This is a transcript of a conversation over Skype, New Year's Eve (December 31), 2010.

The conversation was prompted by a Twitter exchange on exploratory testing (ET) started by Andy Glover, who observed that "When developing scripts you need to explore. But this tends to be exploring with out the s/w so I would say it's not ET." I disagree; developing scripts is test design, and test design is certainly part of testing. Since the process of developing test scripts is an exploratory (unscripted) process, I would contend that script development is both exploratory and testing, and therefore exploratory testing. To get around Twitter's limitations, I proposed an impromptu Skype chat. Anna Baik, Ajay Balamurugudas, Tony Bruce, Anne-Marie Charrett, Albert Gareev, Mohinder Kholsa, Michel Kraaij, and Erkan Yilmaz joined the conversation. Alas, Andy had other commitments and couldn't be with us.

I have edited the transcript as best I can to capture the intended order of the conversation. I've also added a couple of clarifying points, in [square brackets].

---*Michael B.*

Michael Bolton: We appear to be developing a quorum. The one person we're missing is Andy, who begged off. We'll have to draw a cartoon about *him*. (devil).

Tony Bruce: Andy started out with, "Do some people do ET first and then scripts? as I've always read 'scripts first then ET'," and it spiraled into a bunch of different conversations.

Michael Bolton: Yes. Okay. Let's examine the questions of "Which do you do first?" and "Which comes last?" Can anyone see a problem with those questions?

Anna Baik: Excludes doing both simultaneously, for a start.

Anne-Marie Charrett: Assumes that they are separate exercises.

Tony Bruce: I thankfully haven't had to use a script for years but the questions discount which one is going to be the most valuable.

Ajay Balamurugadas: Why these questions? What problem are we trying to solve? Is there a problem (for Andy) here?

Tony Bruce: @Ajay, it came from a thesis (http://bit.ly/hQq9w3). Being a thesis, it's not exactly concise.

Ajay Balamurugadas: Ok. From his first tweet, he asked should we do ST (scripted testing) after ET.

Michael Bolton: Here's the principal problem that I see: we haven't really defined our terms.

When I get into discussions like this, I sometimes run into a problem where I want to figure out what we're talking about. This leads to some people saying that I'm "going meta". Well, yeah...

because it’s kind of pointless to have a conversation about this when no one’s clear on what we’re talking about (or worse, when everyone’s clear of *his or her own* notions). So I’m going to do that first. What is scripted testing? What is exploratory testing? What’s the difference?

Erkan Yilmaz: My definitions: Scripted Testing = following a (certain) script? Must this script be written down? Can it be transmitted orally?

Michael Bolton: The question marks are interesting.

Tony Bruce: Ah but also, one more thing, what are they in the context of where you are using them?

Michael Bolton: I’m thinking of a *relatively* context-free notion.

Anna Baik: Is the difference in the form, or in the intent?

Ajay Balamurugadas: I’m with James’ definition. Scripted testing: test design is done in isolation. Exploratory Testing: simultaneous design, learning, execution.

Michael Bolton: Simultaneous? Lots of people have problem with that word. If I write down some ideas first and follow them, is that non-E.T.?

Tony Bruce: One of the differences is that if you follow the script and only follow the script you are limiting yourself.

Anna Baik: I have watched a lot of intelligent testers executing procedural scripted tests. They rarely stuck exactly to the script (usually it was out of date anyway, and that’s the "excuse" they’d have given you), and would deviate to investigate "something weird". I would say they were exploring under the guise of "following the script".

Erkan Yilmaz: Out of date: I agree.

Michael Bolton: Can you follow the script and *only* follow the script?

Tony Bruce: No, but some people will try and follow the script to the letter, and not think of investigating anymore.

Michael Bolton: Okay. How would you deal with a script that started like this: 1. Restart the test system. (Has anyone seen a script that looks like that?)

Erkan Yilmaz: Sure, even wrote such of these.

Anne-Marie Charrett: Heaps. I probably wrote one like that.

Tony Bruce: Exactly! That's a terrible script but "testers" may well start the system and then restart it, or just ignore it and start the system and follow the script. The thing that is missing is the inclination to question why and what—or actually the inclination to question.

Erkan Yilmaz: I remembers Michael or James blogged about "restarting system" once, or?

(That was here: http://www.developsense.com/blog/2007/04/conversation-about-scripted-test/)

Michael Bolton: Anne-Marie, what did you expect your script-follower to do upon seeing that instruction?

Anne-Marie Charrett: That was up to them really. I tried to keep my scripts as vague as possible.

Michael Bolton: That was up to them? You mean you didn't try to *control* the tester? How can you expect consistent results from that?

Anne-Marie Charrett: I didn't expect consistent results. I expected intelligent answers. Most testers never follow the script precisely so why try and write a precise test script.

Michael Bolton: What would constitute an intelligent answer? Remember, there are lots of steps after that first step.

Anne-Marie Charrett: I was afraid you were going to ask me that. :)

Michael Bolton: There's no need to be afraid, because you're going to provide an intelligent answer. :)

Anne-Marie Charrett: Perhaps "answer" is the wrong word. Better information. A better word would be information.

Michael Bolton: Would "information" *really* be a better word? Substituting one vague word for another like that makes me suspicious.

Anne-Marie Charrett: "Answer" suggests a specific question.

Michael Bolton: Yes. And you didn't ask a question, in that first step.

Anne-Marie Charrett: Information is a better word because it's more open.

Erkan Yilmaz: Vagueness saves the time to adapt script again when software changes, the scripts often lag behind the state of the software.

Tony Bruce: But then if you have vague scripts why bother with a script?

Mohinder Khosla: If managers understand that we are all in together to deliver the perfect software then testers can provide intelligent answers about the software that is fit for purpose or deliverable.

Michael Bolton: If managers understand that we are intending to deliver perfect software, there there's a lot of misunderstanding going on. If software could indeed be perfect, how would we know?

Mohinder Khosla: There should be consensus around the word "information" so it means same to all.

Tony Bruce: Information *can't* mean the same to all because we are all looking for different information.

Mohinder Khosla: We never know until a bugs creeps along. Certainty is rare and we should not agree to deliver.

Michael Bolton: Certainty is rare in the same sense that unicorns are rare.

Erkan Yilmaz: Unicorns exist (in my imagination). ;)

Michael Bolton: Okay, everyone: Let's assume that Anne-Marie is the test manager here, and the rest of us have received that instruction ("Restart the test system.") What, *specifically*, will you do?

Erkan Yilmaz: Okay. Should there be an action before restarting system? e.g. software must be in a certain state?

Tony Bruce: Ask how I'm supposed to restart the system without starting it. Ask why I'm restarting it.

Ajay Balamurugadas: Ask her questions.

Michael Bolton: But the first instruction of the script is "Restart the test system." You'd go off the script on the *first* step?! Tony is going off the script too. And apparently so is Erkan.

Anna Baik: Look at the other scripts. Look at the rest of the script. Try to figure out the purpose of restarting the system there. Then check my assumption by saying "Hey, Anne-Marie?" Before that, I might actually try running a couple of variants, if she's not around.

Michael Bolton: So now Anna isn't following the script. Is *no one* willing to follow the script?

Tony Bruce: Unless somebody else has given me a system to restart I can't start the script. And I wouldn't anyway because I would want to know what they had done to get it to the state to hand it to me to restart.

Michael Bolton: So I'm confused now. I have (or Anne-Marie has) given a perfectly simple instruction, and no one has followed it.

Anne-Marie Charrett: As test manager I am a bit upset. No-one has followed one step!

Tony Bruce: I'm on lunch break!

Michael Bolton: So, what's the problem here? What if I gave you a very general instruction, "Explore the test system." Would you know what to do?

Tony Bruce: Yes and no. The system could be huge. Even if I was just having a skim through I might want to break it into smaller chunks. (Is it just in testing that every answer starts with "it depends..."?)

Erkan Yilmaz: No, system thinkers also use that a lot.

Erkan Yilmaz: One problem with "1. Restart the test system" is that it is vague. There are different ways to do that, and it doesn't say anything why this is needed.

Mohinder Khosla: No in simple words, Michael. Before you embark on a journey you do your home work and get all the information available to you and rest you explore later, learn and move on.

Michael Bolton: I don't do that. I often go out for a walk without gathering any information at all. I often go shopping without gathering any new information.

Mohinder Khosla: You go out with a view of exploring and learning. If say test a system then you need the right amount of information before you start testing.

Anne-Marie Charrett: Will someone please just restart or explore the system! We have a deadline here...

Mohinder Khosla: Michael, I missed the beginning or start of the conversation and your charter. (This is what happens when you join mid-stream.)

Michael Bolton: So the instruction, Mohinder, was "1. Restart the test system." But you have (perhaps accidentally) made the most important point of all. That is: it's crazy to attempt Step 1 without understanding what the charter (more precisely, the mission) is.

Anne-Marie Charrett: Here we go again, that crazy ET dude, going on about missions....

Anna Baik: Is it crazy? How costly is it? Would I understand the mission better if I just tried it a couple of ways first?

Michael Bolton: Suppose it's a nuclear power plant. Ooops... I just told you something about a possible mission. :)

Anna Baik: I'm testing on a production system? Or a virtual system?

Michael Bolton: There *you* go, asking questions again. :)

What we're arriving at, I think, is that the script tells you one thing, and your context and your mission and your management and the system that you're testing all tell you other stuff. Which means that NO test is entirely scripted, unless you're a machine, and therefore in no position to understand or question that other stuff.

Anne-Marie Charrett: As test manager I'm assuming you know all that stuff. You mean you don't? Oops!

Michael Bolton: Heh. Yes, Anne-Marie-the-Manager always has *bags* of assumptions about the level of information in the script vs. the level of information from elsewhere.

Ajay Balamurugadas: One simple question: Which system to restart? (All the machines here are switched off. (It wasn't me!)).
Mohinder Khosla: Restart of a system may require rollback to database, recreating ref data, recompiling code—so you have to be very specific what you mean by restart. You may even mean bounce the system.

Michael Bolton: See, that's interesting. I don't know what you mean by "bounce the system". So I'd blow the script right away.

Mohinder Khosla: Some may even call it reboot the system. Even then "bounce" has a different meaning. Bounce the system without taking it down? Just the bit that is giving you trouble or hung up?

Anne-Marie Charrett: Sorry guys—duty calls. Your test manager has to leave for a very important meeting. Please don't all say bye. I will take it as said.

Ajay Balamurugadas: No surprise (Typical test manager :)).

Erkan Yilmaz: No worries. You will feel it when the plant explodes.

Anne-Marie Charrett: I trust you guys. You will do a great job. Hee hee.

Tony Bruce: Bye, Anne-Marie, and Happy New Year!

Michael Bolton: Again Tony fails to follow the manager's instructions!

Tony Bruce: That's me! If I'd listening to all my test managers in the past I wouldn't be a tester :)

Michael Bolton: So, to me, the question is not "is this a scripted test or an exploratory test"? That's not a terribly helpful distinction for Anne-Marie-the-Manager, or HER manager. When Andy (@cartoontester) says (as he did in a tweet) "My point is that most (not always!) explain ET and ST as opposites because that way is easier to explain process"... well, it may be easier. But it's also easier to close your eyes while driving a school bus, because it saves you from having to consider all that annoying, distracting stuff that's going on around you.

The most helpful metaphor I've seen is (surprise!) James': There's scripted testing, and there's exploratory testing. And there's cold water, and hot water. That is, there are opposite poles for both. But things don't happen at the poles unless (scripted) you're a machine, or (exploratory) you're a baby or toddler, encountering an object for the first time, and playing with it while learning, but without having a learning mission assigned to you.

One big problem that I see with Andy's strategy (or the one he reports on, at least) is that managers who don't know how to observe testing will trust scripting and adopt *the illusion of* control.

Tony Bruce: It's not a illusion! If 97% of the tests passed we're all gravy!

Mohinder Khosla: Are ST and ET two ends of the same string while poles apart? Or do they co-exist?

Michael Bolton: Yes to both questions: Exploratory testing and scripted testing are two ends of the same string *and* they co-exist. We've been talking about ET and ST as points on a continuum for years, and many people still don't understand that. It's sometimes exasperating. But we have to work from the perspective that it's probably our failure. We haven't explained it well enough for it to be clear. We're still learning how to express this in ways that they'll understand and be persuaded. The trouble is that understanding also depends on observation, and many managers have a hard time with observation.

Anna Baik: I'm not sure which tweets you're referring to here. I got the impression that they ran scripted tests alongside exploratory, but reported on scripted?

Michael Bolton: I'm talking about this one particularly: http://bit.ly/dYTYR2, in which Andy says "My point is that most (not always!) explain ET and ST as opposites because that way is easier to explain process".

So, for managers who don't observe testing very well, they'll grasp the poles without understanding the enormous middle between them. Middle, or maybe "muddle". :). Here's the explanation that I feel most comfortable with these days:

The opposite of exploratory testing is scripted testing. Neither of these is a technique; they're both approaches to testing. Irrespective of any other dimension of it, we call a test *more* exploratory and *less* scripted to the extent that

- elements of design, execution, interpretation, and learning are performed by the same person;

- the design, execution, interpretation, and learning happen together, rather than being separated in time;
- the tester is making her own choices about what to test, when to test it, and how to test it—for example, the tester may use any automation or tools in support of her testing, or none at all, as she sees fit;
- everything that has been learned so far, including the result of the last test, informs the tester's choices about the next test;
- the tester is focused on revealing new information, rather than confirming existing knowledge about the product;
- in general, the tester is varying aspects of her tests rather than repeating them, except where the repeating aspects of the test are intended to support the discovery of new information.

Tony Bruce: Unfortunately the amount of test managers who are able to observe testing is scarily small. Not that I'm any great thing but it means that one day I might have to become a test manager. Bah.

Michael Bolton: What I've listed above takes longer, much longer to say than "We mostly do scripted testing" or "We use exploratory testing here." But if we want to serve our clients well, I think we have to be clear on the points above ourselves, and be able to rattle them off at a moment's notice. I agree that managers observe poorly. So we need to help by giving them points of the process, factors, that they can observe.

Tony Bruce: Do you get a lot of clients actually asking about ST or ET? Or do they just want the information?

Michael Bolton: I do hear a lot of managers saying, "We want to start doing exploratory testing," not realizing the extent to which it's happening *all the time*. Another metaphor that James uses (that might get everyone's attention): exploratory testing is like sex. (Good; you're all paying attention now.). We don't have sex education because people don't know how to have sex. We have sex education because people have sex dangerously, or unsafely, or with consequences of which they're not aware.

Ajay Balamurugadas: Brilliant.

Anna Baik: And abstinence programs are of limited utility.

Michael Bolton: In any society, sex is happening all the time. A lot of it is slapdash; a lot of it is, well, unnecessary. A lot of it has undesired effects and severe, long-term consequences. So we're trying to raise *awareness* of sex.

Mohinder Khosla: I hear from some people (can't remember their names) that ST and ET are exclusive and do not compliment each other. Then I see ET being done where you script the test and the results as you test the system and report it. Then I hear some say ET is not for them although it may provide vital inside information about the software.

Michael Bolton: ST and ET are not exclusive, any more than hot water and cold water are exclusive. Those people are talking about *extremely unscripted* (or, quite possibly, *incompetent*) testing. When people say, "We don't do exploratory testing," I reply, "Oh, really. Then *all* of your processes must be scripted. Can you show me the scripts that you use to create your test scripts?"

Ajay Balamurugadas: James' article on ET, "ET Explained", is at http://www.satisfice.com/articles/et-article.pdf. (Most of you must be aware of this).

Michael Bolton: I'd suggest some more recent stuff to supplement that.
http://www.developsense.com/blog/2008/09/evolving-understanding-about/.
http://www.developsense.com/blog/2010/03/coding-qa-podcast-on-exploratory-testing/.
http://www.developsense.com/blog/2010/03/looping-and-branching-in-exploratory-testing/.
http://www.developsense.com/blog/2010/04/coding-qa-podcast-on-exploratory-testing-part-2/.

Erkan Yilmaz: To the thing you said before, that people misunderstand ET–are there thoughts to not talk about ET anymore? Use another term for naming things, or aspects of them?

Michael Bolton: The opposite, I think. "People misunderstand sex, so let's not talk about sex any more." Bad idea, I'd say.

Erkan Yilmaz: I see.

Michael Bolton: Also, "people have the heebie-jeebies about sex, so let's call it something else". Some are going to have the heebie-jeebies about it no matter what we call it, so unless there's a meaningful distinction or refinement in terms to be made, let's learn to get over the discomfort.

Ajay Balamurugadas: Sometimes we might try renaming. For example, MTBS (Managing Testing Based on Sessions) vs. SBTM (Session-Based Test Management).

Tony Bruce: I think people who state ET is not for them and that they don't do it (how to put it politely) are just unaware and have very little understanding about testing. And if they make statements like that are probably not likely to take the initiative to learn. It is impossible to test something without doing ET. It may not have structure to it. It may not be "official". But it's being done.

Michael Bolton: I would argue that there IS structure to it, always. The trick is to become aware of the structures so that they can be observed and improved. (The structures are here, by the way: http://www.developsense.com/resources.html#exploratory).

Anna Baik: I would have said we "weren't doing ET" in my first role. But when I learnt more about exploratory testing, I realised that we did a lot of it, all the time. Sometimes people do stuff but have a different name for it. (We called it "off-piste").

Mohinder Khosla: Awareness requires planning, initiatives, volunteers and authoritative people to implement, and that's where we testers community comes in to educate the nonbelievers.

Tony Bruce: Heh, “nonbelievers” makes it sound like a cult :).

Erkan Yilmaz: Cults get tax benefits?

Ajay Balamurugadas: Okay, a question: Can a script be followed word by word?

Erkan Yilmaz: Probably every word has different meaning to everybody.

Mohinder Khosla: A script can be a one liner to start a conversation and confirm what the test is about and leave rest to the tester. A long script can be confusing, so do not follow word by word.

Michael Bolton: I agree, with one caveat. You don’t have to leave the rest to the tester; you can start the conversation and continue it as testing goes on.

Ajay Balamurugadas: There’s a common misconception: anything with no documents = any testing with no steps followed = ET/Adhoc.

Michael Bolton: Does everyone here know why we don’t call it "ad hoc" testing? Also, does ANYONE know what "ad hoc" means?

Mohinder Khosla: I have seen it called mop up testing. Very odd.

Ajay Balamurugadas: The usual understanding of the word ad hoc is “sloppy work”.

Erkan Yilmaz: Ad hoc: goes in direction of like spontaneous? And probably is more misinterpretable than ET.

Anna Baik: Because most people interpret that as meaning slapdash?

Michael Bolton: Yes... *mis*interpret that as meaning slapdash. The Rogers Commission on the Challenger was an ad hoc commission. That doesn’t mean it was sloppy, or slapdash, or incompetent. "Ad hoc" means, literally, "to this", "to the purpose". In that sense, we believe ALL testing should be ad hoc—"to the purpose".

Erkan Yilmaz: http://en.wikipedia.org/wiki/Ad_hoc.

Michael Bolton: It’s the latter sense in the second paragraph of that Wikipedia entry (“*Ad hoc* can also have connotations of a makeshift solution, inadequate planning, or improvised events) that we want to avoid, or at least downplay.

Anna Baik: So are you saying that we don’t call it "ad hoc" because people get the heebie-jeebies about that word?

Michael Bolton: No. Scripted testing can be ad hoc too, in the good sense. It’s more that people don’t understand what ad hoc means—but since Cem separated the terms and James adopted them, we started discovering implications of the idea exploratory testing, and that continues as time goes on.

Ajay Balamurugadas: This Skype chat was setup in an ad hoc manner to help discuss about ET?

Michael Bolton: Yes; this chat is ad hoc. And that's good example of ad hoc not being slapdash. I don't think this chat is terribly slapdash, as such things go.

Erkan Yilmaz: I must say I got to know ad hoc when I learned about ad hoc networks.

I must think more, but one thought I had was this: instead ST or ET, why not call the testing by something more individualized: (name of person)'s testing? For example, "Thomas's testing". Must ponder more about it.

Michael Bolton: I can *guarantee* I won't understand "Thomas's testing" unless I know something, a lot, about Thomas.

Erkan Yilmaz: Yes. But that gives you the desire to think more about it and ask questions. Anyway, we need not discuss this here. I will ponder it more alone, since these are unfinished thoughts.

Michael Bolton: I can say that I *do* understand something about exploratory testing, in that I understand the factors and the dynamics of it. I can tell you what to observe that indicates a scripted process versus that which indicates an exploratory process. At one point, I also created a definition for scripted testing that respected the valuable points about ST (and there are plenty of them).

Mohinder Khosla: Thinking about Agile testing: it is not planned testing but ad-hoc, because you are in the thick of things but don't know when the code will be ready to test. So you drop everything to test. You may not have time to script tests for new features. You may end up doing ET.

Michael Bolton: Not may, *do*. Exploratory testing is *not* unplanned testing. Exploratory testing, however, *does* take the approach that the tester has the freedom and responsibility to use the plan as a tool, following and deviating from it as new information arrives and learning happens. *Very* exploratory testing is testing that is *not very much at all* guided by specific instructions.

Anna Baik: I may well gather more information about acceptance criteria, have more conversations with developer and the product owner, or put together test data. And I usually know that if there are three stories under development at the moment, I'm going to be looking at those first.

Michael Bolton: "I'm *probably* going to be looking at those first." :)

Mohinder Khosla: During a sprint, sometimes, stories take shape such that the delivered feature is not what was on the card. You are constantly updating the acceptance test, but when you get down to test execution you realise that acceptance tests that you have written are not valid. So you end up doing ET based on the conversation you had with the developer.

Tony Bruce: If features are being delivered that weren't on the card then communication has gone out the window. And not only are you nowhere near 'being Agile' your test ideas won't be the only issue suffered by the team.

Michael Bolton: I'm not so sure about that. Practically any feature (especially one that you can put on a card) has sub-features implicit in it. In that sense, the card is kind of like a script, isn't it? That is, the description now never *perfectly and completely* fits the result later.

Tony Bruce: I'd call it a guide rather than script but that will vary, my point was that if the team is communicating effectively then everybody should be aware of what is coming regardless of how close it fits what's on the card, it may differ completely but that hopefully has been communicated.

Mohinder Khosla: The information on a card should be enough that can start a conversation and can be read a metre away. It should not be a script otherwise you will take it a requirement without questioning.

Anna Baik: I'd expect the original card to "fill out" a bit during the discussions that go on while it's being developed. So long as I'm in on those discussions, no worries.

Tony Bruce: Actually I'll go one further and say that if you have an effective team and what is being developed differs to what is on the card then that card should be de-scoped and a new card written up.

Michael Bolton: Yes, perhaps. But things are what they are; the card is a medium, a representation of things.

Mohinder Khosla: The card would have written on it we need login screen but the details would be discussed by the team what how to achieve it and then written somewhere else for developer to work on and tester to design acceptance tests.

Ajay Balamurugadas: Focus on testers' skill over script… does ET emphasize this point?

Michael Bolton: If it's *good* ET, it must emphasize that point, yes. One thing that we say in the Rapid Testing course: heuristics are applied, not followed, in the process of testing. Scripts provide the illusion of control over the process. Heuristics are applied most productively by those with the wisdom and skill to apply them (or not apply them) appropriately.

Erkan Yilmaz: What I learned today: there needs much interaction and talk to understand others and myself better.

Michael Bolton: Yes. Transpection rules!

Erkan Yilmaz: I heard about transpection for the first time, I think, from Stephen Hill in Weekend Testing. He also blogged about talking with James Bach about it. I have definitely some more things to think about now, plus tons of links to read :)

Ajay Balamurugadas: http://pedantictester.wordpress.com/2010/12/06/transpection-explored/ Stephen Hill's blog post on transpection.

Tony Bruce: I must admit I've not looked into transpection yet.

Michael Bolton: Do a text search on "transpection" in James' blog and mine to see it in action. (In fact, this is also a transpection session.).

Ajay Balamurugadas: So, we have come to a common description of ST & ET by now?

Michael Bolton: Here's another key blog post: http://www.developsense.com/blog/2010/09/why-exploratory-isnt-it-all-just-testing/. I'm still trying to find my definition of ST.

Michel Kraaij: Michael, what is your goal in trying to find a definition on ST?

Michael Bolton: I wrote a definition of scripted testing to help people understand the contrast between the poles of ET and ST, and to recognize the huge middle/muddle/mangle between them.

Michel Kraaij: Michael, what is your current definition of scripted testing?

Michael Bolton: That's what I'm trying to look up. But for now, a test is scripted *to the degree* that the test is guided by a) someone else, b) ideas from the past, c) longer feedback loops, d) mediation.

Michel Kraaij: The keyword is *guided*, whereas ET is not.

Michael Bolton: Not quite "not". ET can be guided too. I'd refine what I've said above to say the more scripted the test, the more guidance leans towards *control*—or, when humans are involved, the *illusion* of control.

Anna Baik: So the more test design is separated from test execution, either by time or author, the further along the continuum towards the scripted pole it lies? I think that's a matter of degree.

Michael Bolton: Yes, exactly. And I trust you see the benefits *and* the risks in each approach there, too.

Ah, I've found something close to what I arrived at in the end. I said, "Scripted testing is an approach to testing that separates the processes of test design, test execution, result interpretation, and learning in person and/or in time, and that emphasizes process, planning and control of testing by focusing on documents to mediate management, supervision, and direction of the individual tester."

Michel Kraaij: Wow, that's a mouthful.

Michael Bolton: Sure, it's a mouthful. But if you want a short version, you can say "A test is scripted to the degree that controlled by someone else, at some point in the past."

Albert Gareev: Why was scripted testing promoted in the first place? To minimize expenses? Increase control?

Tony Bruce: It doesn't increase control, increases the perceived perception of control. Michael, you could always add "at some point in the past and so may no longer be relevant".

Michel Kraaij: That's a *possible* consequence.

Michael Bolton: I could add that. But one might also do something like that with ET definitions too: "...which come into the tester's head at the moment, and therefore might not be relevant to the already-established mission." Adam Yuret's blog post (http://bit.ly/gcEtPX) reminds me of that, in a way.

Anna Baik: I read Andy's question and the context that springs to mind is: are we talking about a project plan, where we must tell the project manager "we will start with two days of ET, then we will begin 14 days of ST, then we will..."?

Michael Bolton: I think that's a good example of the problem of seeing ST and ET as things that you do, as activities, rather than as *ways that you do things*.

Anna Baik: Yes. I think it's hard to disentangle scripted testing from the environment in which it often seems to be conducted. So in my experience, on past projects we've had to say "we are doing two days of exploratory testing" to "justify" why we aren't reporting test case passes. (Which is a whole different can of worms).

Michael Bolton: You're talking about the *learning* and *design* elements of testing, which don't always happen *simultaneously* with execution (that's why we now prefer "parallel" in our talk of ET). At various times, some things are getting more attention than others.

As for justifying the absence of passing test cases, that *is* a different issue. I don't have a problem with justifying not reporting test cases "passing". Most of the time, either tests aren't being executed (while we're learning or designing or doing other stuff), or they're resulting in failures (during execution). The eagerness for passing test cases is like the eagerness for good news in the paper every morning. We'd wish for that, perhaps, but there's this thing called reality...

Ajay Balamurugadas: Exploratory testing is a way that you test. It can be applied to any technique. Could you give an example, please, for the "applied to any technique" part?

Michael Bolton: Domain testing is a technique in which you analyze the data, and make choices about which data you're going to use for your tests. You can do this in a scripted way: let's thumb through our ISTQB course notes, and follow the process of finding every number in the requirements document, adding one to each number and subtracting one from each number, and there's our set of values for our domain tests. [Added later: Or you could do it in an exploratory way, by interacting with the product as well as with the documentation, and by making choices of values to try based on new information and new risks as you're discovering them. Another aspect of the exploratory approach is to *seek* boundaries, where the scripted mindset tends to presume that you already know where they are.]

Mohinder Khosla: Does ET apply to all situations? Planning, design and execution?

Michel Kraaij: Not everything is "testing", but you might call it exploratory planning (which is overkill, because most planning is exploratory).

Mohinder Khosla: Good point.

Michael Bolton: Your planning can be very exploratory, or less so (consider templatized test planning).

Ajay Balamurugadas: So, we can go either the scripted way or the exploratory way on any technique.

Mohinder Khosla: ET is a way of testing where testers express themselves to bring in their personality to the testing practice. Should we call ET as technique or practice?

Michael Bolton: ET is neither a technique nor a practice; it's an approach. “Exploratory” is not a noun; it's an adjective.

Albert Gareev: I see, definitions boiled down as exploratory = conscious thinking, scripted = mindless incurious following. No?

Michel Kraaij: Not just conscious thinking, but also acting.

Anna Baik: Oh good point :).

Albert Gareev: Conscious thoughtful acting?

Michel Kraaij: Possibly, but sometimes I also test based on gut feeling. Is that conscious or unconscious? ;)

Albert Gareev: You consciously listen to the unconscious part of yourself. Basically, you're consciously using an oracle.

Michel Kraaij: Indeed.

Michael Bolton: Yes, but note that your oracles may be applied in advance, in the moment, or retrospectively.

Ajay Balamurugadas: Retrospectively too? I always thought it had to be in the moment. You call it a bug when you have the oracle. Wrong?

Anna Baik: What if I said: “I think this is a bug. I can’t explain what oracle I’m applying yet.”

Tony Bruce: I get what you mean though, it’s not that you don’t want to explain but you’re on the hunt, maybe try talking out loud.

Michael Bolton: I’d say “What makes you think this is a bug?”

Anna Baik: I might say: “I’m not sure yet. Let’s look at this together and see if you think so too. Then maybe we can figure out *why*.” Pairing with our graduate tester has meant I’ve been saying "I’m not sure yet, but I think we have a bug here" more often. Normally I wouldn’t be trying to explain it to someone until a bit later.

Michael Bolton: It would be very, very good to practice that, though... for both you and the graduate.

Anna Baik: Yes, I’m trying to make sure that I try to analyse for her afterwards what made me suspicious. It’s very hard - but good for me.

Michael Bolton: I suggest that you talk it through *in the moment*.

Anna Baik: By afterwards, I mean about two minutes later! Or less. Sometimes it’s hard to verbalise immediately.

Michael Bolton: That’s why it’s important to practice.

Anna Baik: Isn’t there a risk that by trying to put something into words immediately, you change how you’re thinking about something?

Erkan Yilmaz: You mean the observer influences the experiment?

Michael Bolton: There’s a risk in doing pretty much anything, sure. But if you’re not sure there’s a bug there, *one approach* is to think it through out loud. (That may not work so well for introverts, though.)

Tony Bruce: “What makes you think this is a bug?” Your bug sense goes off and you think “something’s not right”.

Michael Bolton: "Bug sense" means that there’s a heuristic being applied. Can you articulate that? "Something’s not right": *what* is not right?

Tony Bruce: I’m not sure I can articulate it. I’ll have to explore it further next time it happens. I’m sure subconsciously something has set the bug sense tingling, and therefore a heuristic has probably been applied but it wasn’t consciously.

Ajay Balamurugadas: You could say, “Not consistent with user’s expectations”—and that you are still thinking.

Tony Bruce: Users expectations or your expectations of a user’s expectations?

Michael Bolton: Have you ever said this? "Oh... THAT’s what was happening. Wow, that was a bug!"

Ajay Balamurugadas: Yes, but I will get back.

Tony Bruce: So, in reference to Ajay’s “You call it a bug when you have the oracle. Wrong?”: There is no wrong, just different circumstances and context but there are some cases (as Michael stated) when something has happened and you later on realise it’s a bug.

Ajay Balamurugadas: I agree. I disagree when people say : Its a bug but oracle - will be decided later. I agree with "the moment you say a bug, you’ve applied an oracle".

Michael Bolton: The moment you say "it’s a bug", yes, you’ve applied an oracle. But my point (I think) was that the oracle can be applied long after the test was “complete”.

Ajay Balamurugadas: Yes, It was a bug ... (And we did not notice it) (Yes).

Tony Bruce: If you talk with a developer or product manager or product owner or business analyst after you found a bug to confirm it’s a bug you could say *that* was retrospective.

Albert Gareev: Unfortunately, often, if it’s signed off, then the page is turned and no one wants to go back. I used to find bugs that went to production only because the testing phase was ended and the chapter in bug repository was closed. Couldn’t log, thus no ticket, thus no need to fix.

Michael Bolton: Logging and reporting are not the same thing. Be careful. Also "couldn’t log" doesn’t mean "no problem".

Tony Bruce: No ticket no problem is a whole other issue.

Albert Gareev: IMO, it relates to "kill the messenger bearing the bad news". Punishment culture in an organization.

Michael Bolton: The good news is that you don’t have to work for such organizations. Fear not; reality will catch up with those who believe that the absence of a ticket means the absence of a problem. “I didn’t get the fax from the doctor, therefore I don’t have cancer.”

Albert Gareev: 1) You may not know that it’s a kill-the-messenger organization prior joining. 2) Even there you can influence a change, can’t you?

Tony Bruce: Sometimes enabling the change can be too much of a battle, and so you may not know prior to joining… but you know on leaving.

Albert Gareev: Often, it is recommended to ask a BA whether it's a bug or not. And if it's not in their BRD (business req doc), then...

Michael Bolton: Does the BRD say, after every line, "and the program shall not crash"? Does the BRD say "and these are the ONLY things that could possibly matter"?

Albert Gareev: No.

Michael Bolton: Then anything is potentially in scope.

Albert Gareev: Such questions will be taken as an offence.

Michael Bolton: I'm not suggesting that you ask those questions specifically, and certainly not so directly as I've done here. But there are related things you can do.

Tony Bruce: It can be phrased differently. You both want the customer to be able to work effectively, or something along those lines.

Michael Bolton: Here's what I've done. "I see a possible problem here, based on the idea that I see this result here and that result there. That is, these results are inconsistent, and that means that the product is handling them inconsistently. I infer—even though it's not explicitly stated in the BRD—that they should be handled consistently. I see a risk that this could cause harm, or confusion, or data loss, for our customer. Would you consider this to be a problem?"

If you're in a blaming organization and they say, "That's not a problem," and you really think it is a problem, the next line in the conversation is, "Okay... just to be sure, I'll send you a quick little note about the problem in email, so you can review it whenever you like."

Anna Baik: Albert, it sounds like you're in a bit of a no-win situation there. Or at least one in which things seem to be generally set up to discourage reporting of potential issues. (The "shoot the messenger" you were referring to earlier).

Albert Gareev: Any big orgs I worked for are kinda like that. But what is a "win"?

Anna Baik: You both want the customer to be happy? More than you want to tick the box for the next step of the "Process"?

Albert Gareev: Honestly, I don't know. It's too abstract for me. I can't give a universal definition of happiness. With the regards to testing, I want time spent with a payback, not wasted.

Anna Baik: Does following a test script designed by someone else as closely as you can, mean that you can't apply conscious thought to analysing the results?

Albert Gareev: I’d like to deviate from a script or abandon it completely whenever I think there is *something*. If it was a wild goose I might get back to the script or start it over.

Anna Baik: That’s one example where you might want to use a more exploratory approach, and it might be justified. But does that mean that someone who notes it for later, carries on, and then analyses the results thoughtfully is being mindless?

Albert Gareev: But if you note for later then how you go back to an app to investigate? Or do you mean future test ideas?

Anna Baik: You might go to your test manager after the session and say "I noticed this while I was testing: is this worth exploring further? Should we add further tests for this?"

Tony Bruce: Somebody posted a good blog post on gut feeling, can’t remember who tho.

Michael Bolton: Instead of saying "gut feeling" as the basis for your testing, consider "heuristics". ‘Cos that’s what they are.

Michel Kraaij: I’m currently reading the book *Gut Feelings*. Maybe a good reading suggestion?

Michael Bolton: No, *Gut Feelings* is not a good reading suggestion. It’s a *great* reading suggestion. Oh, and there’s no "maybe" there, either

Michel Kraaij: :)

Tony Bruce: Thanks, I’ll look it up.

Ajay Balamurugadas: The list of books keeps growing like our test ideas.

Michael Bolton: Apropos of book lists: http://www.librarything.com/catalog/MichaelBolton.

Ajay Balamurugadas: I am thinking of working part time in a library/book store :).

Mohinder Khosla: What should we tell our managers? They ask us “Which techniques are you using?”

Tony Bruce: Ask them what management techniques they are using.

Ajay Balamurugadas: Techniques remain the same. It’s the approach that changes.

Michael Bolton: Yes; techniques are orthogonal to the exploratory-vs.-scripted dimensions.

Anyway, I’m going to have to wrap up. I think that it’s most important to recognize that ET is happening ALL that time; that you can’t create scripts without a good deal of exploration; that testing is about learning to inform knowledge about the product AND knowledge about testing, ideally with informing improvements in both.

In answer to one Tony's questions in Twitter... A little context: Andy said, "I agree, ET & ST are in a spectrum. When explaining process, it's easier to say "we do lots of ST here and a bit of ET". I replied, "That's an accurate statement if you choose to observe poorly the amount of ET that's actually happening." Tony said, "Who do you say that to?" My answer is "to everyone who'll listen, but especially to the managers."

I'd like a moment to return to the original question. "Which should come first? ET or ST?"

Albert Gareev: In what context?

Tony Bruce: Dunno, neither, both, it depends. Actually I'll go further. Dunno, neither, both, it depends, I'm not actually sure why and what you're asking.

Mohinder Khosla: I will vote for ET. Simply ET-ST-ET-ST-ET....

Ajay Balamurugadas: It's a loop? ET (learn)-ST (check)-ET (investigate an issue), or again depends on project, scenario, context?

Michael Bolton: [Added later: You can't really prepare scripts well without a ton of exploration. So in that sense, exploration always precedes scripting. I think it's important for managers to understand that, especially when they claim "we don't do exploratory testing" and take a heavily scripted approach to test execution. Towards the end of the development and testing cycles, you might (and, in my experience, almost certainly should) have a set of automated scripted checks that happen at build or release time. In that sense, there's nothing wrong with doing scripted testing after exploratory testing. And before it.]

I would argue that the entire process of testing is suffused with exploration. As James once said, "If it ain't exploratory, it's avoidatory." I think that, in answer to Andy's original quandary of which should precede the other, that the premise of the question is faulty. It presumes that "exploratory testing" is a thing that you do, a particularly activity. The reality, I think, is something different: exploratory testing is *a way that you test*. It can be applied to any technique. It's not a question of whether you're doing one or the other; it's a question of *the degree to which* the activity you're doing is being directed by someone else, or from the past (scripted), vs. the degree to which it's being done by the tester, in the moment.