RUNDOWN ON ROUTINE 3: ROUTINE 3A CRISS CROSS

A lecture given on
26 April 1962

Saint Hill Special Briefing Course—a lecture, on the subject of Routine 3 and company.

History! The discovery that goals could locate terminals and that terminals could be audited out of existence was followed by a discovery of the identity of the Goals Problem Mass. It was followed also by several pathetic efforts of numerous auditors to find a goal, and more pathetic efforts to find terminals.

Meantime, research continued forward. We got Routine 3A. Routine 3 consists solely of finding a goal. Then finding a terminal that matches the goal, and running the terminal. And then finding another terminal for that goal and another terminal for that goal, till that goal disappeared. And then finding that the goal probably had disappeared, and finding another goal then, and finding a terminal for that goal and so on, and finding and auditing that, and then finding another terminal and auditing that, and finally it disappeared. And eventually you got into a situation where you'd find a goal and it'd blow up and you find a terminal and it'd blow up, and then you just couldn't find anything and you got a free needle.

What you've done in essence was to pick off a number of pieces of the Goals Problem Mass so the pc was floating free of the Goals Problem Mass. Condition of Clear which is attained by this particular process was a very desirable condition.

Now, on many people when it was well attained has been quite stable, but contained the liability of a key-in from the remainder of the Goals Problem Mass which had not been touched by this auditing And this was the reason Clears caved in. That's because there was still a Goals Problem Mass less just a few of its terminals, kicking around ready to kick the pc in his teeth.

All right. So there was an effort made, then, to find the goal and a new piece was discovered: the modifier. And you took the goal and the modifier, and you found the terminal with the goal and modifier. That was Routine 3A.

Now, the goal plus modifier ended up in a terminal which was much more intimate to the Goals Problem Mass, and because we reached time finding the modifier, we did not lose the Goals Problem Mass. The Goals Problem Mass can be keyed-in at any time, merely by repeating the modifier to the pc.

Why? The modifier ordinarily lies in the oppterm—new data for you. You find the goal then you find the modifier, and of course, you're going to find the oppterm. And that is a mass which is more intimately connected with the Goals Problem Mass, so although you find a terminal, all you have to do is chant the modifier at the pc a few times and you pull the Goals Problem Mass right in on the pc. Why?

Well, you've got your two big balls. Freud by the way would have—I'm sorry girls! But the truth of the matter is that that is how Freud thought it was all sex! That's right. That's right. He probably spotted some of these things someday and he says, "Aha! I know what that is!" Invented the libido theory.

Anyway. All right. Here you find the goal and then you find the terminal for that goal. But if you found the goal and the modifier, the modifier will have something to do with the oppterm. See? So, although you try to audit out this terminal, every time—you could audit it out, see. And then you just say the oppterm and you get that again—and you say the oppterm and you get that again, you see? So the Goals Problem Mass is over here quite fixed and you could always swing it in.

You can take any Routine 3 Clear and you can find their first goal, and you can find the oppterm for that goal, and bang—they're about as Clear as mud. In other words you can yank them straight into the Goals Problem Mass by merely finding the modifier.

Routine 3. Well, Routine 3 was designed to be processed. You found the terminal and then you found the goal and then you found the modifier and then you found the terminal, and then you processed the thing and of course it wasn't too terribly successful—it's an experimental process to that degree—because you were always kicking away at an unidentified modifier. You were kicking away at an unidentified oppterm.

So then we got 3D, and 3D—we skipped B and C and—they went by in a half an hour of research and—actually didn't go by at all, just skipped them—and we got to where you found the goal and the modifier and a terminal and then the opposition terminal. You did this in a sequence of finding the goal, finding the opposition goal, then finding the modifier to the goal— that isn't quite right . . .

Female voice: The oppterm.

No, you found the oppterm to the goal, and then you found the modifier, and then you found the terminal—and of course that gave you a package and that package was designed to be audited.

All this material is available in bulletins as I'm just giving you a fast historical review of Routine 3. It has never ceased to be Routine 3, by the way—if you notice that carefully—it never became Routine 4, 5 or 6. All right.

Now, then came Routine 3D Criss Cross. Routine 3D Criss Cross came about because auditors had so much difficulty in finding a goal and in finding a modifier, in finding a terminal, in finding an opposition terminal—had so much trouble finding these parts, that I looked very carefully for numerous new entrances to the Goals Problem Mass.

Various entrances were found; the first of them were merely arbitrary entrances. You said, "Make a list of the things you like." "Who or what do you like?" "Who or what do you dislike?" You had lists, in other words and these lists came on down the line and were nulled and you got items then on these lists, and then you opposed these—got the opposition item to these lists—and you've got a package—right there, bang. In other words you're dealing totally with terminals. So you'd get any—first it was an arbitrary assignment. You just told the pc "like," and then you got a line for "dislike" and then you got a line for something else. All right.

And this was succeeded by Prehav levels. You take a Prehav level and—which is assessed on the pc—and then you would list a large number of terminals for this Prehav level.

Prehav level, let us say is Badly Control. "Who or what would badly control?" And you'd make a long list, and you would assess that list out and you'd wind up with a terminal. And then you would say, "Who or what would oppose—," whatever terminal you found or whatever item you found. "Who or what would oppose that?" And you've got a pair of these things.

And you kept doing this on several lines, and then you got the flows, and you would assess all possible flows. And you would say, "Who or what would enforce inflow on you?" or something like that. And you would make that list and then you would make—after you had found that item—then you would make a list for the opposition terminal.

And the object of Routine 3D Criss Cross was to continue to do this until the Goals Problem Mass was totally shaken down to a point where you were only getting a few specific terminals which were constantly repeating. And this, of course, would be a package. And then that

package could be audited. That was the basic intention on which this was developed. Very good.

A new concept of Routine 3 Processes has just arisen and that is an understanding of what we are doing Several solutions have been found. I found several solutions to problems which opposed the original Routine 3, and what was the matter with original Routine 3? You see? What was the matter with it, basically, is that we were trying to do Routine 3 when we should have been Prepchecking See?

We were trying to do both things at once. We couldn't keep the pc in-session, and God help us, it was all over and splattered on the ceiling, you see? Couldn't keep the rudiments in. We found out that it takes forever to find an item or a goal if the rudiments are out. I can prove that to you over and over and over again. If the rudiments are out, you will find nothing. If the rudiments are out, you will find nothing If the rudiments are out, you will find nothing. I mean, it's been proven and proven and proven. I mean there is no fact that is harder driven home than that fact.

Of course we have Prepchecking You give CCH and Prepchecking to a pc, he'll get into a condition where the rudiments will stay in and then it becomes very easy for the auditor to do a proper job of auditing

Furthermore, auditing skill and the skill of an auditor has been tremendously upgraded. And that skill is tremendously higher this year. you realize that this week finishes a year of the Saint Hill Special Briefing Course? Actually it was Tuesday night, and if I hadn't been so busy getting audited and research and trying to tie this sort of thing up for you, why I would have given you a cake. But we will wait until we have a proper place to give a party and I will give you a party and we will just give you all a rain check on that, okay?

Audience: Okay.

Now, Routine 3's difficulties were basically that a pc was not in any condition to have anything found on him. And auditors were not in any condition, as far as auditing skill was concerned, to find anything on a pc. Let's just face it, we stunk. Pick up an auditor from lower South Amboy and say, "All right, show me how to operate this E-Meter."

And he'd say, "Well E-Meter operation is very easy, very easy. you set it up like this . . . You make sure this switch is at 'off.'" It's almost that bad.

One Saint Hill graduate went back to California and actually had tremendous numbers of auditors and was giving special lectures on just one thing: What is a null needle. Nobody knew what a null needle was. We had people coming in here, as original students last year— they would arrive here; they had never been on any courses or anything of the sort, and—from various parts of the world—and they'd strike the E-Meter Essentials and we would say to them in E-Meter Essentials, question on "Show me a null needle." And they'd say a null needle goes this way. And you say, "How does a null needle go?" and they say... Fantastic! The auditors didn't know anything about this box. I don't know what they thought I was trying to do. Maybe they thought I was kidding them—that it worked at all!

Actually we were. . . This is not too mysterious. Of course they didn't think an E-Meter worked. The E-Meters of Australia, South Africa and America, as well as New Zealand, would not demonstrate a mental charge. They demonstrated electronic charges. They were perfectly good electronically and they had nothing to do with the pc's thinkingness.

The last meters before that, that were of any value whatsoever, was the 1957 US. And I thought it was a very competent meter, until I was looking at one just a few months ago when I was over in the States at a congress and I found out, I found out that a Mark IV just runs rings around them. There's no sense in repairing those old meters; they've got delayed actions. They've got everything else in them they shouldn't have. It's marvelous.

You know, sometimes you turn your meter over to a repairman—he will build a delay into the circuit to keep the meter action from being injured. I think it's so sweet, the knucklehead! In other words, you ask the question and one second later the meter responds. Kazu! How gruesome can it get? It's grim.

Well, with no meters—and actually at that time no textbook on meters— and a few other things lacking, and with about half of the rudiments that are vital to keep a session straight up lacking, of course it was impossible to do Routine 3. How could anybody do it?

So, it was very important to realize that a tremendous number of things have happened in the last year. one of them was the Saint Hill Briefing Course. You take somebody; you set them up, and show them how it works and that sort of thing They get an understanding of it. They go back and they don't waste any time telling people how to do it right, believe me. Returned students can always be heard in here, clear to here, you know, they scream so loud. Auditing looks terrible to them when they get back home; it just looks grim. It looks too ghastly. I mean they always thought it was all right, you know, and they go back home and oh, it looks terrible. Inevitable, we get in those responses.

Developed the technology of an E-Meter; put it down in a very precision stated book. The book is very precisely stated. Frankly, made sure that the E-Meter stayed very stable. Wiped out all corny E-Meters that we couldn't count on. And somebody said well that was a smart commercial move. Hell, no! That wasn't any commercial move—that was a move toward the survival of Scientology. These meters, people going on using these meters—wow. And particularly when we started to get other data; how dangerous it was to audit with these meters. Wonder we ever kept anybody anyways.

And then we developed Model Session out with many more rudiments to keep things smooth, and we developed various technologies of how to straighten this out. And then more importantly, proved conclusively that it was out-rudiments that made it impossible to find items. That was—that was a very important thing. And then developed the technology of keeping rudiments in by Sec Checking, and then eventually developed Prepchecking. And, brought it on up to where we sit now. And it's an awful long jump—just twelve months; it's an awful long jump. But it's a jump that had to be taken.

Now, there are many important discoveries in this past year. one of them is the instant read. Don't pay any attention to anything that isn't an instant read, unless you're trying to spot something for the pc. He doesn't know what he's withholding and every time he goes zzzz, he thinks something, we say, "that." That helps him spot it because he looks at what he's looking at at the moment he—we said "that" and he said, "Oh, that," and then he spills it and it cleans the needle. It's just helping a pc out. Little tricks of this character are only assistive tricks. Remember there's tremendous vital data as run before this.

We have now a session that looks like a session. We have auditors that look like auditors. The thing runs well. This—it's a tremendous jump for an auditor to take, particularly when you've been auditing for years to all of a sudden be crushed into this much precision. It's a horrible thing, perhaps, to do to them, until they start getting some wins through having been pushed there. And they wonder how the hell they audited before.

Now, you've got this—you've got all of these developments and out of this comes new data now on how to clear a person all the way through the Goals Problem Mass. The matter is being given to you and on the research line, as a research activity, should be accepted as such. But nevertheless it's 9ufficiently good that you should do it.

The latest finding—I haven't enumerated all the findings which have made these things possible—but the latest finding on the thing is easily stated: Listing is auditing. Listing is not finding; listing is auditing And that changes the whole orientation of Routine 3—just like that.

I mean, all great discoveries along these lines are in the directions of simplicities, you see? Listing is auditing.

Now, early on, on Routine 3—there are some here that have seen this happen—you started getting an assessment and you would assess a goal, and then you'd assess a terminal and you didn't get a chance to run it, because it blew. And then you'd get another terminal and it'd blow. And then you'd get a goal and it'd blow. And what the hell do we have our hands on, here? Person was just assessing out through the blue.

All right, and my effort at this time is to move assessing all the way forward and dispense entirely with the Class IV activity of auditing anything found. That is the direction we are going right at the present moment. That is what we are trying to do. I don't know that we'll be tremendously successful in this, but we certainly will be more successful in finding a final package that will run. So therefore, whether we accomplish this in its final state or not, is unimportant. We will have a more valid package, and we will have a more comfortable pc.

Now, not the least of the reason for these discoveries along this line, and the reason for my address to this is the pcs get—have gotten terribly uncomfortable running 3D Criss Cross. Man, it can really rack you up. Now, you'd run out into the clear, you would run straight through this thing, but it's pretty gruesome. It's pretty hard, and it gets very difficult to hold the rudiments in if the pc is being run too far after a line is not discharged.

If we fail to discharge line B and we start to do line C, we're in trouble. The pc will have a hard time staying in-session; we'll have more and more difficulty with the pc. And if we then continue the error and finally get line D going, and then we oppterm the wrong terminal on line D; you get the wrong terminal on line D and we—oh, now it's just a dog's breakfast. Of course sooner or later all this will discharge. Sooner or later we get all this down. We get enough terminals. We discharge enough, and so on. We're not going to audit any of these things till we get a final package; the possibility it'll all come down to the end, but why have the pc caved in all this time?

I'm not kind, I'm nowhere near as kind as you are. It's a dirty word in Scientology, "He was a kind auditor." But it isn't just kindness; it's very hard to keep a pc in-session. So I've been studying how you patched up a case that had started going out of session because he'd had some wrong items found and what you do about this and I ran into this other fact: Listing is auditing

We sort of went through a period where we understood this. But we didn't state this, see. So listing is auditing.

All right, now, there's another point about this, is the item—a wrong item found is only remedied by finding the earliest item that was wrong on a case. That's worth knowing

You got D wrong You opptermed it and the case is lying in splinters. You actually are not going to remedy it very well by getting D right. If D could go wrong, A is probably wrong or B. This is the discovery of what is a dirty needle.

A dirty needle has two sources. Both sources are missing withholds. Well, you haven't gotten the item from the pc, so of course you've missed a withhold. But, you've actual—and you will start to get the pc dramatizing missing withholds. And you get a dirty needle, you pick up a missed withhold, you'll straighten out that dirty needle. And you get a dirty needle and you pick up a missed withhold and you'll straighten up that needle. That's just current session withholds. You can always straighten up a dirty needle by picking up enough missed withholds. But let me tell you, if you've got a wrong line sitting on that line plot earlier than the line you were doing, you're going to—liable to get a dirty needle, and you're just going to spend half your time picking up missed withholds, missed withholds, missed withholds.

What do we mean by a dirty needle? We mean a needle that's shivering and shaking and whizzling and it's got a pattern, and you can't read through the pattern, and so on. It's just hell trying to do a Routine 3 on a pc with a dirty needle. Well, you create a dirty needle with Routine 3 by getting a wrong goal or a wrong terminal; oppterming something that wasn't factual. In other words we've got line B and we—what we've got down there is "an acrobat." And what should be down there is "a dog" so we oppterm "acrobat," and by the time we're three-quarters of the way through or less, maybe a quarter of the way through oppterming "acrobat," we've got a dirty needle on our hands.

The way to solve this dirty needle is to go back and find the first error in Routine 3—any Routine 3, you see? Find the first error in Routine 3 and correct it and you'll find out your dirty needle will disappear and all of these later blunders will tend to blow off.

In other words it's like pulling basic-basic on a chain of engrams. You find the earliest improper Routine 3 action and you will get a relief of the later improper actions. Now, that's very well worth knowing, see? So that tells you sweepingly how to patch up a case. To patch up a case that has been misrun on Routine 3 of any kind, it is only necessary to find the earliest incorrect item and correct and complete it. That's the formula.

How do you set up a Routine 3 case that is running around the bend? Well you just find the earliest incorrect item that you can possibly lay your hands on—not the item that you said—not the item where his headache turned on; don't pay any attention to that item—not the item that was—he's having any difficulty with currently; don't pay any attention to that. Don't go getting all scrambled up. The best way to do it is go back and find the earliest incorrect item—the earliest incorrect action. Any type of Routine 3 action that is incorrect—find the earliest one and correct it, right then, and you'll find the rest of the liabilities will blow away. Then you progress forward from that point.

In other words you audit the case all over again. Interesting You're liable to find things that were in, not in anymore. And you're liable to find things, so on. But you'll nevertheless find a lot of charge blowing off the case because it's been stirred up.

Now, do you get that as a solution to the situation? Now, you saw me doing it last night on your television demonstration. I was just taking cases, and I was just—that's all the formula I was applying to the thing It—do you recognize it now as a formula? I'd find the earliest charged lists, where I was doing it, and squaring this around.

And listing is auditing That's very important, because it at once tells us when we have a complete list. All the questions of a flat process can be asked of a list, and if that list is flat then all these questions will go together.

So a flat process no longer produces tone arm action. Okay? So that's your first conditional. You understand, this has not been broadly proven. That's a conditional test. We don't know if that will hold true forever. But flat process—flat list. When is all the charge off of this list? When we get no TA action on the list, of course.

So theoretically you would list to no TA action—that's theoretical. Next is you'd ask the meter if the list was complete. Next is by differentiation. You differentiate the first twelve items of the list and see if you have tone arm action. If you have tone arm action, the list is incomplete. This could be fascinating, couldn't it? I mean, so you just read the first thing; you read the goal and you say, "Would a (whatever item you've written down) do (whatever the goal or line you got it from)." You see? See?

Should it—let's say the item you'd written down is "cat" and the "disregarded when you're trying to do something," you see, so you'd say, "Well, should a cat be disregarded when you're trying to do something?" And the pc says yes or no, we don't care what he says, but we watch that tone arm and we see that it has motion, and our list is not complete and that's all there is to it. So we don't do anything about nulling, we just go back and start listing

again—carefully keeping off the ARC break for asking for more and more and more. you understand? The pc gets to understand that listing is auditing and therefore he doesn't get so anxious to find.

All right, your next test is—consists of a nulling test. We null the first twelve—the first new twelve, because we can do this several times on the same list and if we find any items in, we complete the list.

What was that? What did you say? What did you say? If you find any items in, you complete the list. I mean if they don't null out instantly—they can tick, and that's all—you complete the list. If you have one item in, you complete the list. Simple, huh? I just dropped a grenade amongst you, didn't I?

All right. Here's another test. If you find more than one item alive on the list, you complete the list. you understand, we allow you—we allow you up to five chants of the item. Interesting, isn't it?

Supposing you found three alive that we're going to check out? When there's three, they'll— they stay alive. Bing, bing, bing, you know. Bing, bing, bing. They just go bop, bop, bop. Bop, bop, bop. Nah—complete the list. Complete the list. It'll complete down to one. See, that's what charge means on a list.

We're regarding now listing as auditing, you see. you—regarding it in a different viewpoint than you've been regarding it before. You've been going forward to find something and get on so that you could find something, and eventually having found a bunch of somethings you wind up with a package that you can audit. You see, that's another intention. Well this is not our intention, as will become very plain to you in about 2 minutes.

Remember, I told you Routine 3 would assess out. All right, that's what we're trying to do here. And you understand that I've got that "trying" underscored. You're trying to do this. So, when we complete the list, we'll find there's one item on it. It ought to be very rapid. Although listing may seem to take a long time, you're not going to spend any time nulling because they're all null. You're going to null the thing, you see, but if they're all null, well, they click and they tick and they click and then they're gone or they're not there at all.

Now, this makes you have to be pretty sharp on, "Are your rudiments in or out?" So you have to run your rudiments fairly often because a lot of these items are just going to be flatter than flounders. They were hot while you were listing them but they're not hot now. you see, that's the theoretical approach that we're working on.

So, we've got "Cat, cat, cat, cat," all null, "Dog, dog, dog," all null. "Catfish, catfish, catfish," all null. "Aviator, aviator, aviator," all null. There isn't a tick. Heh-heh. This becomes spooky ground. But remember, you're expert Prepcheckers. See, you know your business by this time. Is your pc in-session or out of session? You had one last test, is did it tick once in a while? See, well that might not be present on a complete list. See, you might not even get a single tick, theoretically. We'll see how it works.

All right. Now, when you've got your complete list, you're going to null it. And it should all null down like hot butter leaving you one item which will go tick-tick-tick. It won't go tick-tick-tick sporadically, that I guarantee you. It'll probably go microscopically tick-tick-tick.

Did you see "master craftsman," on that last night? Could you see the microscopic tick on that? And that was the sweetest tick I ever saw. It was meticulous, you understand? It wasn't doing anything unless I said "master craftsman," and at that moment it went tick. Did you see that? Instant tick. Tick-tick-tick-tick-tick-tick-tick, it didn't miss, see? It's the way an item ought to look. Not necessarily that tiny, but somewhat in that order of magnitude.

Of course the thing had been audited. But that's the way your items are liable to look on a complete list. Just one item, and it just goes tick-tick-tick. Very interesting. All right, that means you have to be pretty sharp-eyed, because if they're going to tick that microscopically you're really going to have to be looking. There's only one item in these five hundred and eighty-five that's going to tick at all, maybe. You see what a steep chance we're taking on this?

All right, so we find this thing. Finally we've got this item. We've got this item on this completed list. And we take this item, which, let us say, is "barrel." "A barrel." And now we're going to oppterm "a barrel." And all the rules of our oppterming "a barrel" apply, as I just gave them to you. When we get the list complete, why all of the tests I gave you, they will still apply, theoretically. See?

All right, we get down to the end of this thing—you understand this is theoretical approach I'm giving you. This actually is probably 3A Criss Cross—because you're going to use goals. I haven't gone into that. You've got this one item and theoretically what should happen, if you've done your job perfectly, you find out that "bungstarter" is the oppterm. You null it down— you got your list complete—and you null it down. It nulls down just as—probably even more gruesomely flat, do you see, like handling a dead fish, you know? And you find that it is "bungstarter."

Now, theoretically this could happen. This has already happened in auditing "Barrel, bungstarter," Pssswww. "Barrel" won't go, "bungstarter" won't go. That's your theoretical approach. There would be no purpose in auditing them, would there?

Now, there's several ways we could get our foot in the door on the Goals Problem Mass, but the way we're going to try to get our foot in the door, having confidence in you all, is we're not going to try to discharge the Goals Problem Mass by door—by, pardon me—we're not going to discharge goals lists to that degree that we discharge items lists. That's very conditional. I have to learn more about this before I give you too much more about this, you see?

It might have to happen that we have to take all the charge out of it. But listen, taking all the charge out of the goals list is pulling all of the little holes out of the balloons in the Goals Problem Mass and I don't think you're going to be able to do it. See, it's not an item list. It's a bunch of think. It's giving you the intention of masses. The intentions of valences is what that goals list is. And it's good enough to identify things with, but believe me, I don't think it's good enough to discharge a bank with; I don't think so.

So theoretically, we do not at this stage of the game—subject to amendation—care whether the goals list shows charge or doesn't show charge, as long as we could find a goal that goes tick-tick-tick, tick-tick-tick, tick-tick-tick and doesn't do anything else. We're going to grind it down to a point to where we've got a goal, and that goal really sticks with us. And it goes tick-tick-tick just as nice as you please.

Now, that may take a much longer goals list than we ordinarily have been writing because the goals list may discharge to a better degree. But it's certainly going to take a goals list long enough that we can identify the goal.

Now, the goal was appearing in about the first hundred and fifty that the person listed. That was quite interesting The goal seems to appear early; your items always appear late. So it obviously isn't the same breed of cat. Listing goals and listing items is apparently different.

Anyway, the way we're going to tackle this, from a Routine 3 action, is going to tackle it from the goal. And we're going to isolate goal. And then we're going to find, if we possibly can, a modifier. If it is too difficult to find the modifier we will skip it. I don't say this will be routine procedure but we is certainly going to find a goal, and we is certainly going to find a

terminal. And we're going to find a terminal on a totally discharged list. We're not going to find a totally discharged goals list, see. We're going to find a totally discharged item list, see.

We've got this goal, "To smoke cigars." And that list is not well discharged. That goals list is still punchy. Things are still happening in that goals list, you see. So we make a list, "Who or what would smoke cigars?" see? And we finally make an item list out of that. Now, that item list has got to be complete and that's got to be discharged. That's flatter than a flounder when we finally finish it up by this theoretical approach, here. When we finish that thing up we really have got that thing finished up. And we get it, "Cigar smoker," whatever the item was, and it's going tick-tick-tick-tick-tick and it doesn't do anything else but go tick-tick-tick. Do you see? And that's all.

And then we're going to oppterm it. "Who or what would oppose a cigar smoker? And we're going to run that list down and we're going to exhaust that list. We're going to run it down flat as a tire. And we should find—we should find a condition at the end where either the terminals are very weak, or whether they blow at once. That's theoretical on its approach.

All right. Then what are we going to do? How are we going to get our next line? Ha-ha— we're going to list goals. Going to list goals. Obviously the pc has changed like the North Star to the Southern Hemisphere, see? You can get new goals off this pc now. So you ask him to list a few more goals and be happy about this sort of thing, and you'll find out that some of the goals that are—weren't kicking might be now kicking. You're going to add some more goals and then theoretically again, we're going to approach this thing by nulling the whole goals list all over again to find out what we've got now. Because goals lists pump up and slow down and do peculiar things that items lists don't do.

Now, we're going to find a new goal that goes tick-tick-tick. And we're going to find a complete terminals list for that goal, and then we're going to find a complete opposition terminals list for that.

Now, this doesn't destroy any of your ability to do 3D Criss Cross. I could look upon 3D Criss Cross as a marvelous training exercise. That's worth much more than that but I could look on it now as a marvelous training exercise.

We let this person do a Prehav Assessment and we let him find an item. You know, when he's getting new on this and we don't pay any attention to what he's found. See, we find an item and we oppterm it. And he—we cuss him out, we say the list is not complete, and he tries to complete the list. And he learns—he learns on something he can't hurt much with, see? See? Pc might feel uncomfortable but nothing compared to how he would do if you mucked up a Goals Assessment, you see? See the training approach on that?

All right, but the actions are always the same. It doesn't matter what Routine 3 you've got, I call to your attention, you've got the same actions. You've got these certain steps and we haven't changed any of those.

Now, because we've put so much artillery behind it now, and because we've found out so many holes in the line, and if we know exactly where Routine 3 was falling down—remember Routine 3 was your most positive clearing process but that it didn't extend to the many. But Routine 3D Criss Cross can be done on anybody. Therefore if you use a Routine 3D Criss Cross but do Routine 3, you of course, got something which is extending to the many. See?

Now, if we do this right, and keep this remedied along the line, and if all this works very smoothly, and you don't develop a whole bunch of bugs that I haven't foreseen, which is always possible, because you see Routine 3 itself was thrown overboard only by auditor bugs. I can do Routine 3 to this day with the greatest of ease. If you don't develop a bunch of bugs and if there aren't a bunch of bugs in this thing, we may have our hands on, here, ample evidence, demonstrating it today, we may have our hands on a fairly rapid clearing thing. So the guy gets clearer and clearer. See, we don't have to push him straight through the knothole.

Get the idea? He just gets clearer and clearer as we go, and therefore, ought to get easier and easier to audit.

See, its basic advantage would merely be as every other development has been here, in terms of speed of final result. That's our basic advantage. Speed of final result is quite important. Auditor's tied up over one pc—even an hour saved by someone becomes important.

All right, well that's your theoretical approach, of what we will call, just for the interim, Routine 3A Criss Cross. The reason I've back-graded it is after we've got the thing totally ironed out, why, we will assign it its final designation and probably there will be a little bit of jostling around; there always is. I have learned to expect that as part of the randomity of it all. Doesn't matter how solidly you lay down these fundamentals, they can go astray.

Now, what I have just given you, by the way, aside from the fact of oppterming things, is contained in all the do's and don'ts of some of the earliest directions of Routine 3—to show you how un-new this is. But we have the wherewithal now, so that those earliest directions can be followed. We needed things like a textbook on meters. And we needed a splendid meter like the Mark IV. We needed this data about instant reads and we needed to know without any doubt whatsoever that you cannot find anything on a pc with his rudiments out. And you see, we needed this and that and the other thing as we've gone along the line, and furthermore we needed this data about complete lists. I've given you several tests for complete lists. And, we just had one, just a few days ago, see.

We needed these things, you see, in order to carry the thing through. But the way you're going, the only mistake which you are making and which you will have to remedy—and this mistake has been uniformiy made. Even I, last week would not have called any list I have recently inspected on your 3D Criss Cross, as a complete list. This—these lists are—man, I've gone over the number of times you've been going across a list and it is many. Why, you have to cover one item four times in order to get the thing—the whole list null—I'd say right then the list was charged.

Now, earlier, there was something else we were doing to take charge off of a list. We were differentiating a list to take charge off of the list. Well, there has been a technical discovery, and that is that a list does not discharge except by being complete. Do you see?

So you add up all the mistakes that you could make, actually come under the heading—except a goals list—is an incomplete list. Incomplete list, incomplete list, incomplete list. So you just get used to it.

What are the phenomena of an incomplete list? Well, the meter will respond to "Are there any more items on this list?" Sometimes, however, the meter doesn't respond to that question. Probably your tone arm action will be out of the listing, see, and those other tests which I've given you.

Now, you can relegate differentiation to a test step because it'll be your most reliable test step. Just differentiate the first twelve and see if there's tone arm action. I would say offhand this possibly won't prove out to be the case broadly. But it—apparently tone arm action would be caused by charge. So the list is still charged heavily if you have heavy tone arm action by differentiation. And tone arm action by differentiation would betoken the fact that nothing is going to null. So you wouldn't have to start nulling at this stage of the game because you say this list is too heavily charged up.

Now, I'm totally prepared to have to do something else with a list in order to discharge it properly. You're—I told you at the beginning of this lecture you're way up on the assembly line—I mean you're way up on the research line. You're not on the assembly line at all. See, that's about where you're sitting right now.

But because we have to patch up Routine 3D Criss Cross errors here and there—I've had to study those errors very carefully. And the errors which I have found in the session tests which you saw me take last night all add up to the conclusion that you might as well straighten up all the Routine 3D Criss Cross which you are doing But if you're going to have to do that, you might as well learn this Routine 3A Criss Cross. Because it's the same as straightening up what you're doing right now.

All right. You've got to do it anyway, so let's go ahead and tackle this thing called Routine 3A Criss Cross, which is really at this stage merely a patch-up of your 3D Criss Cross, plus the fact that nearly all of you have had some action on Routine 3, so we have to take that into account, see. We're in an optimum position where—just to straighten them out and take all the charge off—well let's do this other process and see what we get out of it, okay? That's were we sit, right at the present moment.

I want to see faster clearing than I have seen. I don't want pcs having to be pushed through the knothole. There's too many hours being stacked up on auditing. There are many errors being committed, but all of those errors sum up to rudiments out, which we won't make a criticism on your Routine 3D Criss Cross—that's a criticism of Prepchecking. See, if the rudiments are out that's a criticism of a person's ability to Prepcheck. So just omit that. And the other error is incomplete lists and improper items.

So, you've got to straighten up your incomplete lists and improper items anyhow, so let's do Routine 3A Criss Cross, find out where we land up and see if we haven't got a faster clearing process, okay?

Audience: Mm-mm.

Sounds to me that this might possibly give you an easier run of it with a pc.

I've been studying this hard, and I've—now I'll give you a few of the little items which have given us the gen on this.

Number one: I have discovered the dirty needle phenomena to stem from missed withholds. And at the same time an improperly procured item, that is to say an improper item, gotten and labeled and handled by the auditor gives the appearance of a huge missed withhold, which gives us a dirty needle. Now, this is one of our main difficulties, don't you see, with all Routine 3 Processes. That's a very fundamental discovery.

The next discovery is that listing is auditing.

Next discovery is that cases that have been started on Routine 3 are still continuing to dramatize their goal and modifier in spite of a great deal of Routine 3D Criss Cross run on the case. So that tells you, as I told you in a lecture—a couple of weeks ago I guess it was—it's the one found by Routine 3, the item found by Routine 3 was more fundamental—which I think is quite, quite remarkable—more fundamental than the later items. So these later items must be shallower than the Routine 3 items, because at no time has Routine 3D Criss Cross knocked out, to my knowledge, the goal and terminal activity and dramatization of the pc. In other words we have quite a long way to go before we come around to the initial results of Routine 3. So that looks to me to be a very fine area to shorten up processing. That looks to me to be very splendid.

And then, that the earliest items found are capable of spoiling the reaction and sequence and response of all following items. If you have one—items A, B. C, D, E, F. and an error, and it's found to be consistent in F and we can't keep the rudiments in on F. we look to G in order to correct F. and we only correct G. we'll find all of a sudden that there's going to be something a bit wrong on G. And we will find ourselves straightening out C—that we found something wrong on—and then spending quite a bit of time straightening out C. And then we find out the person didn't get the proper goal in the first place.

And we find ourselves right off the 3D Criss Cross graph, and find ourselves back assessing for a—an earlier error on a Routine 3 Process, which has apparently been able to terribly influence all the Routine 3D we were doing—3D Criss Cross that we were doing, you see, was influenced by the earliest error.

This is quite remarkable. In other words, you're going to run into people who have been assessed on Routine 3.

Now, here's the other—here's the other item that makes this vital to you: You're going to find people all over the place who have been assessed on Routine 3 Processes. And then they heard there was something new, later, so somebody dropped their assessment at that point— and it's lucky they did because it probably was for the birds—and we run on down the line here and you're going to run into somebody and he's going to say, "Well, go ahead and clear me, everything's fine, and go ahead and clear me, go ahead," and you say, "Well, all right." And you're going to prepcheck them, and get them into line, and you do some CCHs and so forth, and you get all set. And if you were just going to do Routine 3D Criss Cross you're going to be sitting there looking—the second you start listing you're going to be looking at a dirty needle.

You're going to say, "It was all straight on Prepchecking, particularly when I got the mud pies out of his mother, so, and here's this dirty needle." And you say, "Well, I'll carry on; I'll get the missed withhold," and you pick up four hundred and sixty-five missed withholds. That's an exaggeration, there's probably only fifty or sixty. And you get the dirty needle all beautifully cleaned up and you start listing again, you see, and there's the dirty needle again. Why, you're going to be baffled, unless at that moment you've been trained to ask the magic question: "Who assessed you on what? When?" and follow it down with the meter. "When was your earliest Goals Assessment?" or something like that.

"Oh, well, that was a long time ago. I remember Charley Gilpen came in from an Academy course that he'd taken—I think he was expelled—and he came in from uh—uh—uh—uhhh— Central Org and so forth, and he did a complete assessment on me. And uh—actually it took him about half an hour, and he found that I was a moped." And this is very peculiar, because his had just broken down.

And you say, "Well, what was he doing"

"Well, he was—he was—he was just listing goals, just listing goals."

"Well, how many goals did you list?"

"Oh, I don't know."

"What happened to the Goals Assessment?"

"I don't know."

You've had it, you see, you'd think, otherwise. But if you just sat down there and listed his goals again, this goal will appear again. If you know this little additional question—and this we went into, oh-ho, my goodness, this must have been ten months ago—something on the basis of invalidated goals. So you have to add—if you're going to do a patch-up list—you have to ask for what goals have been invalidated and clear up every tick you find.

See, it's kind of a funny little process of Prepchecking. "Have you ever had any goal invalidated?" See? And we don't care whether his mother did it or his school teachers or it was done in a session or he's done it out of session or anything else. We've got to take every tick off of that. If we haven't got the man's list, we can get his list back, only if we take the

invalidations off the goals because he's not going to give us the invalidated goal. And that's the only one he's not going to give us a second time. Isn't that interesting?

When you're redoing missing lists then, pick up invalidations of that type of list and clean that needle slicker than a whistle on invalidations of that type of list, and you'll be able to get the list back. If you don't do that you won't be able to get the list back. see how?

So, we also had earlier trouble in that everybody's goals lists were getting lost, and nobody could find their goals list, and we're having to ship them everyplace. And this did make a barrier, to Routine 3. It's just too much administration. I mean auditors couldn't keep track of all of this that— that well. And we lost too many of them. Well, without this little gimmick about invalidation and getting the list back again and working it over, you'll find enough goals doing it that way to carry on. You'll certainly find the important goals. But only if you've gotten the invalidations off of the goals. That's the only one he won't give you a second time. That's too invalidated.

Somebody halfway found on him that he wanted to murder his father and they looked at him and said, "Tsk-tsk-tsk. That's a terrible goal," you see?

And then he said, "Oh, my, that's a terrible goal," or before that he said, "Oh, I wouldn't want to have that goal." That's going to be missing on the redone list, don't you see? That will be suppressed, in other words.

Well, all right. Now, what you learn in doing this, you can apply to straightening out any case. Whether this works all the way through as a clearing process in your hands or not is totally unimportant. Recognize that. You certainly have got to know all of these things in order to straighten out a case that's gone off the rails on 3D Criss Cross, you see?

I'm giving you the added bonus that we may be pioneering a little new way through on an easy route through the GPM, see? But you've got to be able to do all these things. You've got to be able to take all these steps to patch up cases, because you're going to find them that need patching up. And you've got a lot of them right here that need patching up, at this present moment. They're not in particularly horrible writhing agony, and they'd all come out straight in the end, and all that sort of thing, but you might as well learn to patch up a case fast, early, and know how to do it.

Now, I've given you two—I gave you a demonstration two weeks ago, I gave you a demonstration—yeah, a week ago, and I gave you a demonstration last night on just reviewing cases.

Now, I wouldn't bother to try to pattern my advice to these cases and try to apply it to all cases—I mean what you saw on the screen. Because I was just talking to these people, I was interviewing, that was a research session, is what you saw. Nevertheless, that's what I would do to straighten up these cases, see? And if they don't straighten up, I'd do something else. Do you understand?

But certainly, straightening up the first item that went wrong is the inevitable step which you would take in any case. you try to find the first item that went wrong in this person's Routine 3-type processes, and make it right. And you'll find out, I'm pretty sure, uniformly, that a case will continue to hang up until you either got clear to the end of the Goals Problem

Mass, and had about sixty or ninety 3D Criss Cross items, you see—pc in misery all this time, see, till he's getting down toward the end. And you find out your best shortcut—because all that time you're going to have a hard time keeping the rudiments in, see—your best shortcut is to find that first item in.

All right, if you're going to have to find that first item in, you might as well learn this process we call Routine 3A Criss Cross and it makes a patch-up in any event, whatever else it does. Okay?

Audience: Mm-mm.

All right?

Audience: Yes.

All right. Thank you very much.