A Taxonomy of Testing Types

featuring Don Firesmith as Interviewed by Suzanne Miller

Suzanne Miller: Welcome to the SEI Podcast Series, a production of the Carnegie Mellon University Software Engineering Institute. The SEI is a federally funded research and development center sponsored by the U.S. Department of Defense and operated by Carnegie Mellon University. A transcript of today's podcast is posted on the SEI website at sei.cmu.edu/podcasts.

My name is Suzanne Miller. I am a principal researcher here at the SEI. Today, I am pleased to introduce to you to Don Firesmith, who is an SEI principal engineer whose current work focuses on software and systems testing, current and much past work as well. He has been in this business a long time.

Today, we are going to talk about a taxonomy of testing types that Don has recently created. But, first, a little bit about him. And, he has been a frequent guest on our show, so I apologize if you have heard this before. At the SEI, he provides practical guidance with regard to requirements engineering, system and software architectures, and testing to help government program offices acquire software-intensive systems. He also develops and maintains new technologies, such as methods for engineering safety and security requirements as well as assessing the quality of system requirements and architectures. He is also the author of countless journal articles on software and systems engineering as well as seven technical books including, most recently, *Common Software and System Testing Pitfalls: How to Present and Mitigate Them: Descriptions, Symptoms, Consequences, Causes, and Recommendations*. Quite thorough. Don, thank you for joining us today.

Don Firesmith: I'm very happy to be here.

Suzanne: So, let's start off by talking briefly about the current state of software and system testing as you observe it, and particularly in terms of different types of testing. What do you observe, and what drove the need for a taxonomy as far as you are concerned?

Don: Well, what I see often is that the projects out there, they will document what their test plans are, their test strategies, in some kind of document up front. When you review those to see what they are planning on doing, typically only a very small number of different types of testing are even mentioned: unit testing, integration testing, system testing, and so on.

Yet, if you go out into the test community—the literature, the training courses, the books, the articles—there are literally hundreds of different ways of doing testing. And, it just became very obvious that the test planning was incomplete. It's a case of out of sight out of mind. If you don't actually plan on doing a certain kind of testing, odds are that testing is just not going to take place.

Suzanne: You don't plan on doing load testing. You are not going to do load testing.

Don: Exactly. I decided that it would be a good idea to actually organize the information about the different kinds of testing because I could not find any one good source that had pretty much all of the different types of testing in one place. There are good books and things like that, but nothing that really seemed comprehensive. So, I decided to start off by just coming up with a list of testing types.

Suzanne: OK. So, tell us about that. How did you go about finding all those test types, and what did you find when you went looking for different test types?

Don: A lot of it came from past experience. Once again, there are tons of articles, blog entries, books. There are organizations that certify testers, and they have their own training materials, for example. There really truly is a wealth of material out there. The Internet is naturally a great place to find some of this information.

Like I said, I started off with just a list and definitions to go along with the list. But, it didn't take very long before there were just far too many different kinds of testing to have on the list. You get completely overwhelmed. So, I realized what we needed was some kind of taxonomy, some kind of hierarchy, some way to organize and structure them so that similar testing types were next to each other, if you will. And, you could find things easily.

Suzanne: And, also make decisions. If I have a whole group of tests that are related, I can maybe decide that *I do want to use this one, but I don't want to use those two, or I want to use all three*. So, it allows me to make more informed decisions as a test planner.

Don: Absolutely. And, it's not just whether I'm going to use this type or that type. It may very well be, this type of testing makes eminent sense for this functionality or for this part of your system, but it's not cost effective, doesn't make sense to use it any place else.

Suzanne: Sure. Or, I may not have the test equipment to use this testing method, but here's another one that's kind of like it that I have never tried before that doesn't use as much test equipment or the same test equipment. So, I could see where those would be cases where I would want to use a taxonomy to help me in that planning function.

Don: And it is not just the people who are doing the test planning. The government, people with oversight of system development programs, when they are looking at a contractor's test program, it would help them to figure out, *How complete is this test program? Are there any things that are obvious that we know should be done that aren't being done?*

Suzanne: Or are there particular things related to our problem that we know should be done that aren't typical? You may write a perfectly good software-based test plan, but it may not address particular issues that this customer has.

Don: Right. The idea of load testing. Performance testing. Capacity testing. If you have safety-critical software, then safety related testing is important. Robustness testing. Security is another issue. Whether it's penetration testing or any of a number of different kinds of testing types associated with security.

Suzanne: That seems to be a growing area more than many in today's world.

Don: Absolutely.

Suzanne: Give us an example of the taxonomy. Give us a little bit of help. We've got some graphics here that you brought today that help us to go through and understand the taxonomy. So why don't you tell us about those.

Don: Well, if you look at the first graphic, what you'll see is that there's a lot of different kinds of methods for doing verification and validation. There is testing, but there are also [other] evaluation approaches: analysis, demonstration, and so on. The first thing that I want to just make clear is this taxonomy doesn't cover everything. It only covers the testing, the part that's in the upper left.

Suzanne: So, that gray box is the test. And all the other things really are excluded from this taxonomy.

Don: Exactly.

Suzanne: There may be future ones, but this is the test taxonomy.

Don: Or if there was a bigger taxonomy, we're talking about that sub-tree with regard to testing.

Suzanne: And then where do we go from there?

Don: OK, if you look at the second graphic, what you will see is essentially the top three levels of the taxonomy. On the left-hand side, we have all of the different types of testing, if you will. Test types. If you look at the next column over, you will see that all of these different test types have been organized, categorized in accordance to what are historically known as the five W's and two H's.

Suzanne: So, these are the questions that those kinds of tests answer.

Don: Exactly. There would be, *What kind of thing is getting tested?* So the *what* question. *When test?* You know, when does the testing take place. *Why test? Why are you doing the testing? What are you trying to get out of it? Who tests?* OK, *who is doing the testing? What organizations? What people? What roles? Where is the test taking place?* So, that gives you the five W's. Then, you have the two H's. *How is the testing taking place?* And how well, it's not so much how well the testing has to take place, but testing to see *How well the object under test is performing*. So, this would be where you'd have your qualities. Your safety, security.

Suzanne: Quality attributes, sure.

Don: Exactly.

Suzanne: OK. So, from there you have got a set of test types that help you to answer those questions.

Don: Right.

Suzanne: Then, on the third graphic, it looks like we've got something that goes down a little farther.

Don: Yes. So, for example, if you look at the upper right-hand corner of graphic number two, under What Test, you see object under test-based test types. And then, if you look at graphic number three, that expands that one box to show all of the different kinds of testing that would fit primarily under that. So, for example, if we're talking about what is being tested, you could have models that are being tested, executable models. You could be testing hardware. You could be testing software. You could be testing systems, including systems of systems. You could be testing data centers. And, you could be testing tools and environments: both development environments and test environments. So, once again, you just keep taking the tree, and you can break the branches into smaller and smaller branches until at the very end you get the specific concrete types of testing that take place. And so, for example, I've mentioned at the very beginning that we talk about, [under] software [testing], you get unit testing, integration testing, and application testing.

Suzanne: I see those here.

Don: Well, yeah, those happen to be three leaves on one branch of a very large tree.

Suzanne: Yes. Yes. And things like model testing. I know that that is an area that actually gets forgotten quite a bit. We create models, but we don't actually test to see whether the model is valid. We move along and test the results of the model, but we don't actually verify that the model is correct. So, that's nice to see that being included there.

Don: Yes. There are a lot of ways to evaluate the models. Dynamic and static analysis, as I said, are out of scope of this taxonomy. They definitely help make sure that the model has a good quality before you implement from it.

Suzanne: But, many of these models are executable now.

Don: Exactly.

Suzanne: And, if you don't actually take advantage of executing the model, then you're not going to know if it actually gives the result that's expected.

Don: You can do exactly the same kind of testing for models as you do for software. In some sense, these models can be thought of as being software at just that much higher of a level of abstraction. So, it's really no different. You have test cases. You put the model in a pre-test state. You execute the model. And, you see if the model's behavior matches what you expect.

Suzanne: Right. And that's the same as what we do with software and system tests of other types as well.

Don: Exactly.

Suzanne: So, have you used this in customer settings yet? What are you expecting, if you haven't? What are you hoping to see happen when this kind of a taxonomy is used in a customer setting?

Don: This is sort of brand-new-as-we-speak right now. I'm in the process of formalizing it and getting it out there. Essentially, the way I envision it being used, one, is, if you will, as a checklist or a check hierarchy for when people are doing the planning of their testing programs.

Suzanne: Do we need to deal with these kinds of testing and be explicit about that?

Don: Yes. Let's make sure that nothing accidentally falls through the cracks. If we decide not to do a certain type of testing, it is because we thought about it and made that decision. Not because we forgot it even existed. Then, as I said, if your responsibility is oversight of some organization that is doing the testing as part of the development for you, then, once again, it can be used as a

checklist to make sure that you're looking for everything and seeing whether or not the developer is doing everything that they need to do.

This can also be used from a training standpoint. You can almost think of each one of these types of testing as being very similar to a testing pattern in some sense because it's a way of solving a problem in a specific context just like a pattern is. So, one of the things you can do is you can learn about the different kinds of testing by learning the taxonomy. Just like you could learn how to use pattern languages.

Suzanne: Different design patterns, OK.

Don: You know, by looking, learning the patterns that make it up. So, it can be a training thing.

Also like with pattern languages and patterns, we are giving names, if you will, to the different kinds of testing. Now, one of the problems that we have as a testing community is there's a lot of synonyms, near synonyms, and acronyms where the same acronyms means two different things. One of the things I'm hoping to do is to have this become sort of the official, if you will, structure, so that, if you use a term and I use a term, we are not only referring to the same thing in the same taxonomy, but you can also then turn it into an ontology. In other words, it's not just the pretty pictures. It's the data, the information behind the pictures, the definitions.

Suzanne: The meaning.

Don: The meanings of the things and how they relate to each other.

Suzanne: OK. And that may involve a standards effort in some future date. This may be a seed for that.

Don: Of course.

Suzanne: Kind of a standards effort, *IEEE* and other kinds of standards bodies will often take these kinds of things up.

So, how can our readers and listeners access the taxonomy? I know you have several different things coming up. Why don't you tell us about upcoming events and products related to this.

Don: OK. Well, we are going to have a webinar, one hour webinar coming up [Please click on the link to access the archived webinar.] that we'll put in.

We'll publish a link to that, where I will go through all of the different parts of the taxonomy.

Suzanne: OK. That's an education activity.

Don: So that's an education activity.

I will be speaking on this topic at the FAA's Verification Validation Summit coming up next month in Atlantic City, New Jersey.

I will also be doing a tutorial, half-day tutorial on it at our Software Solutions Conference in November.

Suzanne: And, that will be in the D.C. area.

Don: Yes. The last thing is I'm actually turning this into a book. This was going to be added to my Pitfalls book, but it became much too large. So, I am in the process right now of completing the manuscript for the first version of this. [Editor's Note: Since this recording, Don has updated his plans and will be publishing the taxonomy via a wiki site that will be linked here when it becomes available.]

Suzanne: Excellent. Excellent. So, we will include links to all of these resources and events in our transcript as they become available. So, you will be able to keep up with Don. I want to thank you, once again to Don and also to our listeners and viewers for joining us today.

This podcast is available on the SEI website at sei.cmu.edu/podcasts and on Carnegie Mellon University's iTunes U site. As always, if you have any questions, please don't hesitate to email us at info@sei.cmu.edu. Thank you.