Assuring Open Source Software

featuring Kate Ambrose-Sereno and Naomi Anderson interviewed by Suzanne Miller

---

Suzanne Miller: Welcome to the SEI Podcast Series a production of the Software Engineering Institute at Carnegie Mellon University. The SEI is a federally funded research and development center headquartered on CMU's campus in Pittsburgh, Pennsylvania. 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 you to Kate Ambrose-Sereno and Naomi Anderson who are from the SEI's Emerging Technology Center. In today's podcast we're going to be talking about open-source software and the current challenges it presents for government and industry, particularly on the trust front.

First, a little bit about our guests. Kate Ambrose-Sereno is a technical analyst in the SEI Emerging Technology Center. In this role she identifies the state of current practice, key gap areas, and potential technology solutions for government and private-sector needs. Her current research focuses on assurance and open-source software. Recently, she established partnerships with industry collaborators to elicit technological and programmatic gaps in big-data analytics for cyber intelligence.

Naomi Anderson began working for the SEI Emerging Technology Center in December of 2012 as a senior software developer. She has more than 13-years of experience in Java and C++. Her background is in software design and development of large n-tier systems, multi-tier systems. Welcome, Kate and Naomi.

Kate Ambrose-Sereno: Thanks, Suz.

Naomi Anderson: Thanks for having us.

Suzanne: Absolutely. Let's start by talking a little bit about the current environment and where open source fits into that because we've had open source for a long time, but it's sort of an emerging trend again. There are some new players, so why don't you tell us a little bit about that.

Kate: Naomi, if you don't mind I'll take this first one. Everyone seems to be playing in this space right now. There's been an increase in activity. The government certainly is looking at this for cost efficiencies. We're seeing an emergence of social platforms. The software repositories are serving as an opportunity for developers who have an interest in similar products to work on each other's software. We're seeing it as an opportunity for projects that maybe would have been just sitting kind of collecting dust, to actually see the light of day. So, there's been this resurgence in the interest of open-source software.

Naomi: To piggyback on what Kate said, the government is very interested in cost savings and time efficiency. If they can cut down the amount of time of doing trade studies on different open source products that would help them immensely as well.

Suzanne: This takes us into the area of trust and assurance. I've worked in open source in the past, and that's a really big deal is *Can we trust the open source and how much can we trust the open source?* How do you answer that, and what is it that your research is doing to help us to improve the trust in open-source software that is now becoming available or is available again?

Naomi: Kate, I'll take this one. Our research project is called A Data Analytics Approach for Evidence-Based Assessment of Open-Source Software. The main difference between going with a purely open-source software project versus going with a contractor is that you don't have a single point of contact for the trust. With the emergence of all of these repositories, there's a lot of extra data available that you can mine and produce evidence, so that the people making the decisions on which products to use will actually have evidence to say, *I've chosen this because of certain different metrics that they found were important*.

Kate: If I could just chime in, the current approach to assurance is actually to go with almost a trust-based approach where you would potentially know the vendor that is providing that software. You would have a contractual agreement. They would be contractually obliged to deliver certain expected functionality. You could go back to them and ask them for much of their acquisition documents, much of their development and design documents. You would have insights into that potentially. With open-source software, there's a lot of information out there. There's a wealth of material that can be mined to provide a more data-driven, evidenced approach to supporting a claim of assurance.

Suzanne: On the other hand, it means that the acquirer is the one that takes responsibility for mining that data, for finding out what is out there, what is not out there, because they don't have that single point of contact. They don't have a relationship to base the trust on, so they really have to rely on the data. If I were an acquirer, that's a lot of work for me to think about doing. Your project is going to help them by filtering through some of that information and helping them to understand where they should look and what kinds of things are available to them. I'm guessing that there's actually a fair amount of difference between one social platform and

another in terms of how much information they do provide. So, platform A may have a similar product to platform B, but it provides a lot more of the design documentation. It provides a lot more data that can give me, the acquirer, a sense of trust that this is going to continue, and it's with people that are actually professionals. Is that a good assessment of some of the things you're trying to do? Naomi?

Naomi: Yes, that's correct Suzie. So, we were looking at—there's a lot of metrics that have already been evaluated like number of bugs in software and static analysis and things like that, but those don't necessarily give you the context of what they're looking for. It is very time consuming to go to each individual product you may be looking at and trying to pull out all of the information about the developer networks and what their processes are as far as submitting code, doing bugs, and code reviews. What we were looking at was—between the different platforms available like Launchpad and GitHub and the Apache Software Foundation—what kind of information could we pull automatically and generate data so that there could be a compare and contrast against what's offered.

Suzanne: What kind of data are you finding to be data that differentiates between these different platforms?

Kate: What Naomi was talking about was that we sort of avoided just kind of bug count because anybody can look at bug counts and it's really not a meaningful metric. What we did was pursue a path of understanding more about the development environment and the activity surrounding the developers and also the code-review and code-committal processes. These things are very transparent on the social platforms, the software repos [repositories]. In some cases it's a little harder to extract that kind of information, but the data is there to make those conclusions. For instance, one of the things that we pursued was looking at the network of developers who were working on a certain code base. We tried to make connections among the developers and code reviewers to be able to see what ties were stronger. If there was a combination of developers and code reviewers that seemed to be more successful in getting code implemented for instance, then that might be a good thing to look for. Are these pairings actually happening throughout the code base? In which case, maybe you could then draw a conclusion that maybe this is a solid development process.

Suzanne: So, you're looking at some of the networks and analyzing some of the networks of people and the way that they work together, what kind of work they do together, and looking for patterns and that kind of data. Is that one way to talk about it?

Naomi: Yes, that's correct. To expand on what Kate said, one of the things we found though is that—for instance in the Apache Software Foundation they have a lot of documentation about how projects are supposed to go through the process and tools like JIRA for the bug tracking and Gerritt for the code review—every project that uses it uses it a little bit differently. One of the

challenges to creating really side by side view of more than one project is not every project implements their work flow or their processes exactly the same.

Suzanne: So, you're looking for abstractions that can be comparative as opposed to the implementation-level data, is that right?

Naomi: Yes.

Suzanne: Okay. You mentioned a couple of challenges. What are some specific challenges that you found in doing this research that either you're still struggling with or things that you're really proud that you were actually able to overcome?

Kate: Well, I'll take that one Naomi. When we first initiated the research, we were looking at the software repositories and trying to determine what the metadata would be that would be interesting to visualize, so that we can provide some picture of comparable things for the different open-source software packages. As we progressed through the research, we started seeing that the open-source software repositories were starting to implement these things already on their own. I think this shows that there's a demand for that sort of picture to understand a little bit more about the software and not just looking at the code.

For example, GitHub has a whole section that they just call *graphs*. You can go to any software that is in that repository and look at the graphs, and you can see things like code stability, what the diffs were, and what the changes were on the code base. You can see trends over time. You can see something they call *punch card*, which represents the volume of code committal happening during different times of day, during different days of the week. You can actually see—if there's a high volume of code committal happening during Monday through Friday, 8 a.m. to 5 p.m.—you get a sense that these are people that are employed and doing this work as part of the their job versus if you see a lot of late-night, after-hours coding happening, then you may guess that this person is doing it kind of as a hobbyist or just as an after-hours project.

Suzanne: So, those visualizations have two purposes. They have a purpose for the person contributing to the code base, *When is it not busy, for example, for me to put my code in, commit my code*? But, they also provide some metadata that you're looking at in terms of patterns of types of developers that are involved in this kind of code versus maybe you'll see a different pattern for a different kind of code.

Kate: That's right.

Suzanne: What are some of the other challenges that you're dealing with right now?

Naomi: I would say, to follow-up on what Kate said, one of the things when we went into this we were thinking that we can maybe tease out what type of quality attributes people would be interested in looking at and give a rating of *This is good*, *This is bad*. What we started to discover

is it really is context dependent because for one person using it, it's going to be very important that this is a group of people who are employed and working on this during the week. For another group, they may not care about that. It may be okay that it's people doing this part time. . We decided at that point that we should really just figure out what attributes we can pull out and just present them, and let the other person make their own opinion on whether or not this is good or bad.

Suzanne: So, where are you right now and what have you got to look forward to next?

Kate: We have been able to develop a prototype of a tool that will gather the data and provide some visualization for a very targeted set of attributes or characteristics. What we'd like to do certainly is we would like collaborators to help us. We'd like to push this forward. We think there's a lot of value in developing a tool either to expand the selected data that we analyze-maybe we'll find correlations among seemingly disparate dimensions and measures in software development.

We'd like to produce evidence for specific open-source projects that may be of interest to specific potential collaborators. We'd like to also test some assurance claims that are out there now using our data analytics approach to trace the evidence-to-claim sort of paths.

Suzanne: Okay. Sounds like you'll be busy, which I know Kate in particular—I don't know Naomi—but I know Kate needs to be busy otherwise she has to go sing opera. Even though she likes that, it doesn't really pay very well. Okay, so anyway, I want to thank both of you very much for joining us today. For more information about the work of the Emerging Technology Center please visit www.sei.cmu.edu/ about/organization/etc/. That's *etc* not et cetera.

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

Kate and Naomi: Thanks.