

Inventor shares thoughts on creativity and motorcycles

By DON LEIGHTON

For most of us, commuting to work is pretty routine. Hop in the car, follow the same route, turn right here, changes lanes there for best traffic flow. It's so routine, in fact, that we let our minds wander to what we're going to be doing at work that day, or to some other favorite activity.

But that's not the way Phil Crosby does it. When he commutes, his attention is focused on just surviving the ride to and from work on his 500cc Honda motorcycle.

And, befitting his role as one of Tek's top inventors (11 patents), he's very inventive about his rationale for riding a motorcycle:

"I had wanted one when I was 18, but thought the way I drove I'd probably kill myself. Eventually my attitude about driving improved, so I finally bought a motorcycle five years ago.

"There are a lot of things about controlling a motorcycle that are really quite subtle. You have to know as much to ride a motorcycle intelligently as you need to know to fly a light plane. It's tragic that most people on the road don't appreciate this fact.

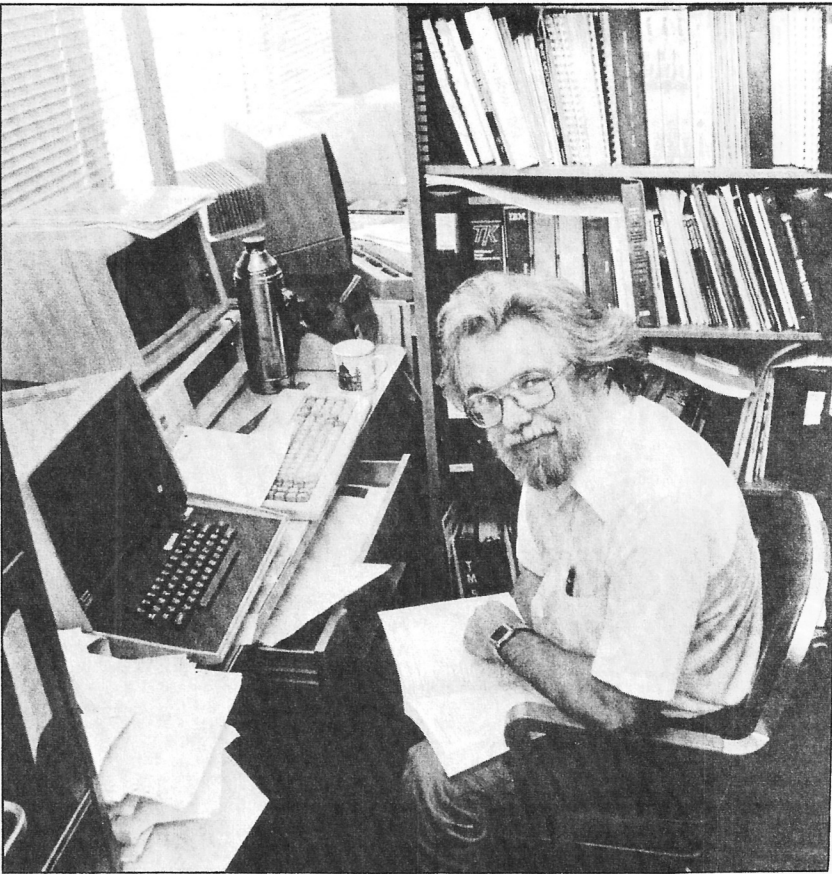
"In a sense, I've got something that's very much like having my own airplanes, only it's a lot cheaper and I get to use it more. It's a basic machine that responds very quickly to whatever inputs I give it.

"Getting the motorcycle was a clever thing. I get involved enough just getting to and from work safely that I'm not thinking about work. As a result, the trip is a really enjoyable, refreshing time."

Phil's motorcycle commute is a rare break from his job since work-related ideas are apt to pop into his head anywhere, any time, as when mowing the lawn or making sourdough pancakes from his 15-year old starter. He's also been known to bolt out of bed at 2 a.m. to enter some thoughts into his home computer.

As a substitute for thinking about work while commuting, he spends a few minutes beforehand at home organizing in his head what he's going to be doing that day.

Phil started at Tek as a summer student in 1957 after his junior year at Portland's Wilson high school. In 1959, after a term at the University of Chicago, he returned to Tek to work full time with Charlie Rhodes on television products. Phil finished his applied engineering degree as a part-time student at Portland State while working at Tek.



PHIL CROSBY

His interest in electronics developed about age 9, at least in part because his father was a ham radio operator. At 10, Phil had his own general class radio amateur license. "When I was 9, I was building things that sometimes worked and sometimes didn't, and things haven't changed a whole lot since then," he recalls. "I still build some things that work and some things that don't."

His early Tek projects included the 529 waveform monitor, the 556 dual beam conventional oscilloscope, and the 520 vectorscope. In 1968 he worked on the early 7000 Series products, the 7503 and 7403, and a number of plug-ins. He served as project manager for the plug-ins and did the mainframe electrical design. In 1971 he went back to television where he worked on the 1480, designed a video analog/digital converter, and became project manager for the 1980 ANSWER system. As a result of some ideas he got while working on the 1980, Phil is now pursuing those ideas in the electronics systems lab in Building 50.

Phil shared some of his thought processes in a Tekweek interview:

How do you get ideas?

I'll look at a problem from as many sides as I can, really think hard about, and then relax. It's like trying to remember someone's name. If you try hard, you can't ever remember. But if you try hard and then relax, it'll come to you—sometimes in five minutes, or sometimes it takes a day and a half.

Nothing ever really quite fits until you relax, and then something will pop up in your mind and you'll say, "Ah, of course. I should have thought of that." Generally I'll have a number of things going on at the same time. I'll work on task A for a while, and then when I get around to task B, I'll find that there are some things about task B that I hadn't thought of earlier that now seem quite clear.

A lot of it is just playing games with your head. If you play games with yourself, like putting yourself in a corner, you can maybe force yourself to come up with something to get out of the corner. Some of the things I've been happiest about were exactly those kinds of ideas.

One of the things I had patented was a circuit in which the only novel thing was one resistor. The original circuit had been patented some years before. In 1974, I was thinking of using that circuit, but I had a problem with temperature stability that kept me from using it. I stared at the wall for a while and figured out that all I needed to do was add one resistor to the old circuit, and that would improve the performance of the circuit by a factor of 20. That was all I needed. The original designer should have thought of that. But I probably wouldn't have thought of it either except that we had only this little bit of board space left and the only alternatives I had either weren't good enough or they took up too much space. So I had to think of something else.

How have technology changes affected your work?

I've gone from vacuum tubes to computers, so I see some big differences between how things were and what they're like now. In '59, for example, a new vacuum tube, the 6B18, came out that was about twice as good as any previous tube. The bandwidth doubled. What a designer could design then was limited primarily by available components.

You could argue that in some areas of lunatic fringe performance that's still true. But in most real world applications today, the limiting factor is not what we designers have to work with but what we can think of. Now it's more of a problem of having the imagination or the vision than it is having access to the things that will let us do the job. That's a significant difference and one that designers need to get used to.

When did you start using a computer?

I made the transition in the late 70's during the 1980 ANSWER project. Computers offered so much potential and were so fascinating that I had to jump in and get my feet wet. I think software is a very under-appreciated environment in which to get some very serious engineering work done.

It's really easy in this business to make a \$2 million mistake. Using computers for project simulation not only saves time but can also head off some costly mistakes.

How do you keep up with technology?

Probably the best way to stay current in this business is through doing design work. Besides building things myself, I try to talk with people about what they're building, or try to help them with problems, or have them help me with problems. That's one of the nice things about this business—there are always fresh people. They come at things with a different approach.

There's a lot more unplowed ground in this business than people think.

When engineers talk to each other about technical stuff, a lot of what they're doing is language definition. If you overhear a conversation between two engineers, very often it won't make sense to someone else, especially if the two engineers know each other well and understand each other. As you start getting inside someone's head you see how they look at problems. When you talk to someone, you are trying to explain something in terms that will work for them. In this whole process, you're picking up not only information from people, but you are also picking up methodology.

Such as?

Steve Roth (former Tek) is a fellow that I worked with very closely for nearly 10 years. There have been a lot of times when I've been stuck on a problem and I'd say, "OK, how would Steve solve it?" I'd lean back and think the way I think Steve thinks and pretty soon there'd be the solution.

You tend to borrow mental processes from people and hand them around. People learn a great deal by sharing these processes. It's important to discuss problems because what you're doing is sharing viewpoints and sharing approaches and you just never know when that's going to pay off for you.

Probably the highest calling of the human race is to have a neat idea and feel good about it.

People who stay aloof and isolate themselves sooner or later notice that everything's passed them by. This business consists a lot more of teaching and learning than is generally appreciated. You need to overcome being competitive. I was hung up on that the first three or four years I worked at Tek. I was looking at this stuff and thinking I wanted to be the best engineer at Tektronix. As I got more and more into the work, I finally realized that it's so doggone much fun that I don't care whether I'm the best or not. Besides, you can't even define best. Best at what—digital, solid state, software, frequency domain, analog? What are we going to talk about? So I was content to be what I figure is fairly good. That means I'm not in a big horse race with anybody else. And I'm willing to go ahead and talk with others about whatever I know and how I thought of it. And I expect they'll talk to me about what they know and how they thought of it. We'll all end up being better off.

This business is hard. We can't be infighting. Being competitive is actually setting yourself off from resources that you need to do the job.

Does problem solving take place in an orderly way?

When you read technical papers and talk to people, the message you get is that whatever it was that somebody thought of occurred in a fairly structured linear process. But the people writing technical papers or talking about their exploits are generally "Monday morning quarterbacking." They're taking what was probably a very chaotic process and trying to make it look a whole lot less chaotic than it really was. That tends to make people think that this business is one that's bound by reason. As a result, once someone comes up with a solution to a problem, others may assume there's not likely to be a better solution. That's a mistake.

Can you give an example?

Cliff Moulton (now profit engineering manager for Portables) is a brilliant engineer who was responsible for the first two TV products Tek ever made, the 524 and 525. He also did the 519, which for many years was far and away the fastest scope ever built. Cliff talked to me in 1969 about an idea he had that still had a couple of bugs. This idea had to do with turning on the writing beam of an oscilloscope, holding it on, and then turning it off. That's what we call dc-coupled unblanking.

We'd been doing dc-coupled unblanking since 1951. It was one of the things that made Tek scopes preeminent in the marketplace because they were the only scopes you could buy that worked the way they were supposed to. But there were still some problems with our method. The high voltage transformer was expensive to build and not as reliable as we'd like, for example. And we had to make some other things work with a bandaid approach. But the idea Cliff had didn't require bandaids.

Software is a very under-appreciated environment in which to get some serious engineering work done.

The point is that the idea Cliff had in '69 could have been implemented in '51. It wasn't that he was using any new components or anything like that. He was simply looking at the problem in a different way. It's a very clever solution that's been employed in virtually all Tek scopes built since then. It's a case where for 18 years we were going with an idea that wasn't as good as it could be. And that sort of thing happens all the time.

For a number of the things I've patented, I've had the feeling that somebody should have thought of them before I did. It seems like there's a lot more unplowed ground in this business than people think. A lot of people coming out of school have been programmed to believe that there are these older and wiser heads that are so much better that anything that has been worked on can't be improved. As a result, they may go off looking for other stuff to work on and generally feel frustrated because they end up expending themselves on a problem that's under-specified. There are still going to be better mousetraps.

Do you aim for patents or do they just come as part of the process?

A patent is always a result of finding a better solution to a problem. It's not the objective. When I start a project, I generally have the anticipation there will have to be some number of things that are going to have to be patented. You sense whether you're going to be able to think of those kinds of things when you fill in the various parts of an idea.

Have you ever had a project where you couldn't come up with the key solution you wanted?

... (long pause). You almost always allow for some kind of tolerances. But there have been times when I've gone ahead and done something while I still had the nagging feeling that there was a better way I hadn't thought of yet. That happens fairly frequently because you can't afford to run out all the possibilities. You look at some things and say, "Well, that works, but it shouldn't have that many parts."

You have the need to think up new stuff, but you also have the need to get everything integrated into some kind of package so you can start selling it. There's always a certain amount of tension between those two directions, and there should be.

So you have to compromise?

Oh, yes.

What do you think of people coming out of college now?

I can't say in general terms as we recruit only from the top of the class. We interview only a small percentage of graduates and then hire maybe only one-fourth of those. The ones we hire are very promising people. They're flexible. They're open to new ideas, and they are able to see connections between things. We've done a lot of filtering to make sure that we hire the kind of people who are going to make it here. What happens to the others I don't know.

Is it more important that they have the right thought process than have technical knowledge?

I think that's generally true, but I'm prejudiced. Getting a degree in applied science at PSU is a long way from attending a top-ranked engineering school, although that may have been an asset for me. The biggest dissatisfaction with the University of Chicago, where I started, was that they were trying to teach us how to think—and they said so explicitly in a couple of cases. That wasn't my objective. I didn't want to have somebody teach me how to think. I wanted to get information on what kind of things to think about. I'm not at all comfortable with having someone teach me how to think as I think that process works fairly well.

The limiting factor is not what we designers have to work with but what we can think of.

Anyway, as a result, I worry about some new employees being too rigid. Too often in school the process is finding the answers to a stated problem, whereas what engineers at Tek are really supposed to be doing is generating and defining problems, not solving problems. The curriculum where design work can be done according to a certain routine is software engineering. You are given a problem and are expected to come up with a solution that works.

It's helpful if a person knows a whole bunch of things, but in our interviewing process we try to find out if the person knows how to apply that information in a new situation, to make connections. That's the kind of thing they will actually have to do here, so we do a lot of sorting on that basis. What we're looking for is creativity, and that's something that's not well tested.

In our interviewing we look for things that are essential, but, unfortunately, are hard to quantify. But we can sense it after talking to a person for a while.

Young engineers are usually reluctant to toot their own horn. If they are reluctant, they are doing a disservice to people around them because they're not promoting the fact that it's OK to have a really neat idea and feel good about it. Of course it's OK. Probably the highest calling of the human race is to have a neat idea and feel good about it. You should rejoice in it and spread it around. When people start feeling enthusiasm and communicating it, you generally see a lot more productivity.

Doesn't it take a certain type of work environment to encourage the free spirit?

People are the work environment. They essentially are all there is. Sure, management is going to try this or that for a while. But don't ever get caught in the trap that they are responsible for your local environment because they can't change your local environment more than 10 per cent. I'll take that back. Make that 25 per cent. Bill Walker cut the engineering benches down from 8 feet to 6 feet.

This business consists a lot more of teaching and learning than is generally appreciated

People tend to blame management for making them unhappy about something. Maybe management has something to do with it, but generally it's a matter of the people in the group not paying attention to the need to keep the group ethic nourished. The environment is largely under our own control.

Do you typically spend long hours on projects?

I've worked on some projects that were real grinders. I'll accept going ahead on something like that if I've looked at it and see that there is a need to do it. I'll work 16 hour days a prolonged period of time to get the thing cranked out. But I would never feel good about being told to do that. Nor would I feel good doing that if it were going to cause serious problems for somebody

else. By and large, on most projects I've done, that's been pretty much the rule.

The only exception I can think of is when we finished the 7403. If I look at the collection of people on that project, about half a dozen, including myself, were divorced within six months. As far as I can determine, however, the project itself didn't have a whole lot to do with my divorce in 1971. (He's since remarried.)

My typical work day is about 9 hours. But when pushing on a project, that may stretch to 12-14 hours. In TV Products, product introductions are paced by the National Association of Broadcasters show so you can figure on a flurry of extra activity during February and March.

At one time on the 1980 ANSWER system, for example, I was at work 52 hours straight and didn't sleep at all for 60 hours. For a break we'd go out by the Coke machine and watch the sun come up. "Yep, the sun's up. Let's go back to work." We usually get into that kind of situation simply because we get too optimistic about what we can do.

We figure that if everything goes exactly right, we could get a project out by a certain date. But nothing ever goes exactly right. As a consequence, we find that we've tripped or really misjudged something and have to make up the time. I've done that a number of times and haven't felt good about making the misjudgment. Someone else might have misjudged also, but that doesn't make a whole lot of difference. I've appreciated that when I've made those kinds of mistakes, I've gotten the backing to push on anyhow. It's nice to know that management will have the faith and expectation of a positive outcome to go ahead and back a play even if I misjudge along the way.

Can creativity be rushed?

I think so. You can rush projects only so fast, however, as projects aren't simply a matter of creativity. They're checking catalogs, getting parts, making sure there aren't any loose ends. You can't accelerate that past a certain point.

But as for creativity itself, you can accelerate creativity by emphasizing the notion that a particular problem needs to be solved quickly. To do that you can shove yourself into a corner, so to speak. Throwing away some of your freedom forces the thought processes.

Is there any one thing you take special pride in?

The first thing that comes to mind is the people. I've worked with an awful lot of wonderful people, and I've watched most of them move on to other projects or into management. They've been damn good. Maybe I flatter myself, but I like to think I've had something to do with their success. And then new people come along. I get a lot from them. All these new people know things I need to know, so I'm getting as much in the exchange as they do. As far as building things, that's all been fun.

You sound like you're happy in your work?

Oh, yeah, I've always been happy in my work. There were times when I was a lot angrier with management than I am now. Two things turned that around: one is that I realized anger really doesn't create positive change, and the other is that there really isn't that much that management can do that impacts the way I do things—so why worry about it.

Because of your reputation, would management be reluctant to step on your toes?

No. I've gotten my toes stepped on. And in those cases, they should have been. There may be some reluctance like that, but if there is, there shouldn't be.

When you were a project manager, would you say you were easy to work with?

Reasonably easy, I guess. I haven't had many people quit on me or ask for transfer.

I don't like to tell people how to do something. I'll hand an idea out or make a suggestion and if the person takes it or not, it's OK because there's a good chance he or she has a better idea than I do. There have been very few situations where I've tried to ram something down someone's throat. It's mostly a matter of courtesy.

It also a way of avoiding responsibility. If they did everything my way, I'd get all the blame if something went wrong. I try to employ a mode of interaction in which all the information comes from the person I'm talking to. I try to play a non-directive role by asking questions.

Sounds like the role of a good teacher.

I've done some teaching. Over the last 10 years at two year intervals I've taught a course on television systems through the Oregon State University grad school. I like teaching. Again, it's the joy of having a good idea and sharing it. I want to spread that idea around because it's so much fun.