The Concept of Cause

For many managers, executives, consultants, and academics, the concept of cause is defined in two basic ways. First, we will examine the Kepner-Tregoe view, then the Total Quality Management (TQM) view.

In the first edition of *The Rational Manager*, Charles Kepner and Benjamin Tregoe defined the cause of a problem as an "unwanted change." In so doing, they confined their problem-solving method to the "something's-gone-wrong" kind of problem represented by Gap 1 above. Sixteen years later, in *The New Rational Manager*, Kepner and Tregoe maintained their original definition of cause, but acknowledged circumstances in which the *should be* condition has never been attained. Kepner and Tregoe termed this a "Day One" problem. It corresponds to Gap 4, the "it-never-did-work-right" class of problem.

Under the precepts of TQM, the "root" cause of a problem must satisfy three criteria. First, it can be shown logically to explain the problem, that is, a cause-and-effect sequence can be traced. Second, it is directly controllable. Third, the effects of correcting it can be determined.

Root causes are not necessarily causes in the Kepner-Tregoe sense of unwanted changes; they are factors that contribute to the current state of affairs. In complex situations, the number of contributing factors is enormous. To isolate those few factors that can be said to "drive" a situation is the essence of effective analysis.

As an area of professional practice, TQM is blessed with numerous tools and techniques for eliciting and organizing knowledge of the factors affecting results. The Ishikawa or "fishbone diagram," tree charts, flowcharts, process charts, and various means of displaying statistical data come immediately to mind. But, whether or not the TQM tool kit contains the kinds of tools needed to engineer solutions to business problems is debatable. In general, in my opinion, TQM tools support refining and improving upon existing business processes, but not engineering or reengineering new ones.

The Goal

Referring to the five gaps shown in Figure 3, paying particular attention to the time line at its top, and ignoring all information to the left of the arrowheads, it becomes obvious that the goal with respect to all problems, regardless of how they originate, is to close the gap, to get from where one is to where one wants to be. This means all problems can be approached from the perspective of engineering a solution. But not all problems can be treated successfully by searching for, finding, and fixing causes, because not all problems have causes. Moreover, not all causes are controllable or correctable. When the stock market crashed in October of 1987, there were doubtless many people who would have liked to put things back the way they were the week before. That was clearly impossible.

There is an important point to keep in mind here. Pointing to causes beyond one's control is not an acceptable response. That things go wrong is understood. That these things very often cannot be corrected is also understood — and irrelevant. The task of the effective manager and executive is to innovate, to invent new ways of achieving improved results at lower costs. Pleading causes beyond one's control is to beg two tasks that are ever at hand: figuring out how to achieve the results desired — and then achieving them. There are no excuses. Solutions must be engineered even if causes can't be found or aren't fixable.

Engineering Solutions

A problem exists when there is a requirement for action coupled with uncertainty regarding the action to take. The requirement for action is generally rooted in a discrepancy between *what is* and *what should be*. This discrepancy alone is not sufficient to launch a problem solving or Solution Engineering effort. Not all discrepancies in results are reason for concern and not all are worth remedying. So, in addition to the discrepancy, there must also be dissatisfaction coupled with determination to do something about it.

To solve a problem, something must be changed. In complex systems, change is typically indirect. Results are achieved as a consequence of intervening at points that are often far removed from those where results are measured. Revenue, for example, is not increased directly, but is increased as the result of actions such as price increases (or price reductions), sales promotions, better-targeted advertising and so on.

In goal-oriented problem solving, the search is not for the cause of the problem but for those factors that, if changed in certain ways, would produce the result desired. This search is made more efficient and effective if it is carried out by systematically examining the structure of the situation in which the problem is "embedded."[23]

Searching systematically through the structure of a problem for the factors to change to achieve some desired result is the essence of a goal-oriented approach to problem solving called "Solution Engineering." Solution Engineering is marked by its intense focus on the solved state, on achieving *what should be* without worrying a great deal about why *what is* exists.

Cause-oriented problem solving is widely known as "troubleshooting." Solution Engineering and troubleshooting are summarized in Inset 2 at the end of this section and briefly contrasted in the following paragraphs.

Solution Engineering and Troubleshooting

Goals. Solution Engineering and troubleshooting have different goals. The goal of troubleshooting is to find the cause and fix it, to put things back the way they were. The goal of Solution Engineering is to get where one is going.

Time Perspectives. As their goals imply, Solution Engineering and troubleshooting have different time perspectives. Troubleshooting looks backward. It seeks causes, which are often past events. Solution Engineering looks forward, to the future, toward some solved state. As a result, the starting points for the two approaches are also different.

Starting Points. Troubleshooting typically begins by defining the problem, usually in terms of a deviation from some previously attained performance. Solution Engineering begins by specifying the goal or solved state. Given different goals, time perspectives, and starting points, it stands to reason the two approaches rely on different search strategies.

Strategies. Problem solving has long been viewed as an information-based search activity. Troubleshooting and Solution Engineering differ greatly in their search strategies. Troubleshooting relies on a search strategy known to technicians around the world as "fault isolation." Solution Engineering relies on design. Solution Engineering and troubleshooting are both "structural" approaches; that is, the search space for both lies in the structure of the situation in which the problem is embedded, but the two approaches differ in the way they deal with structure. Troubleshooting assumes the structure of the existing situation as a given and concentrates on isolating the fault or malfunction in this structure. Solution Engineering seeks to design the structure that would have to exist in order to produce or realize the solved state. These two search strategies have different search objects.

Search Objects. Troubleshooting is first and foremost a search for causes and corrective measures. Something has gone wrong, or is wrong, and the task at hand is to find the cause and fix it, to put things back the way they were or were supposed to be. Instead of a search for causes and corrective measures, Solution Engineering is a search for change targets and for the means of changing them. Change targets are those factors that, if changed in certain ways, will lead to the solved state.

Outcomes. Both approaches have their uses and both can be successful; in some cases, dramatically so. But when all is said and done, the outcomes, too, define differences between the two approaches. When troubleshooting succeeds, the outcome is the *status quo ante* — the conditions that existed before. When Solution Engineering succeeds, the outcome is a goal achieved, the solved state, whatever it might be.

Incentives. No discussion of problem solving efforts carried out by human beings would be complete without some mention of incentives. The incentives for problem solvers engaged in Solution Engineering and for those engaged in troubleshooting are radically different. In many cases, when things have gone wrong, the view from on high is that someone "screwed up." There is a guilty party to be found and "hanged." This provides ample incentives to hide the problem and its causes. In contrast, Solution Engineering has a goal all are trying to attain. The incentives are to expose the problem and its solution. This difference in incentives is reflected in a difference in politics.

Politics. The politics of troubleshooting are frequently those of "Cover Your Fanny." Problems are seen as bad situations that shouldn't have happened in the first place and, after the guilty parties have been found and punished, this question will be posed: "What are you going to do to make sure it never happens again?" The by-products of this problem-solving atmosphere are predictable: disconnects, disengagement, low levels of trust, high levels of anxiety, finger pointing, scapegoating, and gamesmanship of the most sophisticated and counterproductive kind. In contrast, the politics of Solution Engineering are those of change management, of involvement, participation, commitment, buy-in, support, alliances, negotiation, change, and innovation.

Inset 2: A Comparison of Troubleshooting & Solution Engineering

	Troubleshooting	**Solution Engineering**
Goals	Find the cause and fix it	Create a new solution
Time Perspective	The past (restoring what was)	The future (creating what should be)
Starting Points	The problem state	The solved state
Strategies	Fault Isolation	Design
Search Objects	Causes & corrective measures	The required structure
Outcomes	What was	What should be
Incentives	Hide the problem	Find the solution
Politics	Blame	Change Management

SECTION IV: Recapitulation

No reader should construe this article as a condemnation of troubleshooting. I spent 20 years as a weapons systems technician in the United States Navy and I have the highest regard for technicians as well as intimate familiarity with the techniques employed in the course of isolating faults in malfunctioning systems. However, there are limits to the troubleshooting approach. It is appropriate in some situations but not in others. There are better ways to solve some kinds of problems and Solution Engineering is one of these better ways.

Nor should this article be seen as critical of TQM. But the tools and techniques of TQM have their limitations, too. Failure to recognize and account for these can lead to failure elsewhere, especially in high-risk, high-payoff reengineering efforts. The problem solving approach employed in TQM efforts is unmistakably a find-the-cause-and-fix-it approach. Finding and fixing causes is not the aim of Solution Engineering.

A basic meaning of engineer, as a verb, is to arrange, manage, or carry through by skillful or artful contrivance, as in "engineer a solution to the problem." It is this sense of engineer that all who live and work in an organizational setting must satisfy as they cope daily with the complexities they encounter. In a volatile business environment, instead of solving problems in any permanent sense, each day is often a struggle just to stay on top of things.

A search for the cause of a problem can be a futile undertaking. Not all problems are caused in the sense that unwanted changes cause discrepancies in results. Although the TQM definition of "root cause" is less bothersome, it, too, suggests searches that might prove fruitless. No matter how hard or how long one digs, there are many situations in which one can dig forever and still not unearth any causes, "root" or otherwise.

Moreover, a search for cause, by its very nature, takes place within the structure of existing processes and systems. Increasingly, companies are faced with requirements to engineer new processes and systems. The task at hand is to get on with tomorrow, not simply improve upon today, and most certainly not to put things back the way they were yesterday.

Most important, regardless of how a gap in results occurs, all gaps in results are amenable to closure through an engineering approach if, instead of focusing on the causes of the problem, the effort concentrates on finding ways of achieving the goal or solved state. A search for causes is relevant when dealing with a gap in results that comes about when things go wrong, but only if the aim is to put things back the way they were. Even so, correcting the cause of a problem might not be possible. If not, a solution to the problem still must be engineered.

Routine, stable business processes represent successful solutions to past problems. Unfortunately, the very success of these solutions can hinder successful adaptation to new and changing circumstances. When continuous, radical change is required in the service of renewal, business processes must be redesigned and reengineered. This must be done repeatedly, not just once. There can be no return to "business as usual" in what has been termed variously as "The Age of Discontinuity,"[24] "The Age of Uncertainty,"[25] and "The Age of Unreason."[26]

Efforts to improve process performance typically target business processes on the surface of an organization. It can be the case, however, that reengineering efforts must be supported by accompanying changes to deep structure processes. Problem solving is a deep structure process. In conjunction with decision-making, it is the process whereby the courses of action that determine success or failure in other arenas are initially identified and settled upon. Consequently, an organization's problem solving and decision-making processes are its chief reengineering tools and, at the same time, they can be its primary targets.[27]

Of the two problem solving approaches examined in this article, Solution Engineering is far more likely than troubleshooting to successfully support efforts to realize radical improvements in performance. As was stated earlier, such efforts are not about finding causes and fixing them. As for the problem solving model or "root cause" oriented approach generally employed in TQM projects, it will suffice here to say that although TQM and reengineering are complementary business process improvement strategies, their tool kits are not interchangeable.

In conclusion, those embarking upon efforts to renew, rethink, reinvent or reengineer their company's key business processes and otherwise radically improve process and workforce performance, might profit from this advice: Forget about causes, focus on solutions!

SECTION V: References

1. The quotation cited here is from Thomas Davenport's review of the Hammer & Champy book. See Davenport, T. (Fall 1993). "Book Review: Reengineering the Corporation." *Sloan Management Review*. Massachusetts Institute of Technology.
2. Richard Swanson, editor of *Human Resource Development Quarterly*, suggested in an editorial that Stephen Covey's *Seven Habits of Highly Effective People* is likely to have more staying power than reengineering because reengineering lacks a "psychological leg." See "Scientific Management Is a Sunday School Picnic Compared to Reengineering" (Fall 1993). *Human Resource Development Quarterly*. Jossey-Bass.
3. Certainly the most recent and perhaps the sharpest of Hammer's critics is Paul A. Strassmann, a Czech partisan during World War II, director of a $35 billion cost-reduction and reengineering program for the Department of Defense, and presently Visiting Professor of Information Warfare at National Defense University. See Strassmann, Paul. (June, 1994). "The Hocus-Pocus of Reengineering." *Across the Board*. The Conference Board.

4. Hammer's view of reengineering is expressed in two primary sources: Hammer, M. (July-Aug 1990). "Reengineering Work: Don't Automate, Obliterate." Harvard Business Review. Harvard University; and, Hammer, M. & Champy, J. (1993). *Reengineering the Corporation: A Manifesto for Business Revolution. HarperCollins.*
5. Davenport's "kinder, gentler" view of reengineering can be found in Davenport, T. & Short, J. (Summer 1990). "The New Industrial Engineering: Information Technology and Business Process Redesign." *Sloan Management Review*. Massachusetts Institute of Technology; and in Davenport, T. (1993). *Process Innovation: Reengineering Work through Information Technology*. Harvard Business School Press.
6. The way performance appraisal maintains the vertical hierarchy is expressed as "looking up" instead of "looking out" in a well-written article by James Thompson, manager of human resource research and development at Alabama Power Company. See Thompson, J. (1993). "Quality Improvement and Performance Appraisal." *Human Resource Development Quarterly*. Jossey-Bass.
7. Barnard, C. (1938). *The Functions of The Executive*. Harvard University Press
8. The view of organizations as "a collection of solutions to old problems" was first made known to me by William Buxton, the managing director of Manpower Development Services (MDS), a consulting firm in Gaithersburg, Maryland.
9. For a lucid explication of how focus shifts from goals to compliance see the seminal work by John W. Gardner: Gardner, J. (1981). *Self-Renewal*. (Revised Edition). W. W. Norton & Company. This same phenomenon is known in organizations as "bifurcation of interests." For a definitive treatment of bifurcation of interests, see March J. & Simon, H. (1958). *Organizations*. McGraw-Hill.
10. Nickols, F. (1993). *Solution Engineering*. Unpublished Manuscript.
11. Wilson, P., Dell, L. & Anderson, G. (1993). *Root Cause Analysis: A Tool for Total Quality Management*. ASQC Quality Press.
12. Nowhere is the impact of "functional silos" better described than in *Improving Performance* by Geary Rummler & Alan Brache (Jossey-Bass, 1990). The sub-title of this book, "How to Manage the White Space on the Organization Chart" refers to the "space" between functions.
13. A technique for tracing these linkages is explained in two articles. Nickols, F. (1979). "Measurement-Based Analysis: Hooking What You're Doing to The Bottom Line." *Conference Proceedings*. Organization Development Network; and Nickols, F. (1979). "Finding the Bottom Line Payoff of Training." *Training and Development*. ASTD: Alexandria.
14. Kepner, C. & Tregoe, B. (1965). *The Rational Manager*. McGraw-Hill.
15. Simon, H. (1960). *The New Science of Management Decision*. Harper & Row.
16. Newell, A. and Simon, H. (1972). *Human Problem Solving*. Prentice-Hall.
17. Dewey, J. (1910). *How We Think*. D. C. Heath.
18. Guilford, J. (1975). *The Nature of Intelligence*. McGraw-Hill. Guilford describes Dewey's formulation of the problem solving process as consisting of the following five steps: (1) a difficulty is felt, (2) the difficulty is located and defined, (3) possible solutions are suggested, (4) consequences are considered, and (5) a solution is accepted (p.313).
19. Kaufman, R. (1991). *Strategic Planning Plus*. Scott Foresman.
20. Brightman, H. (1980). *Problem Solving: A Logical and Creative Approach*. Georgia State University.
21. Gupta was quoted by John Byrne in "A Global Citizen for A Global McKinsey." *BusinessWeek*. (April 11, 1994).
22. Kepner, C. & Tregoe, B. (1981). *The New Rational Manager*. Princeton Research Press.
23. Senge, P. (1990). *The Fifth Discipline*. Doubleday/Currency.
24. Drucker, P. (1968). *The Age of Discontinuity*. Harper & Row.
25. Galbraith, J. (1977). *The Age of Uncertainty*. Houghton Mifflin Company.

26. Handy, C. (1990). *The Age of Unreason*. Harvard Business School Press.
27. Nickols, F. (1994). "Reengineering the Problem Solving Process." *Performance Improvement Quarterly*. NSPI.

Contact the Author

Fred Nickols can be reached via e-mail at fred@nickols.us. More of his articles can be found at www.skullworks.com.

INSET 1: THE FIVE-STEP STAFFING PROCESS

The five-step staffing process is as follows:

1. Determine the minimum number of rooms required to accommodate the testing population.

   This step requires you to array the rooms available in descending order of capacity (that is, from the largest to the smallest). Next, the capacities of these rooms (Rc) are added to one another until their sum (ΣRc) is greater than or equal to the number of candidates to be tested (n): (ΣRc >= n).

2. Allocate the candidates to the rooms in "chunks" equal to the room proctor break points or to the room's capacity, whichever comes first.

   This allocation should be done in "passes," that is, fill to the first break point or to room capacity for each room, in descending order, then repeat, filling to the second break point or to capacity, and so on.

3. If room proctors are required, check to see if candidates can be moved from one room to another, thereby eliminating the requirement for one or more proctors.

   This is the case when a) one or more rooms are at or just above a room proctor break point and b) one or more other rooms can accommodate additional candidates without reaching the next room proctor break point. In other words you can take candidates out of one or more rooms, reducing the number of proctors required there, without increasing the number of proctors in the room(s) to which the candidates are being moved.

4. If adding a room will not require adding a hall monitor, check to see if opening an additional room can reduce staffing costs.

   This is the case when a) adding a room won't increase the number of hall monitors required and b) the capacity of the room being opened will allow you to reduce the number of room proctors by at least two. In effect, can you swap two room proctors for an assistant administrator?

5. Make any final adjustments.

   These adjustments include "leveling" the number of candidates in a room so as to create equitable assignments for the staff, spreading or concentrating open seats for unforeseen circumstances, and so on.