As trainers we must transcend our tool-kits; getting out of — and staying out of — the "technical trap."

Designing From the Logic Of People

BY ANTHONY O. PUTMAN

There is an ancient Sanskrit sutra which ought to hang on the wall directly in front of every trainer's desk. It says: "Jnanam bandhah," which translates roughly as: "Knowledge is bondage."

Now I seriously doubt if the author of that cryptic gem was thinking of training design when he wrote it. But with a little linguistic updating and expansion it points to perhaps the single most important realization about effective training: the more we know about the technology of training — the methods, tools and techniques that form a trainer's "tool-kit" — the more likely we are to fall into the "technical trap" when designing programs. And once in that trap, we are very likely to produce something we have all seen more times than we like to admit: a training program that looks great on paper, but doesn't work very well in actual practice.

This article is about transcending our tool-kits; getting out of — and staying out of — the "technical trap" in designing training programs.

The Technical Trap

Consider the following (completely hypothetical) scenario:

Super Trainer: "So, what did you find out from your interviews at the finishing plant?"

Novice Trainer: "Man, have they got problems over there! Morale is lousy, turnover is way too high, and they've got a whole new crop of supervisors who are running around trying everything imaginable to cope."

ST: "Hmmm . . . sounds like they need some supervisory-skills training."

NT: "That's exactly what the plant manager said! (Pause) So, what exactly is supervisory-skills training?"

ST: (Smiling indulgently) "No problem, kid. We can give them some stuff on positive discipline, maybe some Maslow, be sure to hit them with a little TheoryX — Theory Y, right? And I just saw this great new package on listening for productivity, we could. . . ."

(That faint hissing sound you hear is the jaws of the technical trap closing.)

Now you and I know that nobody shoots from the lip as crudely as poor Super Trainer did, but the point remains: *before* designing began, the trainer's attention was primarily on the people in the finishing plant and their difficulties. As soon as Super Trainer started designing, the primary focus shifted to the technology of training —the lecturettes, modules, packages, handouts, etc. which form Super Trainer's "tool kit" — and the resulting design will almost certainly reflect the "logic" of that technology.

Unfortunately, it quite likely will not reflect the "logic" of the people at the finishing plant, because when designing began, they slipped completely out of view.

This dilemma is not peculiar to training; it can be seen in virtually all fields in which a substantial body of technical knowledge and skills exists. Computer programmers are notorious for getting so caught up in the "internal logic" of their programs that they lose track of the use (and users) for which the programs are being written; some hospital personnel seem to regard patients as a bothersome intrusion into their orderly medical procedures; a classic trap in advertising is producing ads that dazzle the ad industry but don't cause people to buy more Alka-Seltzer. In each case the same dilemma exists: the technology in which one has acquired such expertise has its own, internal logic which becomes powerfully seductive. The more expertise you have, the more seductive its logic becomes. And once seduced, you often lose track of the "external" logic that is required to keep your product on target, namely: the logic of people.

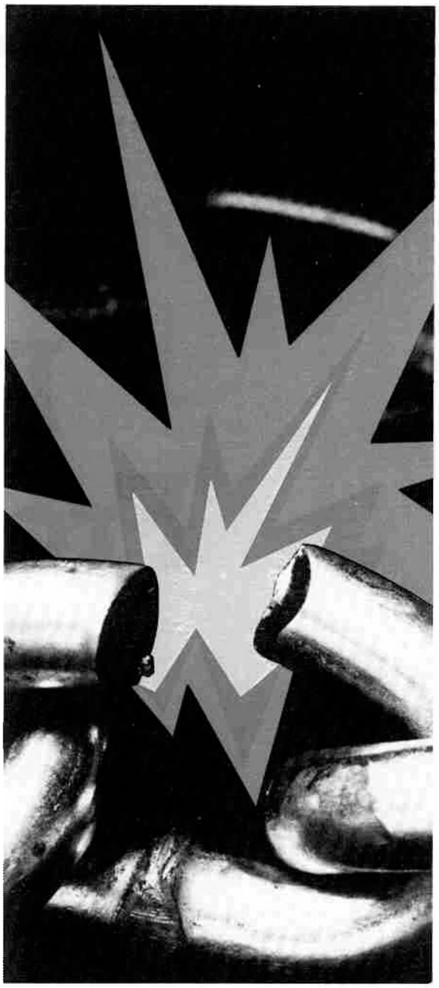
The Logic of People

Two years ago Chip Bell and I published an article[1] in which we suggested that real masters of training design concentrate primarily on its "art" rather than its "craft." We stressed that this "art" required operating from the perspective of "people logic," and gave a few examples of the difference using "people logic" makes.

The response to that article was quite positive. Typically, however, after saying how much they enjoyed it, readers asked, "Could you say a little more about how to design using people logic?" The article you are reading now was written primarily to respond to those requests.

Truth in advertising requires a word of warning: this will likely prove to be the most unusual "how to" article you will ever read, because it begins and ends by insisting that *there are no "how to's" about people logic — nor could there be.* "How to" implies that there exists a procedure, method, established practice, or formula that, if you competently carry it out, will reliably produce the desired outcomes. But this is exactly what we mean by "technology"; as soon as you have a "how to" you have added to your technical expertise. And the inescapable fact is that the logic of humans and human situations is far more complex than any "if this, then do that" technical logic. As Ron Lippitt so aptly puts it, "The biggest trap for a designer is to assume that, just because it worked in that situation, it will work in this one."

Instead of the "how to's" of technical logic, the ground floor of people logic is human skill, ability, and judgment. The bad news is that these skills cannot in general be reduced further to "how to's"; the good news is that we don't need such reduction, because all of us already have the necessary skills in using people logic — we wouldn't be people ourselves if we didn't. What we need is not more skills or mythical "how to's"; what we need is to be effectively reminded of what we already know, and to get some help in resisting being seduced by our own expertise. To these ends, then, I offer the following policies for designing with people logic. The list is by no means exhaustive, nor systematic; it is a set of reminders about how people differ from machines (or "trainees" — for that matter, "learners"). These are policies, not procedures; in any given instance you must (and can) figure out for yourself how to act in accordance with them.



Policies

1. *Focus on the person — not the program.*

This is the fundamental statement of the people logic perspective; all the other policies are elaborations and derivations of this one. Stated so directly it seems quite simple, and indeed it is, in much the same sense that walking a tightrope across the Grand Canyon is simple. Here's "how" you do that: "First, get your balance. Then take one step on the rope and keep your balance. Then take another step, etc., until you get to the far side."

Simple? Sure. Easy? Hardly.

The difficult task of keeping the person in focus is made easier by starting in the right place. Before thinking about handouts, content, modules, or any other "tool kit" question, a designer should ask four simple "people logic" questions:

(1) What psychological state are the people attending the program in to start with?

(2) What state should they be in when they leave?

(3) What do I need to do to get them to move from their beginning state to the end state?

(4) What do they need to do, see, hear, and experience to enable them to move from their beginning to the end state?

By psychological state, I mean the totality of personal, psychological characteristics which in some sense define what that person is at a given time. Using Peter G. Ossorio's classic paradigm of Intentional Action,[2] we can usefully talk about at least six fundamental parameters of a person's psychological state; these are:

1. *Knowledge, concepts, perspectives:* What facts, information, conceptual distinctions, and points of view (e.g. employee vs. supervisor) does the person have available to act on?
2. *Skill, abilities, competencies:* What does the person demonstrably know how to do?
3. *Motivations, wants, reasons:* What states-of-affairs does the person actually want (have reason enough) to bring about through action?
4. *Performances:* What is the person's "repertoire" of actual behaviors?
5. *Attitudes, beliefs, interests:* What are the stable characteristics which "pre-dispose" the person to perceive and act in certain ways?
6. *Eligibilities:* Which actions from his "performance repertoire" does the person perceive himself as being eligible for (allowed, permitted)? For which actions is he acknowledged by others as being eligible?

A couple of points regarding the "Performance" and "Eligibility" parameters:

Trainers frequently blur the distinction between "skill" and "performance," and then pay the price in the form of trainee resistance and resentment. Example: doesn't it strike you as odd, or perhaps a bit arrogant, to gather an experienced sales force together in a conference room and proceed to teach them "selling skills?" Problem is, those "old hands" have the misguided notion that they already know how to make a cold call, close a sale, and so forth, and they tend to get a mite testy with some trainer who tells them they don't. And the fact is, of course, that by definition they do already have all the skills they need for selling or they could never have sold anything successfully. What they do not have (nobody ever does) is a wide enough range of performances to allow them as much flexibility in exercising those skills as they would like.

Contrast: "I'm here to teach you the skill of closing a sale" with "You already know how to close a sale; most of you know several different ways. I'm here to teach you two proven ways to do it that you may not already have in your kit-bag." I suggest that most of what we talk about of as skills training is more appropriately thought of and approached as expanding the performance repertoire, that is, giving new options for the exercise of an already acquired skill.

The crucial necessity of changing the eligibility parameter is, if anything, even more frequently overlooked. The classic management rap against training is that it makes no difference in the bottom line, that is, while people may have learned a great deal in the training session, they are not able to use their learning to good effect on the job. This can hapen in one of two ways: either the designer got so deeply mired in the technical trap that the participants were not trained in what they actually needed, or else the training was well-targeted but the designer failed to provide for changing self- or other-perceived eligibilities. After all, knowing how to implement bottom-up communications avails you little if your boss continues to expect and permit only the top-down variety. We do little good (and sometimes actual harm) when we expand a person's knowledge, skills, motivations, or performances, but fail to ensure that he/she is in fact eligible to *act* on them.

Having thoroughly answered the four basic questions, we can now safely turn to our tool-kit to figure out how to bring about the desired changes in psychological state. But to keep from getting lost in the tool-kit, I find this second policy useful:

II. *Pay attention to the movement — not the method.*

Max Shulman in his brilliant comic novel *Sleep Till Noon* created a character who built an entire career on a single line. The man knew nothing about making movies, but he had learned to watch for moments in creative conferences when things seemed bogged down. At that point he would casually drop in his line: "Hey — how about a flashback?"

And since a flashback is almost never that bad an idea (and sometimes is quite a good one) everyone assumed the man had a canny knowledge of film-making.

Fortunately, it's hard to conceive of a trainer whose knowledge of training is restricted to a single line. But don't you know someone with the disconcerting habit of saying, almost any time there is a long pause in the design meeting, something like: "Hey — how about Desert Survival?"

Now I'm not knocking Desert Survival — it's a fine exercise. In fact that's part of the problem; it is a fine exercise, so you can almost always get something out of it. But this policy asks: is using Desert Survival the best way to cause movement in the participant's psychological state from where they are now to where they need to be next? In our enthusiasm for a tool, we sometimes forget to ask that question. Indeed, to steer clear of the technical trap, we should perhaps sharpen this policy into a stern admonition: Unless a tool clearly is the best way to get movement in the needed direction — don't use it, no matter how clever or impressive it is.

There's another aspect to this policy which is helpful when we take off our designer's hat and put on our session leader's beret. Sometimes we get so caught up in orchestrating the role-play or feeding back the instrument's scores or processing the exercise that we fail to notice something obvious — it isn't working. That is, it is not causing the movement we need. Worse, we may notice this fact but be so attached to this technique that we see the participants as resisting. When this happens, remember the policy; shift your attention to the movement. If your method isn't working, change it. If it still doesn't work, drop it and do something else. You gain nothing by grimly completing an exercise that is not bringing about a needed change in the participants.

III. *Follow the "3-D" Cycle: Diagnose, Design, Deliver.*

An old Scottish recipe for rabbit stew begins, sensibly enough: "First, catch the rabbit." Many recipes for training design have a similar sequential flavor: "First, diagnose the needs; then design the program; then deliver the training." What they generally omit is the next clause, which matches the recipe to the reality, namely: "And then repeat the diagnose-design-deliver cycle as many times as needed to complete the session."

Viewing training with the logic of people, we recognize two simple truths: (1) Every group of people (trainees) is irreducibly unique, in an important, non-trivial sense, and (2) The actual outcomes of any given training intervention will vary from group to group. Conclusion: we can only make shrewd guesses in advance about what state our group will be in at any given point in the training. Accordingly, as the previous policy stresses, we would be wise to pay continual attention to the movement (i.e. "diagnose") throughout the training; further, we should be prepared to redesign *on-the-spot*, depending on what our diagnosis shows us. Diagnose-design-deliver; diagnose-design-deliver; etc.

Having stressed the importance of on-the-spot *re*-design, I would like to get in a good word for *pre*-design. Some trainers (the brave, the confident, the risk-takers, or the just-plain-lazy) prefer to go into a session and "wing it" — find out "where the group is at," then "give them what they need." I tend to think of such performances as great stunts, something like a trapeze artist's triple-

somersault-without-a-net: a real grabber if you bring it off, but messy if you don't. I prefer the safety-net of a well-thought-out design; again to quote Ron Lippitt, "A design at least gives you something to depart from."

IV. *Make it — and keep it — intrinsic.*

A great deal has been written about how to create the necessary conditions for learning. Trainers are advised to design for participation, to work for involvement, to establish learning contracts, stimulate curiosity, release creativity, enhance self-responsibility for learning, etc. At the risk of adding yet another admonition to an already crowded list, I would suggest that none of the already mentioned ideas work particularly well unless they are undertaken within a particular human context, namely: talking/reading/thinking about something in which the "learner" is intrinsically interested.

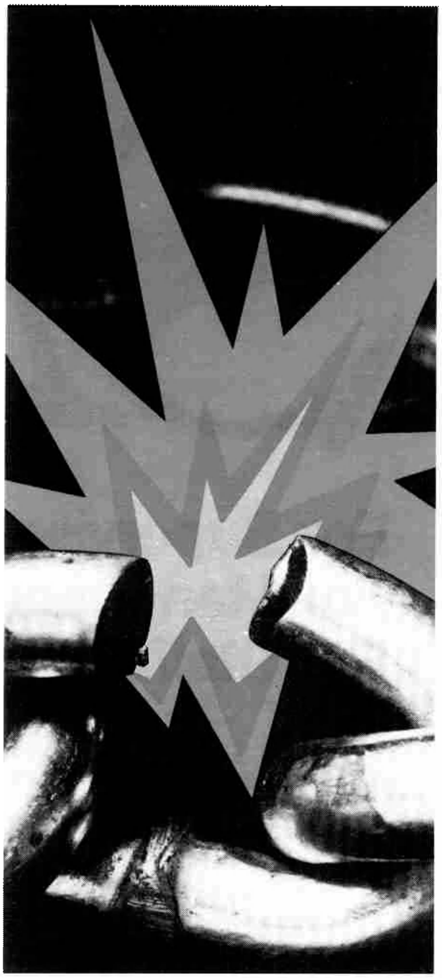
The distinction is a classic in psychology: an extrinsic activity is one I do primarily in order to achieve some other thing (including avoiding the consequences of not doing it); an intrinsic activity is one I engage in for its own sake. Some people jog to lose weight; real joggers like jogging. Most of us interact with computers because we must; many programmers do it because they enjoy it, and you have a difficult time prying them away from the terminal. You need to provide me with reasons (motivations, consequences) to learn about something that I see as extrinsic to me; for something I see as intrinsic, all you need provide is an opportunity.

Making the Topic Intrinsic

The psycho-social dynamics by which an activity becomes intrinsic to a given person are fascinating (at least to a psychologist), but for our purposes we need only recognize three points: (1) There are very few activities that are intrinsically interesting to everyone. (Recall the old joke about the sales trainer who walked into a room of bored salesmen, wrote "SEX" on the flip-chart, and said to a room of suddenly alert faces, "Now that I have your attention. . . .") (2) Learning occurs far more rapidly and effectively when the "learner" is intrinsically interested in the subject. (3) Most people are intrinsically interested in their work — or, to put it more exactly, in what they see as intrinsically part of their work.

That last point is crucial. All of us, mostly unconsciously, tend to divide our work into two parts: the intrinsic, consisting of all those things which we see as actually necessary to getting the job done, and the extrinsic, consisting of all the Mickey-Mouse, red-tape, and B.S. that other people (regulators, our boss, personnel, etc.) think is necessary.

Good people-logic design, then, consists of formulating training activities within the context of topics that are intrinsically interesting to the trainees (which, of course, requires a combination of diagnostic assessment and shrewd guessing). But what if they do not see the topic we need to present as intrinsic to them? A quick example



will reveal that there are ways of making the topic intrinsic.

Let's be honest about it: many managers only pay lip-service to performance appraisal. No matter how many times we tell them that "Performance appraisal is one of the most important parts of your job as a manager," even if we write it into their job descriptions, many managers will continue to see performance appraisal as a necessary evil, imposed by personnel or their bosses, having nothing to do with their real job of, say, keeping the plant operating at top efficiency. And, not surprisingly, it's hard to get such managers to learn how to do performance appraisal.

We could, of course, tell them once again that this is important, on the "If at first you don't succeed, try try again" principle. But a better approach might be to find out what they see as intrinsic to their jobs, and then describe again (in the recent jargon, "re-frame") the topic. I may not be interested in performance appraisal; I certainly am interested in finding a way to get Joe to meet deadlines, or developing three employees to promotable level, or getting my supervisors to take more personal responsibility for production. If I see what the trainer is offering as a direct means of doing what I already want to do, then my interest will be intrinsic. And I will be ready to be involved, participating, self-responsible, and so on.

In passing, we should note that one topic certainly is intrinsic to everybody, namely: me. What kind of person/leader/supervisor am I? What's my style? What do others think of me? Activities which address the endlessly fascinating topic of "me" will almost always be interesting to people, and therefore usually show up very well on evaluation sheets — whether they in fact helped create needed movement or not. In the legal jargon such things are called "attractive nuisances"; they are seductive, but potentially time-wasters. As designers seeking to create effective training programs, we should at least be healthily skeptical of their actual value.

V. *Never resist resistance — use it to facilitate movement*

This last policy reminds us that "resistance" is simply a message that we have misjudged the situation, and that what we are doing now is not what is called for. Instead of "resisting resistance," i.e. grimly sticking to the design or exhorting the trainees to get with the program, we should stop and re-diagnose. To help orient the "re-diagnosis," it is useful to recall a basic principle of psychodynamics, namely: "Coercion elicits resistance."

Any time you see a trainee resisting, ask yourself: who do they perceive as pushing them around? Were they ordered to be here? Do they resent having no say in the agenda for this training? Do they see you as trying to force your views on them, or tell them "the right answer?" Find where the perceived coercion is coming from — and then handle it.[3]

For those of us who have collected a substantial training tool-kit over the years, one final Zen story to remind us of our dilemma:

The Japanese master Nan-in received a university professor who came to inquire about Zen.

Nan-in served tea. He poured his visitor's cup full, and then kept on pouring.

The professor watched the overflow until he could no longer restrain himself. "It is overfull! No more will go in!"

Nan-in smiled. "Like this cup," he said, "you are full of your own opinions and knowledge. How can you see reality unless you first empty your cup?"

More tea, anyone?

REFERENCES

1. "Mastering the Art of Training Design," *Training and Development Journal*, May, 1979.
2. Ossorio, Peter G., *Persons*, Los Angeles: Linguistic Research Institute, 1964.
3. For more details on handling resistance, refer to "How to Handle Resistance to Learning," by Tony Putman, ASTD-'79 cassette tape, available from ASTD.

Anthony O. Putman is president of Descriptive Systems, Ann Arbor, MI.